The Journal of Supercomputing (2020) 76:7438–7457
https://doi.org/10.1007/s11227-020-03169-2

New comprehensive model based on virtual clusters and absorbing Markov chains for energy-efficient virtual machine management in cloud computing

Mehdi Rajabzadeh[1] · Abolfazl Toroghi Haghighat[2] · Amir Masoud Rahmani[1]

Published online: 29 January 2020

Abstract

Utilizing from energy-aware solutions along with maintaining service-level agreements is one of the most important research issues in cloud computing. In the proposed model, monitoring the status of resources and analysing the obtained data have led to proper placement and consolidation of virtual machines through targeted migrations at the right time. In the virtual machine placement policy, the definition of absorption mode has been used in simulated annealing algorithm in addition to the formation of virtual clusters to prevent from unlimited increase in the length of created Markov chain in any temperature while maintaining the convergence. The results of simulations obtained from various scenarios in CloudSim indicated the proposed model has led to energy savings up to 14.3%, 19% and 21% on low load, average load and high load, respectively, compared to the best understudy algorithm, while the SLA violation has also led to a decrease in all three modes.

Keywords Cloud computing · Energy-efficient model · Virtual machine management · Virtual cluster · Absorbing Markov chain

1 Introduction

With users' increasing need for various services, cloud computing was emerged as one of the new technologies. Cloud computing is an Internet-based computational model to provide industry-wide services. In this technology, the users' requests are provided as a service through the cloud. Generally, cloud computing is offered in

✉ Abolfazl Toroghi Haghighat
haghighat@qiau.ac.ir

[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] Department of Computer Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

the three forms of services: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).

Cloud computing contains a number of benefits such as reliability, quality of services, and robustness in their delivery. In recent years, the infrastructures of cloud computing have been growing rapidly since there is demand for computation required by users, and advanced datacentres in the cloud host a variety of applications. The extensive use of clouds is accompanied by an increase in energy consumption and carbon dioxide emission. That is why green computing is of particular importance in this area [1].

Large cloud service providers use several megawatts of power for datacentre operations. They annually spend millions of dollars on electricity. Electricity energy consumed by datacentres is still growing worldwide. In fact, if all the datacentres were a country, it would rank as the fourth country in the world in terms of electricity consumption [2].

Extremely high energy costs in cloud datacentres are caused by the large number of computing resources and hardware inefficiencies. According to Fig. 1, the data obtained from 5000 servers during a six-month period demonstrate that the servers have usually not been idle. Their productivity, on the other hand, has hardly been high, and they function with 10–50% of their capacity most of the time, which causes the cost-efficiency rate to increase [3].

On the other hand, servers in low-load or zero-load states consume energy more than two-thirds of the time when they are at peak load state; therefore, to keep low-load servers turned on is not economical in terms of energy consumption. Attempts should be made to the extent possible to keep fewer servers turned on through adopting techniques such as virtualization, migration of virtual machines from low-load servers and their consolidation on other servers [4].

In all resource management techniques, according to the concerned objectives, virtual machines are migrated from a source machine to a destination machine. In other words, the virtual machine, along with all its embedded programs (which are

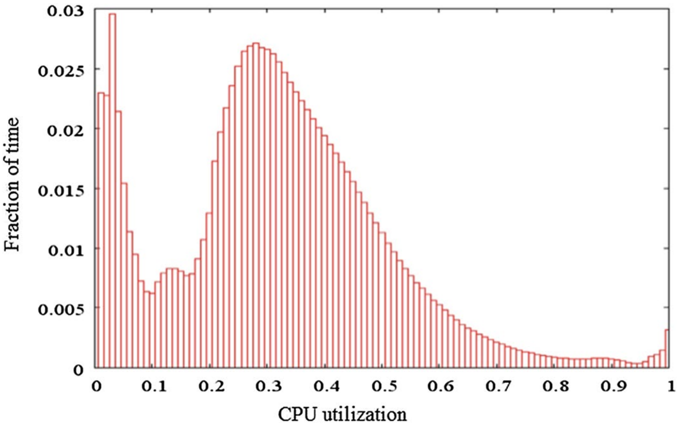

Fig. 1 Average CPU utilization of more than 5000 servers during a 6-month period [3]

running), migrates from one source to another, without interrupting the execution of the programs and making the users aware of such a migration. In general, the migration technique is used for purposes such as balancing and sharing loads, having error tolerance and energy management, reducing response time, improving accessibility, and maintaining servers.

With a virtual machine located in a low-load or high-load host migrating to another host, it is possible to improve energy consumption and utilize resources to deliver the service quality [5, 6].

In this work, a new and more complete model has been proposed along with more detailed investigation of present studies, and better results were obtained through introducing new methods or improvement of algorithms proposed in our previous work [7]. In the proposed model, virtual clusters have been used to cluster servers. In optimization section, the proposed model was much faster and its convergence was proved based on the definition of absorbing state in Markov chain.

One of the points distinguishing the present study from other previous works is that previous works considered the high loading of servers as the basis for making decisions about virtual machine migration. According to our perspective, it needs to be investigated when SLA violation is probable. With defining the critical conditions for servers, the criticality of the condition was regarded as a basis for migration. This would reduce the number of migrations as well as the number of unnecessary migrations. Reducing the number of migrations can play a vital role in decreasing energy consumption, reducing migration overhead, and promoting efficiency.

Based on the details provided in the paper, the innovations of the research can be summarized as follows:

- presentation of a new energy-aware model
- definition of virtual clusters for classification of the physical servers
- definition of the critical conditions of the servers and decision-making about migration time on that basis
- use of a combination of the absorbing Markov chain and the population-based simulated annealing algorithm for optimal placement of the virtual machines within a short time
- use of improved algorithms in the virtual machine selection and consolidation policies.

Table 1 lists some general specifications of the work.

The remainder of this paper is organized as follows: Sect. 2 presents a review of the state of the art in the literature. Section 3 gives the details of the proposed comprehensive model, step by step. This is followed by performance evaluation and experimental results of the algorithms in Sect. 4. Finally, the conclusion and future works are summarized in Sect. 5.

Table 1 General specifications of the proposed work

Specifications	Level used in the paper
Virtualization	Virtualized datacentrs
System resources	Multiple resources (CPU, RAM, network resources)
System type	Heterogeneous
Goal	Reducing energy consumption along with a decrease in SLA violation
Workload	Arbitrary mixed workloads
Architecture	Dynamic VM management system
Mechanism	Simulation (in CloudSim)

2 Related works

Reducing energy consumption in datacentres along with a decrease in SLA violation is one of the main objectives of cloud computing. Virtualization and efficient algorithms for replacing and consolidating virtual machines are of essence to reach this goal.

Several studies have been carried out in this filed, some of which are addressed in this section. The main specifications of some other studies are also summarized in Table 2.

In [8], an adaptive fuzzy threshold-based algorithm was proposed to detect high- and low-load hosts. The proposed algorithm dynamically develops rules and updates membership functions to be adapted to workload variations. It also collects some information about datacentre servers and applies to a fuzzy inference engine in order to make decisions about high and low loads based on the defined set of rules.

Nadjar et al. [9] classified the placement strategies of virtual machines in the datacentre into five categories (namely network-aware, CPU-based, memory-aware, application-aware and load distribution-based strategies), according to which some hybrid placement strategies were proposed. Regarding the selection of virtual machines to migrate from high-load servers, four other algorithms were proposed and tested instead of the commonly used minimum migration time algorithm, in which the highest deviation in the resource records requested by the virtual machine, minimum SLA violations, the lowest CPU utilization, and the highest CPU efficiency were concerned for making decisions. Furthermore, the load distribution of the physical hosts was also considered for decisions made for the migrated virtual machines in the placement phase. With regard to the findings, a process based on the minimum CPU utilization in virtual machine selection for migration along with a method based on the load distribution of the physical hosts in the placement phase brought about the best results. In this study, fixed threshold values were used to detect high or low loads.

In [10], a virtual machine placement procedure was proposed according to the simulated annealing algorithm. Furthermore, a dynamic scheduling model was suggested for the virtual machines, and a virtual machine placement model was then proposed using an improved simulated annealing algorithm. In this model,

Table 2 The main specifications of some other studies

References	Objectives	Studies dimension	Descriptions
[17]	To reduce energy consumption and to increase resource efficiency	CPU, network, storage	The study uses the genetic algorithm to place virtual machines. Complicated computations and slowness are two shortcomings in this research
[18]	To minimize turned-on hosts and to avoid the formation of focal points (bottlenecks) in datacentr network	CPU	The study considers network traffic but not the energy consumption of cooling systems, and it ignores migration-related overheads as well as efficiency
[19]	To reduce energy consumption by considering coolers' consumption	CPU, RAM, disk, memory	Migration-related overheads are not considered
[20]	To reduce energy consumption and to decrease SLA violations	CPU, memory, network	The study considers network traffic but not the energy consumption of cooling systems, and it ignores migration-related overheads
[21]	To reduce energy consumption and to distribute load	CPU, memory, network	It is based on Bin-Packing and ignores SLA violations
[22]	To develop a parallel distributed infrastructure to reduce energy consumption	CPU	
[5]	To provide a dynamic algorithm for placement and consolidation of virtual machines	CPU	It examines delays in activating turned-off servers
[23]	To reduce energy consumption based on network traffic	CPU, network	Large number of migrations and the resulting overheads are the weaknesses of the proposed method
[24]	To provide an energy-aware model	CPU	Resource allocation is based on Bin-Packing
[25]	Optimize resource utilization	CPU	Use mathematical modeling of peak similarity to measure similarity of VMs' peak workload and avoid VMs with high correlation for better VM placement and consolidation
[26]	The main goal of the proposed algorithm is to maximize PMs utilization	CPU, memory	A placement algorithm is proposed based on a fitness function that considers the utilization of both the PMs and the VMs. The Best Fit placement algorithm is used along with the fitness function to evaluate the selections of both PMs and VMs

there were some modifications in the sampling phase and temperature reduction in the simulated annealing, in comparison with the basic algorithm.

Ferdaus et al. [11] addressed network-aware placement of virtual components of multi-tier applications in datacentres and formally defined the placement as an optimization problem. The simultaneous placement of virtual machines and data blocks aims at reducing the network overhead of the datacentre network infrastructure. A greedy heuristic is proposed for the on-demand application components placement that localizes network traffic in the datacentre interconnect. Such optimization helps reducing communication overhead in upper layer network switches that will eventually reduce the overall traffic volume across the datacentre. This, in turn, will help reducing packet transmission delay, increasing network performance, and minimizing the energy consumption of network components.

In [12], an energy-aware cloud-based model called the Green Cloud Scheduling Model (GCSM) was proposed. In this model, cloud nodes were assumed to be heterogeneous, and the energy-aware capability of task assignment and scheduling decisions was considered to be in nodes. The capability of nodes to complete tasks was also perceived according to the user-defined constraints. In other words, the GCSM, in addition to reducing energy consumption, seeks to meet the service quality requested by users through completing real-time tasks within a due time.

Aryania et al. [13] proposed a new algorithm based on the ant colony system for solving the virtual machine placement problem at datacentrs. One of the topics covered in this research was the consumed amount of energy for VM migration. With this amount of energy being extremely difficult to calculate, the common practice is to estimate its value based on the size of the VMs being migrated. In the proposed algorithm, the physical machines were classified, based on their loads, into four classes: normal, low-load, high-load, predicted high-load servers. Accordingly, no migration could originate from a normal server, with all migrations leading to only normal or low-load servers rather than the high-load or predicted high-load servers.

Mohiuddin et al. [14] presented a methodology for concentrating VMs based on balanced work load distribution. Depending on the resource capacities, the servers were classified under four classes, and each VM was mapped to one of the mentioned classes based on its demand for resources. The classification was based on processing resources, network bandwidth, and memory of the machines. Accordingly, Class 1 and Class 4 enjoyed the largest and smallest assets of resources, respectively. In order to minimize the number of powered-on servers, the migration was performed in such a way to minimize the migration cost while distributing the work load evenly between the powered-on servers. The migrations were performed only between servers of identical classes or, if not possible, from a higher-class server to a lower-class server.

In [15], the VM migration cost and remaining runtime were acknowledged as two important factors that were rarely regarded. In the proposed algorithm, the entire deal of time was taken as a set of time intervals, with the VMs whose remaining runtimes were shorter than a particular interval not allowed to migrate to avoid unnecessary migrations. Moreover, no VM was allowed to have more than one time intervals violating the SLA. The VM cost was calculated using a weighted function of normalized migration time parameters, inactivity time, and required energy for migration.

In [16], authors propose an energy-aware dynamic virtual machine consolidation method that migrates virtual machines while satisfying constraints on the probabilities of multiple types of resources being overloaded. In their method, a series of algorithms for selecting and placing virtual machines to be migrated are utilized, with constraints on the probabilities of various resources in a physical machine being overloaded. Their algorithms integrate and cooperate similarly to artificial bee colony foraging behaviour to perform an optimized search for the mapping relation between virtual machines and physical machines for consolidation.

3 Proposed model

Virtualization is a technique that, in addition hiding the resources' physical properties, makes possible the users' resources access. This technique creates the possibility of simultaneous separating or sharing computer resources between several different environments, which are called virtual machines, so that these virtual machines either can interact with each other or without awareness of each other. In other words, virtualization is the process of allocating virtual resources and managing them to different services, so that applications can use provided virtual resources in the real-world context. Using virtualization is considered as a complementary technique in many algorithms to reduce energy consumption.

Migration of virtual machines, from an under-load or overload physical machine to another physical machine, is one of the techniques used to save energy and utilize resources [19, 24].

In this paper, in order to improve the resource management process and reduce energy consumption, a comprehensive model is presented as Fig. 2, which the

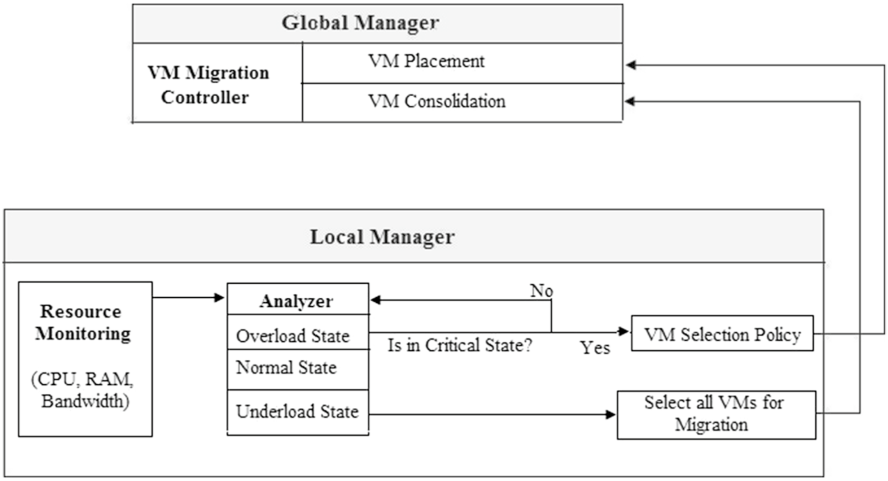

Fig. 2 The proposed model architecture

algorithms used in it are sequel discussed in more detail. In this model, management process is divided into two parts: local and global managers. These two parts are related to each other, and the local manager sends information related to the condition of the host to the global manager at specified intervals. The global manager has information about all hosts and uses them in the migration decisions required to placement and consolidation of virtual machines, which require global visibility from the datacentre.

The local manager is located on the hosts and, as it is clear from Fig. 2, has different sections that interact with each other. The resource monitoring section monitors the state of resources, such as processing resources, memory, and bandwidth and gave them to the analysis section. Analysis section, according to algorithms, determines the state of local machines in terms of over, average and under-loads. In the case of under-loading, all virtual machines are selected for migration, and in coordination with the global manager, necessary measures are performed to consolidation. In the case of overloading, the critical state of machines is examined, and if confirmed, based on the policy of selection part of virtual machines, the global manager will be informed, then some virtual machines are selected for migration so that the system exits from the critical state.

3.1 Recognizing the critical condition of physical machines

According to the performed research, any live migration can consume up to 10% of processor utilization. In addition, if there is no proper migration policy, we may have to switch some servers from Sleep to ON state, which results in high energy costs.

Therefore, in cloud computing centres with thousands of hosts, an unnecessary migration disrupts the overall system equilibrium and negatively affects the performance of running programs [27].

Due to the heterogeneity of systems in the datacentre, considering a constant value, as a threshold for overload, cannot be appropriate. For example, there is less probability that a host with more number of cores goes into overload state by adding a virtual machine, but at the same circumstances, a host with a less number of cores more likely goes into overload state by adding this virtual machine. It is why overloading of each machine should be considered according to its particular conditions. On the other hand, the processor utilization level is not the only important factor, and the amount of main memory fullness should be addressed as another important factor. It is possible that a machine, in terms of processor utilization, is in a good condition, but a large amount of its memory is full, so, in this case, the machine should be considered as an overloaded system, as it may lead to an SLA violation.

Therefore, in the analyser, the processor and memory status is monitoring together. Given the overloading of the processor, and its single or multi-core state, as well as the amount of filled memory, the eleventh states are defined as Table 3.

The local regression(LR) algorithm is used to detect whether the CPU is overloaded. If the processor load is under 20%, it is considered under-load. In the case of the main memory, if more than 80% of the memory is full, the amount of used

Table 3 Definition of virtual clusters based on processor and main memory status

Using memory	Single or multi-core	Processor status	Virtual clusters
Low	Single or multi-core	Under-load	C1
Average	Single or multi-core	Under-load	C2
Low	Single or multi-core	Average load	C3
Average	Single or multi-core	Average load	C4
High	Single or multi-core	Under-load	C5
High	Single or multi-core	Average load	C6
Low	Single core	Overload	C7
Low	Multi-core	Overload	C8
Average	Single core	Overload	C9
Average	Multi-core	Overload	C10
High	Single or multi-core	Overload	C11

memory is considered high, and if memory is filled less than 20%, we consider its use low.

Therefore, according to Table 3, C7, C9, and C11 clusters are considered clusters with critical machines.

3.2 Selecting virtual machine for migration

An important factor to choose a virtual machine to migration is migration time.

Migration time, according to Eq. 1, depends on both the amount of using main memory and bandwidth. Hence, the lower use of main memory by the virtual machine leads to faster migration. The shorter time migration leads to the lower cost of the migration, according to Eq. 2. Another factor that many studies have paid less attention to it is the amount of VM's CPU usage. The migration at a short time, which results in the release of a greater percentage of processors, is more suitable.

$$T_{mi} = \frac{M_i}{B_i} \tag{1}$$

$$U_{di} = 0.1 \int_{0}^{t_0+T_{mi}} U_i(t)\mathrm{d}t \tag{2}$$

M_i is the amount of memory used by VM$_i$, U_{di} is the total performance degradation by VM$_i$, t_0 is the initial time of migration, T_{mi} is the time taken to complete the migration, $U_i(t)$ is the CPU utilization by VM$_i$, and B_i is the available network bandwidth.

According to the mentioned points, the basis of choosing a virtual machine to migration from critical servers is migration index. This index is determined based on the ratio of using processor to the occupied main memory by each virtual machine, according to Eq. 3.

$$\mathrm{MI} = \frac{U_c}{M} \tag{3}$$

The algorithm of this section is presented below.

- Set a list of critical servers of C7, C9, and C11 clusters.
- Set a list of virtual machines for each critical server, and assign each one a migration index according to Eq. 3.
- Arrange the list of virtual machines in descending order, based on the migration index. If this parameter was equal to some virtual machines, arrange based on less use of main memory.
- Take out the first virtual machine from the list, and if it excludes the server from the critical condition, select it for migration, and put it on the migration list. This takes out the server from the critical state; however, it remains close to the threshold.
- If the server is still in the critical state, then we have to choose the next virtual machine from the list and transfer it to the migration list. This work is repeated so that the server exits from a critical state.

3.3 Virtual machine placement policy

In this policy, to replace the selected virtual machines to migration, it is indicated that migration should not overload the destination. Because such migration has two major disadvantages: firstly, it increases the possibility of SLA violation at the destination and secondly, increases the possibility of another migration in the destination, which in turn needs energy consumption. Therefore, the introduced virtual cluster in Table 3 has been used, and their application is explained in the steps of the algorithm.

This policy has used a population-based simulated annealing algorithm, which is a parallel run of this algorithm. According to the function of simulated annealing algorithm, and the outputs of the fitness function (Eq. 3) at any temperature, and their adaptation to the problem conditions, which indicates they are Markov-based, the following strategies are presented.

1. To develop the initial population, there should be several lists of servers. This list should not include overload, under-load, and off-line servers. To prepare these lists, the available servers in the virtual clusters C3, C4, C5, C6, C8 and C10 are used according to Table 3. In these lists, servers are arranged in ascending order based on the amount of using the processor.

   Then, we create a list of chosen virtual machines to migration. (The number of members in both lists of servers and virtual machines should be equal; otherwise, the primary servers of the list repeat due to under-loading.)

   Several randomized arrays are selected from this server list and are considered as the initial population. Then, this initial population (lists) are evaluated. The normalized valuation criterion is the total increase in the listed servers' consumed energy, in which, the lower, the better.

$$F = \frac{\sum_{i=1}^{n} \acute{E}_i - \sum_{i=1}^{n} E_i}{\sum_{i=1}^{n} \acute{E}_i} \tag{4}$$

In this equation, E is the energy before assigning virtual machines and E' is the energy after their assign.

It is obvious that the total resources assigned to virtual machines should not be greater than the total assignable resources to the server.

2. Determining the best answer, we find the best list from the list sets and keep it.
3. Setting the initial temperature $T = T_0$
4. Repeating steps 5–8 until reaching absorption state or the number of steps seems sufficient (internal loop of the algorithm).
5. For each member of the population, a certain number of neighbours are produced and evaluated. Providing the neighbour in each list is through certain methods. To this end, methods such as inserting, inverting, replacing, etc., can be used.
6. List neighbours are sorted in descending order based on the evaluation criterion, which is normalized of increasing the consumed energy, and among them, the best members win, the same number of main populations.
7. According to SA law, each member of the main population compares with a member of the winning neighbours' population. (If it was better, it would be accepted; otherwise, it might be accepted with a possible probability.)

   As we know, in the simulated annealing algorithm, if $\Delta f < 0$, a new change is accepted, but if $\Delta f > 0$, the new change is accepted or rejected based on the Metropolis algorithm. The Metropolis algorithm is such that a random number R is chosen from the normal distribution in the interval of 0 to 1, if $\exp(-\Delta f/T) > R$, the new change is accepted; otherwise, it is rejected.

   In fact, according to the above, it can be said that by repeating the production of the answers and accepting them by using the Metropolis rule, there is a sequence of answers, which forms the Markov chain. The ending of this chain, with a limited and sufficient length, can be similar to a state, in which a physical system at a given temperature reaches a thermodynamic equilibrium.

   Since the state of the system does not depend on previous states, and only depends on the current state, the Markov chain conditions are established, and the studied system is a Discrete Time Markov Chain.

   In this case, we define the state of space S according to the values that the fitness function F can have. This space consists of twenty states, with an equal length of the interval (0–1). Hence, the first (0.95–1) and the last states (0–0.05) have been defined (because we are looking to reduce the amount of energy consumption).

   In other words, at this stage, the states of each temperature are modelled by a Markov chain. Markov chain length is considered equivalent to the number of repeat steps at a given temperature. Since in the final stages of the problem solving, in which the algorithm approaches the minimum point, many of the proposed changes are rejected and the admission rate decreases; therefore, the length of the chains may infinitely increase. To prevent such a problem, the concept of the absorbing state in the Markov chain is used.

The ‘i’ is called absorbing state if, by entering this state, it is impossible to exit. Considering the nature of the problem, the state (0–0.05), which is the best one, is considered as the absorption state, and achieving it, we finish repeating at that temperature. In fact, the condition of ending the repetition at any temperature is reaching the absorbing state, or the constant number of repetition at that temperature. Of course, it can be proved that this absorbing Markov chain finally reaches the absorption state, but to high speed implementing the algorithm, if we do not reach the absorption state after a certain number of attempts, we will end the repetition at that temperature.

Theorem *The defined absorbing Markov chain eventually enters the absorbing state.*

Proof Given the nature of the problem, and created Markov chain at each temperature, the Ergodicity of the chain is obvious, because there is the probability of going from one state to another (not necessarily in a single move). Assume that the starting point is a state such as S_i where ‘i’ can be each of defined 20 states, then the problem would be solved in the state of (0–0.05). Otherwise, the state of (0–0.05) would be achieved after passing n step with the probability of p. Otherwise, other states would be achieved with the probability of 1–p. The probability of not to achieving absorption state after n time is less than or equal to $(1-p)^n$, where the possibility tends towards zero with an increase in n and eventually the absorption state is achieved.

8. The best answer so far has been updated. If this answer within the specified interval is as an absorbing state, end repetition at this temperature, and if the termination conditions of the algorithm are not met, the temperature is reduced based on Eq. 7, and start from step 4.

$$T_{K+1} = \alpha \times T_K \tag{5}$$

   α is a constant parameter, whose value is arbitrarily chosen from 0.8 to 0.99. The performed studies show that when the temperature is high, α can be small to increase the computational speed, but as the calculation progresses and the temperature decreases, α value should be large to ensure the convergence of the method.
9. End.

3.4 Selecting under-load machines

At this stage, the defined clusters in Sect. 3.1 are used. The physical machines, which are located in under-load clusters, are in priority of shutting down. All virtual machines, respectively, migrate from cluster servers C1 and C2 to other cluster servers. In fact, several points are considered in these migrations. For example, the destination should not include C7, C9, and C11 clusters’ machines.

Table 4 The specifications of the simulation environment

Servers	Number of single cores	Number of dual cores	CPU frequency of each core (MHz)	Memory (GB)
HP ProLiant ML110 G4	100	300	1860	4096
HP ProLiant ML110 G5	100	300	2660	4096

Table 5 The three VM types used in our experiments

VM types	Cores	Capacity (MIPs)	RAM (MB)	Storage (GB)	Bandwidth (Mbit/s)
Large	1	2000	1536	2	1000
Medium	1	1500	768	2	1000
Small	1	1000	512	2	1000

4 Performance evaluation

In this work, we choose simulations to evaluate the algorithms. CloudSim has been chosen in our evaluations. For experiments, the data provided as a part of the CoMon project, a monitoring infrastructure for PlanetLab [27], were used. These values are also found in the CloudSim simulator in the examples/workload/planetlab path, which we used the case 20110303.

Table 4 shows the specifications of the simulation environment. Each server has a bandwidth of 1 Gbps, and half of the bandwidth has been considered in the simulations made for migration and the other half for VM communication. The simulation period has been considered to be 86,400 s.

In simulations, we used the three types of Virtual Machines as shown in Table 5.

Actual workload has been used in the scenarios so that the results are reliable. It is a part of CoMon project [28] that has been collected from over a thousand VMs from the servers located in over 500 places around the world. Table 4 of [29] gives a brief overview of the workload used in our experiments.

4.1 Performance evaluation metrics

For performance evaluation of the algorithms, some metrics were used. In this section, we define them.

OTF: The fraction of time during which active servers have experienced the CPU utilization of 100%.

PDM: The total performance reduction occurring as a result of virtual machines migrations.

SLAV: OTF and PDM are two metrics for measuring the level of SLA violations.

$$\mathrm{OTF} = \frac{1}{N} \sum_{i=1}^{N} \frac{T_{si}}{T_{ai}} \tag{6}$$

$$\mathrm{PDM} = \frac{1}{M} \sum_{j=1}^{M} \frac{C_{dj}}{C_{rj}} \tag{7}$$

In Eqs. 6 and 7; N denotes the number of serves, T_{si} shows the total time during which 100% utilization of server i leads to SLA violation, and T_{ai} is the total time in which server i is in the active mode. Moreover, M stands for the number of virtual machines, C_{dj} shows the estimated decrease in performance of virtual machine j as a result of migration, and C_{rj} refers to the total processor capacity requested by VM_j during its lifetime. In this work, $C\ dj$ is assumed to be equal to 10% of the processor capacity during all migrations of VM_j. The OTF and PDM criteria determine the SLA violation level, separately. Therefore, a hybrid metric, which considers decreased performance with regard to the overload of server and migration of virtual machines, is proposed and is called SLA violation (SLAV). This metric is calculated as shown in Eq. 8.

$$\mathrm{SLAV} = \mathrm{OTF} \times \mathrm{PDM} \tag{8}$$

Energy: The total energy consumption in cloud datacentre.

Migrations: The total number of migrations performed in N servers in the datacentre.

ESV: This metric is proposed to combine the two parameters Energy and SLAV. The main objective of resource management in cloud-based datacentrs is to minimize both energy cost and SLA violations. Therefore, we consider a combined metric in Eq. 9 to take into account both energy cost and the level of SLA violations.

$$\mathrm{ESV} = \mathrm{Energy} \times \mathrm{SLAV} \tag{9}$$

4.2 Simulation results

In the proposed algorithm, the LR is used to determine overload. Therefore, in simulation, the combination of LR-MMC and LR-MMT has been used to compare with the proposed model.

The specifications of the simulation and details on the physical and virtual machines are presented in Tables 4 and 5. The comparison between the algorithms has been performed in overload, average load, and under-load. This comparison is based on standard metrics, which are defined in Sect. 4.1.

In the simulations, the proposed algorithm has a considerable advantage over others, and its performance in all comparison metrics has been better than other algorithms. The simulation results, to further study, are discussed in Figs. 3, 4, 5, 6, 7 and 8.

Considering the improvements in the proposed algorithm, as Fig. 3 implies, energy consumption has significantly decreased. These reductions at low load, average load, and high load have been 14.3%, 19% and 21%, respectively. Preventing unnecessary migrations, along with shutting down more under-load servers have played an important role to achieve this goal. In a high load, for instance, the average number of server migration to Sleep state in this algorithm has been 4107 times, and this number in other ones is 3951 and 3938 times. The initial suitable arrangement of virtual machines and their consolidation over specific periods, which are according to the placement strategies introduced in Sect. 3.3, has played a significant role to reduce energy consumption.

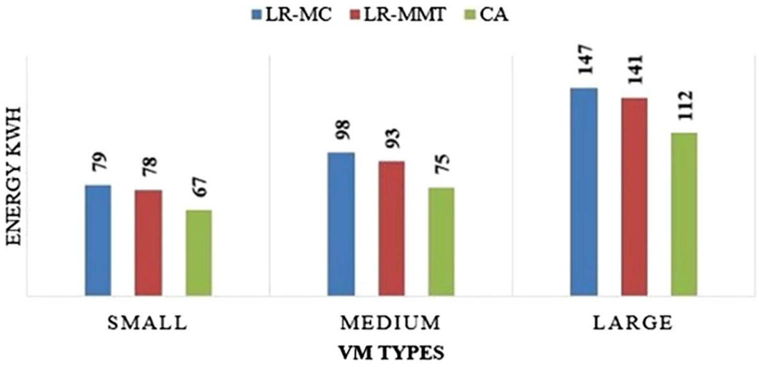

Fig. 3 The energy consumption of algorithms

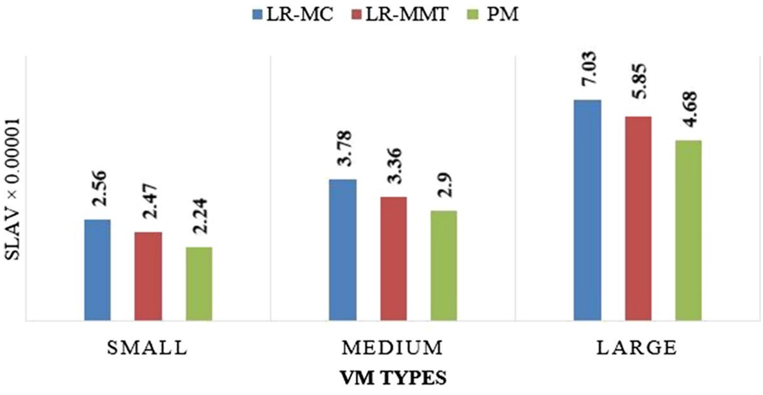

Fig. 4 The SLA violation evaluation of algorithms

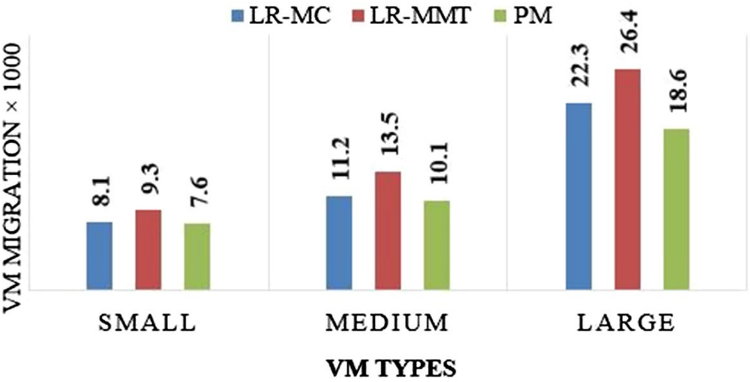

Fig. 5 The migration evaluation of algorithms

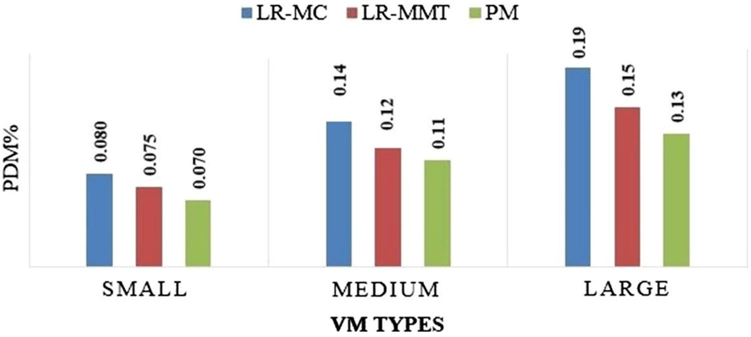

Fig. 6 The PDM evaluation of algorithms

In many studies on reduction of energy consumption, the low increase in SLA

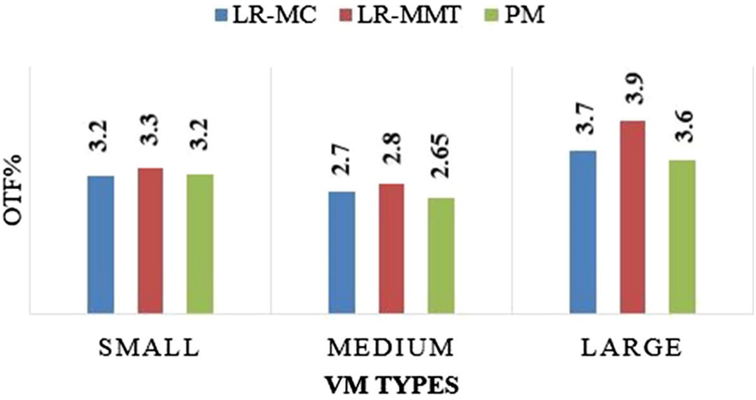

Fig. 7 The OTF evaluation of algorithms

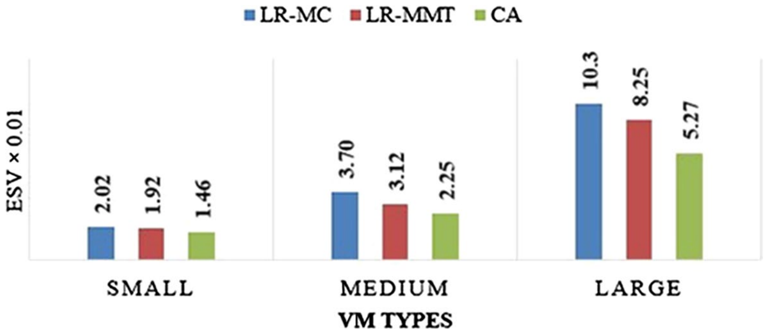

Fig. 8 The ESV evaluation of algorithms

violations is considered acceptable. However, considering the important points of the proposed algorithm, such as the reduction in unnecessary migrations, on the one hand, and considering the processor and memory conditions, on the other hand, have resulted in the reduction in the SLA violation compared to other algorithms. Figure 4 shows the performance of compared algorithms in this field in which the proposed algorithm compared to the best ones, at high, average and low loads have had 10%, 13.6% and 20% reduction, respectively.

The proposed approach has prevented unnecessary migration, and in the case of overload, criticality of the server is also examined, and, if necessary, migration will be performed based on processing ability and the amount of memory. The definition of virtual clusters and the servers' membership in these clusters, given the state of the processor and memory along with a proper definition of regulations, have led to timely and targeted migration. As a result, the number of migration has dramatically reduced as shown in Fig. 5. This decline is considerable in high load and is about 17% of course, and this decline also on average load and low load has been 11% and 7%, respectively.

Another reason for declining the number of migrations is by using VM placement policies. Applying policy that prolongs the interval between two migrations, as possible as, is particularly important. In other words, the choice of migration destination should not be such that in the near future, lead to another migration, or lead to situations to inevitable migration. Markov chain-based placement policy, as described in Sect. 3.3, due to considered points tries to find a proper arrangement of virtual machines on physical machines to reduce unnecessary migrations.

According to Eq. 2, the increase in VM migrations can increase performance degradation. Therefore, reducing unnecessary migrations is needed. Due to the reduction in migrations in the proposed algorithm, its performance degradation is also in a better status than the other algorithms and its improvement compared to the second algorithm is 10% (Fig. 6).

In Fig. 7, the percentage of cases when the CPU is overload is compared. Although in all three algorithms, similar LR policies have been used to recognize overload, adopting appropriate policies has caused better performance of the proposed algorithm than other two ones that are 1.8 and 2%, respectively.

In Fig. 8, the algorithms are compared in terms of the ESV parameter. As it is mentioned in Eq. 9, the ESV parameter is directly related to energy consumption and SLA violation. The reduction in SLA violations and energy consumption compared to the other two algorithms has caused also the better performance of the proposed algorithm in this case. The proposed algorithm, in comparison with the best-compared algorithm, is better, 24%, 27% and 35% in low load, average load, and high load.

5 Conclusions and future works

In the present study, a comprehensive model was proposed to reduce energy consumption with regard to service-level agreements. In order to improve the resource management process and reduce energy consumption through breaking down the main problem into smaller sub-problems, either a new algorithm was introduced for each sub-problem or previous algorithms were improved. In the proposed model, all phases were run in a distributed manner; however, the phases were centralized when replacing a virtual machine requiring a global perspective. First, the algorithm examined the critical state of machines using an algorithm that determined a high-load host with specific conditions among other active hosts. Then a virtual machine selection algorithm was used to select the virtual machines to migrate from critical hosts. After preparing a list of selected machines for migration using the virtual machine selection algorithm, a new host was considered as the destination for the migrating virtual machines. In this phase, Markov chain along with the simulated annealing meta-heuristic algorithm was employed. Finally, low-load hosts were selected using a low-load host selection algorithm to be hibernated when all virtual machines completely migrated. For future studies, the model presented in this study can be further complemented from several other aspects. Future works can also focus on the following novel aspects of the present study:

- Research has confirmed that a high percentage of traffic in a datacentre is associated with its internal traffic and communication among programs [30]; therefore, it is of paramount importance to use network-aware policy that is to place virtual machines with higher communication traffic on adjacent physical servers through examining the relationships among different virtual machines. To do so, clustering and placing virtual machines linked together in a cluster or in adjacent clusters might work. Due to the reduced use of network equipment such as switches in this case, energy consumption further reduces.
- With understanding the behaviour and pattern of using resources in different applications, further attempts should be made to the extent possible to avoid placing applications with similar requests for resources on a server as this may lead to a competition for capturing such specific resources, resulting in further SLA violation.
- Update period and how information is exchanged between local and global managers would also be addressed in future studies. A dynamic algorithm with high prediction accuracy can significantly affect the efficiency.

Acknowledgements This work sponsored by Islamic Azad University Science and Research Branch.

References

1. Beloglazov A (2013) Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing. PhD Thesis, Melbourne University
2. Ahmad F, Vijaykumar T (2010) Joint optimization of idle and cooling power in data centers while maintaining response time. ACM SIGPLAN Notices 45(3):243–256
3. Barroso LA, Holzle U (2007) The case for energy-proportional computing. Computer 40(12):33–37
4. Quan DM, Mezza F, Sannenli D, Giafreda R (2013) T-Alloc: a practical energy efficient resource allocation algorithm for traditional data centers. Future Gener Comput Syst 28(5):791–800
5. Kumar MVR, Raghunathan S (2016) Heterogeneity and thermal aware adaptive heuristics for energy efficient consolidation of virtual machines in Infrastructure clouds. J Comput Syst Sci 82(2):191–212
6. Zhao DM, Zhou JT, Li K (2019) An Energy-Aware Algorithm for Virtual Machine Placement in Cloud Computing. IEEE. https://doi.org/10.1109/ACCESS.2019.2913175
7. Rajabzadeh M, Haghighat AT (2017) Energy-aware framework with Markov chain-based parallel simulated annealing algorithm for dynamic management of virtual machines in cloud data centers. J Supercomput 73(5):2001–2017
8. Salimian L, Esfahani FS, Shahraki MN (2016) An adaptive fuzzy threshold-based approach for energy and performance efficient consolidation of virtual machines. Computing 98(6):641–660
9. Nadjar A, Abrishami S, Deldari H (2017) Load dispersion-aware VM placement in favor of energy-performance tradeoff. J Supercomput 16(4):112–127
10. Su N, Shi A, Chen CH (2016) Research on virtual machine placement in the cloud based on improved simulated annealing algorithm. In: IEEE World Automation Congress (WAC), USA, pp 23–32
11. Ferdaus MH, Murshed M, Calheiros RN, Buyya R (2017) An algorithm for network and data-aware placement of multi-tier applications in cloud data centers. J Netw Comput Appl 98(2):65–83
12. Kaur T, Chana I (2016) Energy aware scheduling of deadline-constrained tasks in cloud computing. Clust Comput 19(5):66–75
13. Aryania A, Aghdasi HS, Khanli LM (2018) Energy-aware virtual machine consolidation algorithm based on ant colony system. J Grid Comput. https://doi.org/10.1007/s10723-018-9428-4
14. Mohiuddin I, Almogren A (2018) Workload aware VM consolidation method in edge/cloud computing for IoT applications. J Parallel Distrib Comput. https://doi.org/10.1016/j.jpdc.2018.09.011
15. Heyang X, Yang L, Wei W, Ying X (2019) Migration cost and energy-aware virtual machine consolidation under cloud environments considering remaining runtime. Int J Parallel Program. https://doi.org/10.1007/s10766-018-00622-x
16. Zhihua L, Chengyu Y, Lei Y, Xinrong Y (2019) Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic consolidation method. Future Gener Comput Syst 80(3):139–156
17. Gao Y, Guan H, Qi Z, Hou Y, Liu L (2013) A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. J Comput Syst Sci 7(8):1230–1242
18. Kliazovich D, Bouvry P, Khan SU (2013) DENS: data center energy efficient network-aware scheduling. Clust Comput 16(1):65–75
19. Deng W, Liu F, Jin H, Liao X, Liu H (2014) Reliability-aware server consolidation for balancing energy-lifetime tradeoff in virtualized cloud datacenters. Int J Commun Syst 27(4):623–642
20. Garg SK, Toosi AN, Gopalaiyengar SK, Buyya R (2014) SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter. J Netw Comput Appl 45(6):108–120
21. Song W, Xiao Z, Chen Q, Luo H (2015) Adaptive resource provisioning for the cloud using online bin packing. IEEE Trans Comput 63(11):2647–2660
22. Rethinagiri SK, Palomar O, Sobe A, Yalcin G, Knauth T, Gil RT, Prieto P, Schneega M, Cristal A, Unsal O (2016) ParaDIME: parallel distributed infrastructure for minimization of energy for data centers. Microprocess Microsyst 39(8):1174–1189
23. Dong J, Wang H, Cheng S (2015) Energy-performance tradeoffs in IaaS cloud with virtual machine scheduling. Communications 12(2):155–166

24. Carli T, Henriot S, Cohen J, Tomasik J (2017) A packing problem approach to energy-aware load distribution in clouds. Sustain Comput Inform Syst 9(2):20–32
25. Lin W, Xu S, Li J, Xu L, Peng Z (2017) Design and theoretical analysis of virtual machine placement algorithm based on peak workload characteristics. Soft Comput 21(5):1301–1314
26. Zhang R, Zhong AM, Dong B, Tian F, Li R (2019) Container-VM-PM architecture: a novel architecture for docker container placement. In: International Conference on Cloud Computing. Springer International Publishing, Cham
27. Dhingra A, Paul S (2014) Green cloud: heuristic based BFO technique to optimize resource allocation. Indian J Sci Technol 7(5):685–691
28. Park KS, Pai SV (2006) CoMon: a mostly-scalable monitoring system for planet-lab. ACM SIGOPS Oper Syst Rev 40:65–74
29. 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 13(1):1397–1420
30. Ferdaus MH (2016) Multi-objective Virtual Machine Management in Cloud Data Centers. PhD Thesis, Monash University

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