The Journal of Supercomputing (2020) 76:10240–10257
https://doi.org/10.1007/s11227-020-03248-4

Prediction-based underutilized and destination host selection approaches for energy-efficient dynamic VM consolidation in data centers

Kawsar Haghshenas[1] · Siamak Mohammadi[1]

Published online: 16 March 2020

Abstract

Improving the energy efficiency while guaranteeing quality of services (QoS) is one of the main challenges of efficient resource management of large-scale data centers. Dynamic virtual machine (VM) consolidation is a promising approach that aims to reduce the energy consumption by reallocating VMs to hosts dynamically. Previous works mostly have considered only the current utilization of resources in the dynamic VM consolidation procedure, which imposes unnecessary migrations and host power mode transitions. Moreover, they select the destinations of VM migrations with conservative approaches to keep the service-level agreements , which is not in line with packing VMs on fewer physical hosts. In this paper, we propose a regression-based approach that predicts the resource utilization of the VMs and hosts based on their historical data and uses the predictions in different problems of the whole process. Predicting future utilization provides the opportunity of selecting the host with higher utilization for the destination of a VM migration, which leads to a better VMs placement from the viewpoint of VM consolidation. Results show that our proposed approach reduces the energy consumption of the modeled data center by up to 38% compared to other works in the area, guaranteeing the same QoS. Moreover, the results show a better scalability than all other approaches. Our proposed approach improves the energy efficiency even for the largest simulated benchmarks and takes less than 5% time overhead to execute for a data center with 7600 physical hosts.

Keywords Linear regression · VM consolidation · VM migration · Energy efficiency · Cloud data centers

✉ Siamak Mohammadi
smohamadi@ut.ac.ir

Kawsar Haghshenas
khaghshenas@ut.ac.ir

[1] School of Electrical and Computer Engineering, University of Tehran, Tehran 17469-37181, Iran

1 Introduction

High energy consumption has become one of the critical issues for today's data centers. High operational cost and environmental effects are two main consequences of this usage. According to Koomey's report [1], estimated annual electricity cost in large data centers is about 41% of the data center's total operating costs. However, according to Shehabi's latest report [2], a potential of 45% reduction in electricity usage can be achieved in new generation of sustainable and energy-efficient data centers by new resource management approaches.

Virtualization technology in modern data centers [3] provides the opportunity of consolidating virtual machines (VM) of the data center dynamically. Dynamic VM consolidation techniques utilize live VM migration to pack as many VMs as possible on one physical host and switch idle hosts to low-power modes to decrease hosts' energy consumption [4]. However, given the variable nature of VMs loads [5, 6], and energy and delay overheads associated with VM migrations, dynamic consolidation may degrade the quality of service (QoS) and even increase overall energy consumption, if not effectively applied.

Increasing the workload of some VMs may cause the corresponding physical hosts to be overloaded, but some other hosts to be underutilized [7]. Dynamic VM consolidation helps improve the resource utilization while keeping a satisfactory level of QoS. In this regard, several steps are taken to reallocate VMs to the hosts effectively. Main problems of this procedure are (1) detecting overloaded hosts, (2) selecting migrating VMs of overloaded hosts, (3) detecting underutilized hosts, (4) selecting destination hosts for migrations and (5) performing migrations [8]. Previously published papers in this area have mostly focused on one problem of the whole procedure, while in this paper we will focus on the third and fourth ones.

Mostly, past works in this context have taken the current CPU utilization into account to decide whether a physical host is overloaded or underutilized [9–12], whereas few of them have tried to estimate future utilization or state of the hosts and have taken decisions accordingly [13–16]. Noting that the hosts of a data center experience dynamic and used dependent workloads, the former approaches may cause unnecessary migrations and unnecessary host power mode transitions, which finally decrease the efficiency of the dynamic consolidation process.

As mentioned before, consolidation approaches try to pack more VMs on fewer physical hosts. Therefore, the hosts with higher CPU utilization seem to be more efficient destination candidates for VM migrations. On the other hand, migrating more VMs to the hosts with higher CPU utilization increases the probability of overutilization, and hence, this approach has not been exploited so far. Therefore, concerning QoS requirements, most previous works select destination hosts by other methodologies like selecting based on the additional power usages imposed by new migrated VM or limiting number of active hosts [9, 13, 17–19], which are not in line with packing more VMs on fewer hosts.

Our work tackles the aforementioned challenges of the dynamic consolidation procedure. Our proposed approach uses known linear regression method to predict future resource utilization of all hosts using their historical data. Considering the accuracy of linear regression for prediction, our approach firstly checks the capacity of candidate destinations and then migrates VMs to the hosts with higher resource utilization, which leads to a better placement in the view of consolidation. Furthermore, our approach predicts the utilization of all running VMs and selects the host with minimum predicted load as underutilized host accordingly.

Our solution is evaluated using real traces from enterprise servers. The proposed techniques, as well as our baselines for comparison, have been implemented in the CloudSim simulator. The results show that the proposed algorithm improves energy efficiency while meeting SLA requirements and keeps its efficiency even for larger workload traces while imposing less than 5% computation time overhead (compared to used 5-minute time interval).

The remainder of this paper is organized as follows. Section 2 summarizes the related work. Then, Sect. 3 introduces the considered problem in this paper. Section 4 describes our proposed approaches for the explained problem. Section 4 also represents the basic concepts of linear regression method. The experimental setup is described and results are provided in Sect. 5. Finally, a summary of our conclusions is drawn in Sect. 6.

2 Related work

There has been an extensive research in data center energy efficiency. With the wide acceptance of the virtualization technology, most of the previous approaches focus on VM consolidation methods as an effective solution to save energy in data centers. VM consolidation methods use live VM migration to pack the existing VMs into fewer hosts and switch off the idle hosts periodically [9, 11, 13, 14, 20]. In general, the problem of dynamic VM consolidation can be split into several subproblems. Previous approaches in this regard usually focus on one subproblem of the general process.

In some approaches, the VM consolidation is treated as an optimization problem and solved by known convex optimization solutions. Wu et al. [11] proposed a technique based on an improved grouping genetic algorithm (IGGA), which switches off idle hosts to save energy. Ashraf et al. [21] proposed a multiobjective ant colony system-based algorithm to build VM migration plans, which are used to minimize overprovisioning of physical machines (PMs) by consolidating VMs on underutilized PMs. The main bottleneck of this approach is its large computation time overhead, especially for large-scale data center scenarios.

Beloglazov et al. [9] introduced several dynamic consolidation algorithms which work along with power-aware best-fit decreasing (PABFD) algorithm for destination host selection. The main differences between these algorithms are about their overloaded host detection and VM selection methods. In these algorithms, the overloaded host detection method is based on the host's CPU utilization and VM selection method is based on the VM's characteristics. Sercon [10] is another algorithm which minimizes the total number of active hosts together with the number of migrations. The key idea of Secron is migrating VMs from least loaded hosts to most loaded ones. Given that the CPU usage of the hosts changes arbitrarily based on the users behavior, these approaches fall short in prediction hosts' utilization.

Fahimeh et al. [13] proposed the linear regression-based CPU usage prediction algorithm (LIRCUP) to consolidate VMs dynamically. LIRCUP uses the same methodologies of [9] for different steps of consolidation process. However, it predicts hosts' CPU utilization by linear regression and considers predicted values instead of current CPU utilization to detect overloaded hosts, underutilized hosts and destinations of migrations.

Suhib et al. [14] proposed a VM placement algorithm, which determines the set of candidates hosts to be the destination of migrating VMs. The proposed approach in [14] uses historical data to build probabilistic model that predicts the future host state and then selects the host whose future state is normal as the destination of a migration. Each host in [14] can be in normal, underutilized or overloaded state based on its CPU utilization. EQ-VMC [20] is another VM placement algorithm, which uses an improved discrete differential evolution (discrete DE) algorithm to search for the global optimization solution for VM placement of migrating VMs. This algorithm regards all mappings between VMs and PMs as a population and uses heuristic evolutionary approach to obtain optimal VM placement. Wang et al. [22] proposed HS and SABFD algorithms to select migrating VMs and destination hosts, respectively. Both of these algorithms are based on the current utilization of VMs and hosts.

Saeid et al. [15] proposed an adaptive threshold-based algorithm for overloaded host detection subproblem of dynamic VM consolidation using dynamic fuzzy Q-learning (DFQL) method. The proposed algorithm in [15] learns when a host must be considered as an overloaded host with regard to the energy-performance trade-off optimization.

Abbas et al. [16] proposed an algorithm considering the trade-off between energy consumption and performance. The proposed algorithm in [16] combines the current host utilization and the number of its VMs to predict its future utilization. The host with minimum predicted CPU utilization is considered as underutilized host. This algorithm also uses host utilization and minimum correlation (UMC) method to select destination hosts of migrating VMs. Given that the CPU usage of a host can be predicted based on its historical data, these approaches fall short in proposing a destination host selection method which is in line with consolidating VMs.

Finally, many studies exist that try to improve energy efficiency of a data center. To the best of our knowledge, our proposed linear regression-based approach is the first work that besides predicting CPU utilization, migrates VMs of underutilized hosts to the destinations with maximum predicted utilization, which leads to packing as many VMs as possible on fewer hosts while keeping QoS.

3 Problem formulation

Live VM migration provides the opportunity of improving VMs placement over time and after the initial VM allocation. This procedure is called dynamic consolidation. In contrast to static consolidation which assigns VMs to empty hosts, dynamic VM consolidation allows cloud providers to pack VMs into fewer hosts dynamically and set more hosts into low-power modes with time. Thus, dynamic consolidation improves energy efficiency through the VM migration as the main control knob.

The first objective of our proposed approaches for dynamic consolidation is increasing energy savings. As this procedure may affect QoS, minimizing service-level agreement (SLA) violation is considered as the second objective. We allocate host CPU capacity (measured in million instructions per second, i.e., MIPS) to VMs, up to their current requested MIPS. Therefore, two conflicting objectives here are minimizing energy consumption and the difference between the requested and allocated MIPS to all VMs (which represents SLA violation).

We formulate our problem in Eqs. (2)–(5). Let S_H and S_V represent the set of hosts and VMs of the data center, respectively, with N and M denoting their number of members during the time period (t_1, t_2). At each time slot t, each VM is running on a host that is described by $a_{i_j}(t)$:

$$a_{ij}(t) = \begin{cases} 1 & \text{if } \; VM\ j\ is \text{ assigned to host } i,\ \forall j \in S_V\ \&\ \forall i \in S_H \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Based on the above explanations, we can formulate the optimization problem for the time period (t_1, t_2) as follows:

$$Min \quad E_{DC}(t_1, t_2) + SLA_v(t_1, t_2) \tag{2}$$

Subject to

$$\sum_{i=1}^{N} a_{ij}(t) = 1 \ \ \forall j \in S_V \tag{3}$$

$$\sum_{j=1}^{M} C^j_{\text{Alloc}t} \cdot a_{ij}(t) < C^i_{\max} \ \ \forall i \in S_H \tag{4}$$

$$\sum_{j=1}^{M} mem^j_{\text{Alloc}t} \cdot a_{ij}(t) < \text{mem}^i_{\max} \ \ \forall i \in S_H \tag{5}$$

The first constraint (3) ensures that each VM is running only on one host, and the set of constraints (4) and (5) guarantee that the allocated CPU and memory resources from a host to its VMs are not higher than its capacity. C^i_{max} and $\text{mem}^i_{\max}$ represent maximum CPU and memory capacity of host i. C^j_{Alloct} and $\text{mem}^j_{\text{Alloc}t}$ stand for allocated MIPS and memory to VM j, at time slot t, respectively. The first and second terms of the objective function represent the total energy consumption of the IT equipment ($E_{DC(t_1,t_2)}$) and the average SLA violation ($SLA_{v(t_1,t_2)}$) of the data center for the time period (t_1, t_2). $E_{DC(t_1,t_2)}$ can be defined as:

$$E_{DC(t_1,t_2)} = \sum_{i=1}^{N} [P_i \cdot (t_2 - t_1)] \tag{6}$$

where P_i represents the power consumption of host i during time period (t_1, t_2).

The SLA violation is defined based on the *'difference between requested and allocated MIPS'* of the VMs. Therefore, the second objective of the resource management optimization problem is represented as follows:

$$\mathrm{SLA}_{v(t_1,t_2)} = \sum_{j=1}^{M} (C^j_{\mathrm{Req}} - C^j_{\mathrm{Alloc}}) \cdot (t_2 - t_1) \tag{7}$$

where C^j_{Req} refers to requested MIPS of VM j and C^j_{Alloc} stands for its allocated MIPS during time period (t_1, t_2).

The nature of the VM allocation problem is NP-hard. Therefore, potentially suboptimal solutions are required to obtain reasonable results in as short as possible runtime. In this paper, two linear regression-based approaches are proposed, where further details are provided in the following sections.

4 Proposed approaches

As previously mentioned, five main steps are taken in VM consolidation algorithms: (1) detecting overloaded hosts, (2) selecting migrating VMs of overloaded hosts, (3) selecting underutilized hosts, (4) selecting destination hosts for migrations and (5) performing migrations. Among these steps, we focus on the third and fourth steps of the whole process. Given that the explained problem in Sect. 3 is NP-hard [23], heuristic algorithms are proposed for different steps, to determine reallocation map in a reasonable runtime. Therefore, all the explained constraints in the problem formulation have to be guaranteed in all steps of the proposed approaches. In this section, we first represent the basic formulation of linear regression and then propose two linear regression-based VM consolidation algorithms.

4.1 Linear Regression

Among machine learning techniques, regression is a widely used technique for prediction. Linear regression estimates the linear model between independent variables to a dependent variable. The independent variable is a predictive variable for the dependent variable [24]. The linear regression models the relationship between independent variable x and dependent variable y by:

$$y_k = \beta_0 + \beta_1 x_k \tag{8}$$

where β_0 and β_1 are the regression coefficients. Each y_k is forecasted using its past values. For each predicted model y_k, the estimation of β coefficients is computed to minimize the sum of square error of n past data. The error is the difference between predicted and actual values. The sum of square error for n past data is defined as follows:

$$\sum_{k=1}^{k=n}(\epsilon_k)^2 = \sum_{k=1}^{k=n}(y_k - \hat{y_k})^2 \tag{9}$$

where ϵ_k represents the difference between the kth predicted $(\hat{y_k})$ and actual (y_k) values.

The coefficients are chosen such that the residual sum of squares (RSS) over all past data points is minimized [25]. It can be shown that the minimizing values are:

$$\beta_1 = \frac{\sum_{k=1}^{k=n}(x_k - \overline{x})(y_k - \overline{y})}{\sum_{k=1}^{k=n}(x_k - \overline{x})} \tag{10}$$

$$\beta_0 = \overline{y} - \beta_1\overline{x} \tag{11}$$

where $\overline{x}$ and $\overline{y}$ represent the sample means.

4.2 General VM consolidation procedure

Algorithm 1 reviews the used methods for different steps of dynamic consolidation process in this paper. As mentioned before, we propose linear regression-based methods for Step 3 and Step 4 of Algorithm 1. For other steps, we use simple methods that have been used in the literature.

Algorithm 1, first, detects overloaded hosts using a static threshold heuristic. If the CPU utilization exceeds the considered threshold (Ut) of total capacity, the host is assumed overloaded. Then, some VMs are selected to be migrated from overloaded hosts to resolve their overutilization (Lines 1–7). In this way, a VM with minimum requested MIPS is selected for migration and the VMs are chosen to be migrated, while overutilization of the host is resolved. Actually, the first and second steps prevent high utilization rate for all hosts. In the third step, the destinations for migrating VMs of the overutilized hosts are selected using Algorithm 2 (explained in the following subsections). Finally, the underutilized hosts are selected in Step 4 using Algorithm 3. All the VMs of underutilized hosts are migrated in order to switch the underutilized host to a low-power mode. The destinations of migrating VMs from underutilized hosts are selected with the same method used in Step 3. The following subsections explain the proposed algorithms focusing on Steps 3 and 4 of the whole consolidation procedure.

Algorithm 1 Dynamic Consolidation Procedure

```
Step1:
$OverloadedHosts$ : detect all overloaded hosts using a static threshold.
Step2:
for all hosts in $OverloadedHosts$ do
    while host is overloaded do
        $SelectedVm \leftarrow$ VM with minimum $C^j_{Req}$
        Add $SelectedVm$ to $MigratingVMs$
    end while
end for
Step3:
for all VMs in $MigratingVMs$ do
    $SelectedDestination \leftarrow$ select destination host for migration among all active hosts
    that are not in $OverloadedHosts$, using Algorithm 2.
end for
Step4:
for All active hosts that are not in $OverloadedHosts$ and are not the destination of a
migration do
    $SelectedUnderutilizedHost \leftarrow$ Select the underutilized host using Algorithm 3.
    if All VMs of $SelectedUnderutilizedHost$ are migratable to an active host that is
    not in $OverloadedHosts$ then
        for All VMs of $SelectedUnderutilizedHost$ do
            $SelectedDestination \leftarrow$ Select destination host for migration using Algorithm 2
            among active hosts that are not in $OverloadedHosts$ and have not been selected
            as an underutilized host yet.
        end for
    end if
end for
Step5:
Perform migrations.
```

4.3 Linear regression-based consolidation algorithms

This section describes the proposed approaches for underutilized and destination host selection steps of Algorithm 1. Host CPU utilization has been a key parameter in most of the previously presented dynamic consolidation algorithms. This parameter is used in different parts of the algorithms to select a better VM placement. Concerning that the VMs of a data center experience dynamic workloads and consequently their CPU usage varies arbitrarily over time; the current host CPU utilization alone is not an appropriate indication of future utilization. Thus, disregarding past data of hosts' CPU utilization decreases dynamic consolidation efficiency resulting in lower energy savings, higher VM migrations and consequently higher SLA violations.

In this section, we present two approaches that use linear regression method as a known prediction methodology to predict hosts' CPU utilization for different steps of dynamic consolidation process. One-level linear regression-based approach (OLLRBA) uses linear regression method at one level to select the destination hosts of migrations (using Algorithm 2). This algorithm also selects a host with the minimum CPU utilization as underutilized host.

Two-level linear regression-based approach (TLLRBA) uses linear regression method at two levels to select both the destination hosts of migrations (using Algorithm 2) and underutilized hosts (using Algorithm 3). We consider both OLLRBA and TLLRBA as our proposed approaches in the whole paper.

4.3.1 OLLRBA: one-level linear regression-based algorithm

In the third and fourth steps of dynamic consolidation process (Algorithm 1), destination hosts are selected for migrating VMs. Selecting the host with maximum CPU utilization as the destination is completely in line with the consolidation goal, which is packing as many VMs as possible on fewer hosts (which leads to less active hosts and consequently less energy consumption). This approach has not been used in most of the previous works in the area. This is because the CPU usage of VMs and hosts varies arbitrarily and is user dependent. Thus, selecting the host with maximum CPU utilization as the destination host can cause high SLA violations and even higher energy consumption.

OLLRBA predicts the hosts utilization based on their past data using linear regression and selects the host with maximum predicted CPU utilization as the destination host. OLLRBA also selects the host with minimum CPU utilization as underutilized host in Step 4 of Algorithm 1.

Algorithm 2 reviews the used procedure in OLLRBA to select the destination hosts. This algorithm first predicts the future utilization of all hosts based on their current utilization and using their n past training data. In this way, the CPU utilization of host i at time slot $t + 1$ is predicted by:

$$U_{t+1}^{i} = \beta_0 + \beta_1 * U_t^{i} \tag{12}$$

where β_0 and β_1 are calculated using n host's past data by Eqs. 10 and 11, respectively. In fact, the hosts' past CPU utilization are used as the training data to model the relationship between the current and future CPU utilization.

At the second step of Algorithm 2, for each migrating VM, the host with maximum predicted utilization is selected as the destination host.

Therefore, Algorithm 1 describes OLLRBA if (1) Algorithm 2 is used in Steps 3 and 4 to select the destination hosts and (2) the host with minimum CPU utilization is selected as underutilized host in Step 4. Our simulations results show that OLLRBA achieves higher energy savings while keeping SLA (see Sect. 5).

Algorithm 2 Destination Hosts Selection Procedure

```
 Step1:
for all hosts do
    Predict host's CPU utilization using its current utilization by Equation 12
end for
 Step2:
for all VMs in $MigratingVMs$ do
    for all candidate destination hosts do
        if host's $predictedUtil > maxUtil$ then
            $destinationHost \leftarrow$ Select the host as the destination host.
        end if
    end for
end for
```

4.3.2 TLLRBA: Two-level linear regression-based algorithm

TLLRBA selects the destination hosts with the same methodology as OLLRBA approach. The difference between these two approaches is about underutilized hosts selection. OLLRBA selects the host with the minimum CPU utilization as the underutilized host and continues selecting underutilized hosts until all the VMs of selected host can be migrated. As migrating VMs imposes energy and delay overheads, selecting inappropriate new VMs placement decreases the efficiency of the consolidation process. Therefore, TLLRBA uses linear regression algorithm at two levels. Besides predicting hosts' CPU utilization (using Eq. 12) to select destination hosts, it predicts hosts' VMs utilization to select underutilized hosts. Similar to the host CPU utilization prediction, the utilization of VM j at time slot $t+1$ is predicted by:

$$U^j_{t+1} = \beta_0 + \beta_1 * U^j_t \tag{13}$$

where β_0 and β_1 are calculated using n VM's past data by Eqs. 10 and 11, respectively. After predicting the utilization of all running VMs of all hosts, the host with minimum load (based on its VM's predicted utilization) is selected as underutilized host.

Algorithm 3 represents explained procedure of selecting underutilized hosts used in TLLRBA. At the first step of this algorithm, all active hosts that are not overloaded and are not the destination of migrations are considered as the underutilized candidates. Then, the utilization of all the VMs of candidate hosts for the next time slot is predicted. Finally, the host with minimum future utilization (based on its running VMs) is considered as underutilized. Lines 5–11 of Algorithm 3 are repeated until all VMs of the last selected underutilized host cannot be migrated (i.e., there are not host destinations for all of its VMs). As explained above, this algorithm is run inside Algorithm 1 in order to select underutilized hosts. Therefore, Algorithm 1 describes TLLRBA if (1) Algorithm 2 is used in Steps 3 and 4 to select the destination hosts and (2) Algorithm 3 is used to select the underutilized hosts in Step 4.

Algorithm 3 Underutilized Host Selection Procedure

```
Step1:
candidateHosts: all active hosts that are not overloaded and are not the destination of a migration.
Step2:
for all VMs running on candidateHosts do
    Predict VM's utilization using its current utilization by Equation 13
end for
Step3:
while all VMs of selected underutilized host can be migrated do
    for all candidateHosts do
        predictedHostUtil ← Sum up all its VMs' predictedVMUtil
        if predictedHostUtil < minUtil then
            underUtilizedHost ← Select the host as the underutilized host.
        end if
    end for
end while
```

Let N and M represent the number of hosts and the number of VMs of the data center, respectively. n is the number of considered training data to predict the future utilization. The complexity analysis of TLLRBA and its basic operations are as follows:

- Predicting the future utilization of all hosts and VMs based on their past data: $O(n.N + n.M)$
- Detecting overloaded hosts: $O(N)$
- Selecting migrating VMs: $O(N.M)$
- Recognizing underutilized hosts: $O(N)$
- Selecting destination hosts: $O(N.M)$

The number of training data to predict CPU utilization is a constant value. Therefore, overall complexity of our proposed algorithm, TLLRBA, is: $O(N.M)$. This complexity order is the complexity of most of the algorithms in this area [9, 13]. The complexity analysis of OLLRBA is the same, with only one difference. OLLRBA does not predict the future utilization of VMs. However, its complexity is the same as TLLRBA. Our evaluations show that TLLRBA achieves better VMs placements in terms of consolidation and energy savings, compared to OLLRBA (see Sect. 5).

5 Experimental setup and results

This section describes experimental setup used for the simulations of the paper and then compares the proposed and compared approaches in terms of energy consumption, average SLA violation and computation time. For a fair comparison, all the considered approaches are applied at the same 5-minute time interval, and Algorithm 1 has been used as the main model of the algorithms. The difference between proposed and compared approaches is in underutilized and destination host selection methods. Indeed, compared algorithms use different methods instead of Algorithm 2 and Algorithm 3 in their whole process. The name of the first and second compared algorithms indicates the used methods for underutilized and destination host selection. Compared approaches are as follows:

– MinUtil-MinPower [9]: This method is the default method implemented in CloudSim. This method selects the underutilized hosts based on their current CPU utilization starting from the host with minimum CPU utilization. Also, for all candidate destination hosts, it calculates additional power imposed by hosting the new VM and selects the host with minimum additional power consumption.
– MinUtil-MaxUtil: This method selects the underutilized host with the same method of MinUtil-MinPower algorithm. It also selects a host with maximum CPU utilization among all candidate destination hosts.
– MPABFD [14]: MPABFD is a dynamic consolidation algorithm presented in [14]. The destination host selection method in this algorithm is different from OLLRBA. Each host in [14] can be in normal, underutilized or overloaded state. MPABFD [14] selects the destination host based on two conditions: (1) the host's utilization is less than a defined upper threshold and (2) the predicted future state using recent historical data is normal.
– LIRCUP [13]: This algorithm first predicts the CPU utilization of hosts by linear regression and then determines the overutilized and underutilized hosts consequently. This algorithm also uses the same method as MinUtil-MinPower to select destination hosts of migrating VMs.

5.1 Experimental setup

To evaluate the effectiveness of our solution, we use the known data center simulator, CloudSim [26]. Two types of servers have been simulated in CloudSim: (1) Type 1: HP ProLiant ML110 G4, and (2) Type 2: HP ProLiant ML110 G5. We consider a data center with 800 hosts and a shared storage infrastructure. The half of the data center's hosts are Type 1 and the other half are Type 2. Also, four types of VMs are simulated whose characteristics are shown in Table 1. Algorithms are evaluated using different real workload traces provided as a part of the CoMon project [27]. In this project, CPU utilization has been obtained every 5 minutes from more than a thousand PlanetLab VMs hosted on more than 500 hosts. Five-minute time interval is selected to apply dynamic consolidation in our simulation. Indeed, the time interval used to apply dynamic consolidation depends on the nature of the applications running on the data center. We select six days of traces collected during March and April 2011 of the CoMon project to cover different number of VMs and resource utilization patterns (which leads to different CPU utilization means and standard deviations). The

Table 1 Characteristics of defined VMs

Characteristics	Type 1	Type 2	Type 3	Type 4
Number of cores	1	1	1	1
MIPS	2500	2000	1000	500

Table 2 Characteristics of workload traces

Characteristics	03/06	04/12	04/11	04/09	04/03	03/22
Number of VMs	898	1054	1233	1358	1463	1516
CPU Util. Mean (%)	11.44	11.54	11.56	11.12	12.39	9.26
CPU Util. St.dev (%)	16.83	15.15	15.07	15.09	16.55	12.78

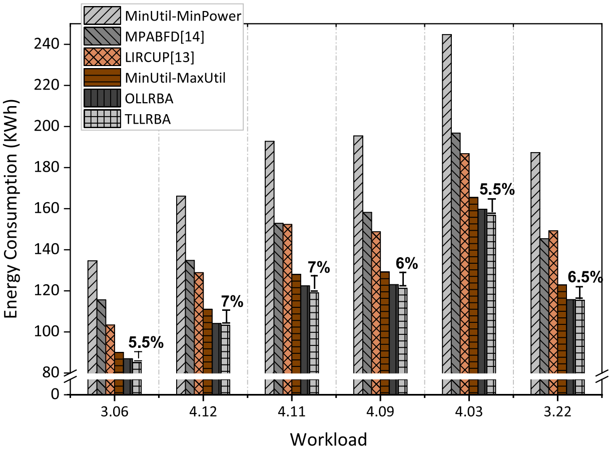

Fig. 1 Energy consumption for 23-h operation of the modeled data center

characteristics of the workload traces are shown in Table 2. The name of each workload trace represents the day and month of collecting data.

As mentioned before, consolidation algorithm is applied every 5 minutes, in our simulations. CloudSim simulator saves four utilization values during this 5-minute time interval, for each host. Therefore, to use the historical data of one last hour as the training data, *n* has been set to 84. To evaluate the accuracy of predicting host and VM utilization by linear regression, we have calculated R-squared for the last prediction of one host and one VM. R-squared for the considered host and VM is 0.68 and 0.76, respectively. Moreover, the overfitting test has been done using cross-validation, for one prediction of one host and one VM. The test exhibited that the learned models fit the test data as well as the training data.

5.2 Energy consumption and SLA violation

We begin the evaluation by calculating the energy consumption of the data center for the proposed and compared algorithms. To avoid the effect of the initial VM allocation on the results, we compute the energy for 23 hours of the data center from the second hour up to the end of the day. Figure 1 represents the energy consumption of the data center running different benchmarks for the proposed and compared algorithms. Figure 1 shows that OLLRBA and TLLRBA approaches achieve minimum energy consumption for all traces. Furthermore, the proposed approaches keep their efficiency even for large workloads. The proposed algorithm TLLRBA improves the energy consumption 6–37% for 03/06 benchmark and 6–38% for 03/22 benchmark compared to other approaches.

As mentioned before, the first objective of consolidation is increasing energy savings, which leads to packing more VMs on one host. As this may affect QoS, SLA violation is considered as the second parameter to be investigated. Figure 2 represents the average SLA violation of 23-h operation of the data center. As OLLRBA and TLLRBA approaches migrate VMs to the hosts with maximum predicted CPU utilizations, they increase SLA violation compared to MinUtil-MinPower and MPBABFD [14] approaches, although this increase is less than 1% for all benchmarks. As we expected, MinUtil-MaxUtil imposes maximum SLA violation. This is because MinUtil-MaxUtil migrates VMs to the hosts with maximum current CPU utilization.

5.3 Scalability evaluation

The scalability of our proposed approaches compared to other introduced approaches is investigated in this section. The largest real traces in terms of the number of VMs in PlanetLab consist of 1516 VMs. Therefore, we used the same strategy of [12] and combined the available traces to produce larger ones, obtaining synthetic traces shown in Table 3.

Figure 3 represents the energy consumption of the data center running synthetic benchmarks for the proposed and compared algorithms. OLLRBA and TLLRBA achieve minimum energy consumption for all synthetic traces showing the ability of our proposed approaches in keeping their efficiency even for large workloads. The proposed algorithm TLLRBA decreases energy consumption by 6–38%, 6–36% and 5.5–36% compared to other approaches, for synthetic trace 1, synthetic trace 2 and synthetic trace 3, respectively. Also, the difference between

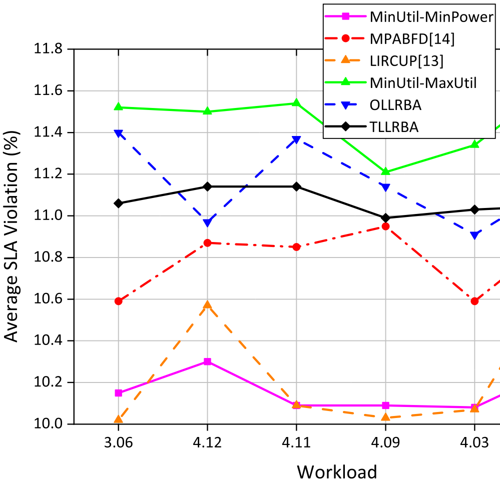

Fig. 2 Average SLA violation of the modeled data center for 23-h operation

Table 3 Synthetic workload traces

	Number of VMs	Number of hosts
Synthetic trace 1	3877	4000
Synthetic trace 2	5108	5200
Synthetic trace 3	7522	7600

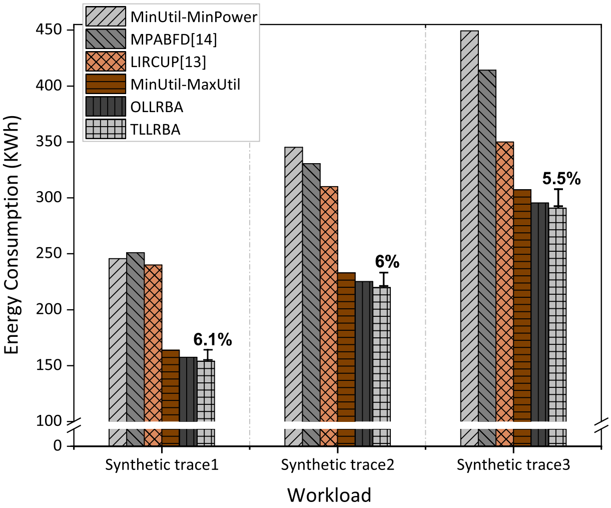

Fig. 3 Energy consumption for 23-h operation of the modeled data center

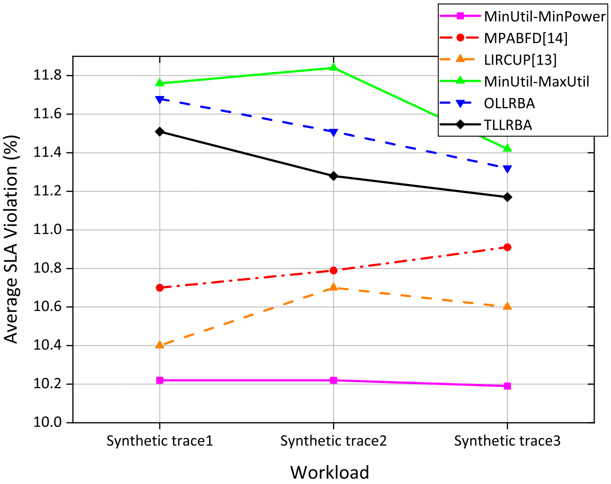

Fig. 4 Average SLA violation of the modeled data center for 23-h operation

the average SLA violation of MinUtil-MinPower (with minimum SLA violation) and our proposed approaches remains less than 1% for all three synthetic traces (see Fig. 4).

The computation time of the dynamic consolidation algorithms is another important parameter, which increases with the size of the data center and its workload. Noting that the larger traces have been simulated in this section, we investigate the runtime of the proposed and compared algorithms in this section.

Figure 5 represents the runtime of a single run of the introduced algorithms. The runtime of MinUtil-MinPower and MinUtil-MaxUtil algorithms is less than MPABFD [14], LIRCUP [13], OLLRBA, and TLLRBA algorithms. This is because a single parameter (current CPU utilization) is used in different parts of MinUtil-MinPower and MinUtil-MaxUtil algorithms, while the MPABFD [14], LIRCUP [13], OLLRBA, and TLLRBA algorithms review the hosts' historical data. However, the computational complexity analysis at the end of Sect. 4 shows the complexity of our proposed algorithms is the same as the basic algorithms, i.e., MinUtil-MinPower and MinUtil-MaxUtil.

Among MPABFD [14], LIRCUP [13], OLLRBA, and TLLRBA algorithms, MPABFD [14] imposes much higher computation time (58–71% higher than TLLRBA) because it reviews the history of all candidate hosts to calculate the probability of moving to all possible host states. Finally, to gain more insight into the time overhead of introduced algorithms, MinUtil-MinPower and MinUtil-MaxUtil impose 0.4–0.7%, OLLRBA imposes 2–4%, TLLRBA and LIRCUP [13] impose 2.7–4.6% and MPABFD [14] imposes 4.5–10% overhead in the considered 5-minute time interval to run synthetic benchmarks.

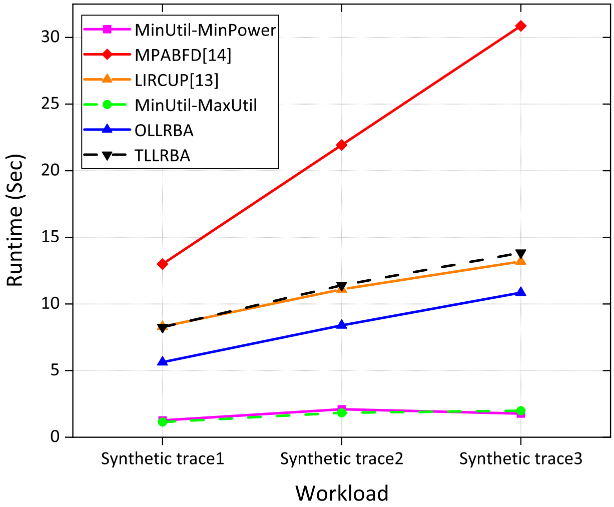

Fig. 5 Average runtime of each approach for 23-h operation of the modeled data center

6 Conclusion

In this paper, we proposed an approach to improve energy efficiency of the data center's resources. To achieve this objective, first, a linear regression-based algorithm has been designed to select the appropriate destination hosts for migrating VMs (during the dynamic consolidation procedure). Then, the proposed approach has been improved to also predict the hosts utilization and select the underutilized hosts accordingly. In the evaluation section, we have considered several previously presented approaches to compare with our proposed approach. The proposed and compared schemes have been evaluated using CloudSim simulator and PlanetLab workload traces in terms of energy consumption, SLA violation and computation time. The results show that the proposed approach decreases energy consumption when compared to approaches in the current state of the art. This improvement comes with some SLA violation. Our proposed approaches cause about 1% higher average SLA violation than the basic consolidation schemes; however, their imposed SLA violation is less than the violation imposed by SOA approach MPABFD [14]. Moreover, our proposed approach keeps its efficiency for higher scale benchmarks and data centers, which proves its scalability. As future work, we envision to consider memory utilization besides the CPU utilization, in designing consolidation algorithm.

References

1. Koomey JG (2007) Estimating total power consumption by servers in the U.S. and the world. Lawrence Berkeley National Laboratory, Stanford University
2. Shehabi A, Smith S, Sartor D, Brown R, Herrlin M, Koomey J, Masanet E, Horner N, Azevedo I, Lintner W (2016) United states data center energy usage report. Lawrence Berkeley National Laboratory, Berkeley
3. Barham P, Dragovic B, Fraser K, Hand S, Harris T, Ho A, Neugebauer R, Pratt I, Warfield A (2003) Xen and the art of virtualization. ACM SIGOPS Oper Syst Rev 37:164–177 ACM
4. Leelipushpam PGJ, Sharmila J (2013) Live vm migration techniques in cloud environment–a survey. In: 2013 IEEE Conference on Information & Communication Technologies. IEEE, pp 408–413
5. Sobel W, Subramanyam S, Sucharitakul A, Nguyen J, Wong H, Klepchukov A, Patil S, Fox A, Patterson D (2008) Cloudstone: multi-platform, multi-language benchmark and measurement tools for web 2.0. Proc CA 8:228
6. Pahlevan A, Qu X, Zapater M, Atienza D (2017) Integrating heuristic and machine-learning methods for efficient virtual machine allocation in data centers. IEEE Trans Comput Aided Des Integr Circuits Syst 37(8):1667–1680
7. Monil MAH, Rahman RM (2015) Implementation of modified overload detection technique with vm selection strategies based on heuristics and migration control. In: 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS), IEEE, pp 223–227
8. Cao Z, Dong S (2014) An energy-aware heuristic framework for virtual machine consolidation in cloud computing. J Supercomput 69(1):429–451
9. Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr Comput: Pract Exp 24(13):1397–1420
10. Murtazaev A, Oh S (2011) Sercon: server consolidation algorithm using live migration of virtual machines for green computing. IETE Tech Rev 28(3):212–231

11. Wu Q, Ishikawa F, Zhu Q, Xia Y (2016) Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters. IEEE Trans Serv Comput 12(4):550–563
12. Haghshenas K, Pahlevan A, Zapater M, Mohammadi S, Atienza D (2019) Magnetic: Multi-agent machine learning-based approach for energy efficient dynamic consolidation in data centers. IEEE Trans Serv Comput. https://doi.org/10.1109/TSC.2019.2919555
13. Farahnakian F, Liljeberg P, Plosila J (2013) Lircup: linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers. In: 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, IEEE, pp 357–364
14. Melhem SB, Agarwal A, Goel N, Zaman M (2017) Markov prediction model for host load detection and vm placement in live migration. IEEE Access 6:7190–7205
15. Masoumzadeh SS, Hlavacs H (2013) An intelligent and adaptive threshold-based schema for energy and performance efficient dynamic vm consolidation. In: European Conference On Energy Efficiency in Large Scale Distributed Systems, Springer, pp 85–97
16. Horri A, Mozafari MS, Dastghaibyfard G (2014) Novel resource allocation algorithms to performance and energy efficiency in cloud computing. J Supercomput 69(3):1445–1461
17. Nguyen TH, Di Francesco M, Yla-Jaaski A (2017) Virtual machine consolidation with multiple usage prediction for energy-efficient cloud data centers. IEEE Trans Serv Comput 13(1):186–199
18. Khoshkholghi MA, Derahman MN, Abdullah A, Subramaniam S, Othman M (2017) Energy-efficient algorithms for dynamic virtual machine consolidation in cloud data centers. IEEE Access 5:10709–10722
19. Khan MA, Paplinski AP, Khan AM, Murshed M, Buyya R (2018) Exploiting user provided information in dynamic consolidation of virtual machines to minimize energy consumption of cloud data centers. In: 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), IEEE, pp 105–114
20. Li Z, Yu X, Yu L, Guo S, Chang V (2020) Energy-efficient and quality-aware vm consolidation method. Future Gener Comput Syst 102:789–809
21. Ashraf A, Porres I (2018) Multi-objective dynamic virtual machine consolidation in the cloud using ant colony system. Int J Parallel Emerge Distrib Syst 33(1):103–120
22. Wang H, Tianfield H (2018) Energy-aware dynamic virtual machine consolidation for cloud datacenters. IEEE Access 6:15259–15273
23. Alicherry M, Lakshman T (2013) Optimizing data access latencies in cloud systems by intelligent virtual machine placement. In: 2013 Proceedings IEEE INFOCOM, IEEE, pp 647–655
24. Sousa S, Martins F, Alvim-Ferraz M, Pereira MC (2007) Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations. Environ Model Softw 22(1):97–103
25. Seber GA, Lee AJ (2012) Linear regression analysis, vol 329. Wiley, Hoboken
26. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R (2011) Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw: Pract Exp 41(1):23–50
27. Park K, Pai VS (2006) Comon: a mostly-scalable monitoring system for planetlab. ACM SIGOPS Oper Syst Rev 40(1):65–74