Energy Efficient Resource Selection and Allocation Strategy for Virtual Machine Consolidation in Cloud Datacenters

Yaohui CHANG†, ‡, Student Member, Chunhua GU†(b), Fei LUO†, Guisheng FAN†, Nonmembers, and Wenhao FU†, Student Member

SUMMARY Virtual Machine Placement (VMP) plays an important role in ensuring efficient resource provisioning of physical machines (PMs) and energy efficiency in Infrastructure as a Service (IaaS) data centers. Efficient server consolidation assisted by virtual machine (VM) migration can promote the utilization level of the servers and switch the idle PMs to sleep mode to save energy. The trade-off between energy and performance is difficult, because consolidation may cause performance degradation, even service level agreement (SLA) violations. A novel residual available capacity (RAC) resource model is proposed to resolve the VM selection and allocation problem from the cloud service provider (CSP) perspective. Furthermore, a novel heuristic VM selection policy for server consolidation, named Minimized Square Root available Resource (MISR) is proposed. Meanwhile, an efficient VM allocation policy, named Balanced Selection (BS) based on RAC is proposed. The effectiveness validation of the BS-MISR combination is conducted on CloudSim with real workloads from the CoMon project. Evaluation results of experiments show that the proposed combination BS-MISR can significantly reduce the energy consumption, with an average of 36.35% compared to the Local Regression and Minimum Migration Time (LR-MMT) combination policy. Moreover, the BS-MISR ensures a reasonable level of SLAs compared to the benchmarks.

key words: residual available capacity model, server consolidation, virtual machine migration, energy consumption, cloud computing

1. Introduction

Cloud computing leverages utility computing, grid computing, and distributed computing to provide services of infrastructure, platform and software for users [1] and supplies the on-demand services via the network. Due to the ever-increasing cloud infrastructure demand, the sharp increase of data center (DC) in size and numbers has a significant increase in power consumption. The energy consumption of datacenters increased by 56% worldwide from 2005–2010, which accounts for 1.3% of total electricity use [2]. And a global annual datacenter construction size will be $78 billion by 2020 [3]. The energy consumption cost of datacenter accounts for 45% of the total operating cost [4], which was the largest part of all. The high quality of service (QoS) requires CSPs to make a trade-off between the energy and performance, as aggressive energy saving may lead to performance degradation. High energy consumption not only brings high operating cost but also results in higher carbon emissions. Therefore, it is the major concern to design energy efficient resource management strategies [5].

Low resource utilization [6] is a major factor in the high power consumption of data centers. It reported that physical servers’ average CPU utilization is only 10% to 50% at most of the time in the datacenters [4]. Taking Google as an example, the utilization of servers of Google’s clusters is less than 50% on average [7]. Hence, it is an urgent challenge to design the efficient resources allocation schemes, which will not only reduce the energy consumption but also improve the resources utilization level under the SLA and QoS constraints.

Dynamic Voltage and Frequency Scaling (DVFS) and Server Consolidation (SC) are the two effective energy saving techniques widely adopted in virtualized cloud data centers. DVFS technique by adjusting the frequency and voltage of CPU to save energy, which may result in performance degradation and prolongs the runtime of the tasks. Virtualization technique (VT) promotes the utilization level of the resource by sharing a physical server with several VM instances. With VT, SC aggregates VMs into fewer servers through migration to reduce the number of active hosts to save energy.

One major drawback of the current server consolidation approaches is that proposed solutions only concentrate on the CPU dimension and ignore other dimensional resources such as RAM and bandwidth. VM migration is expensive, as it not only expands the network bandwidth overhead but also causes data centers’ network congestion. Furthermore, it can lead to performance degradation and SLA violations. Hence, it is necessary to reduce the number of VM migrations. Therefore, efficient VMP schemes are needed to achieve efficient resource utilization and energy conservation.

The main contributions of this paper are: Firstly, residual available capacity (RAC) model is presented to carve the availability degree of physical servers in the heterogeneous datacenters. Secondly, an energy efficient VM selection strategy named Minimized Square Root available Resource (MISR) that used for selecting the proper VM to migrate is proposed. Moreover, an efficient VM allocation policy based on RAC model named Balanced Selection (BS) that used for finding a new placement for the VM.
to migrate is presented. Thirdly, the proposed combined optimization policy BS-MISR was evaluated on CloudSim with real workload traces from the PlanetLab. Experimental results show that the BS-MISR significantly reduces the energy consumption while providing a reasonable level of SLAs.

2. Related Work

Many studies both in industry and academia have focused on the energy efficient research of cloud data centers. Seen from the CSP perspective, a key requirement is to ensure efficient resource utilization and energy efficient resource provisioning [9]. Generally, the VM migration [10] consists of four divisions: overloaded hosts (hotspot PMs) detection, under-loaded hosts (cold-spot PMs) detection, choose proper VMs to migrate (VMs selection) and allocate the VMs to under-loaded hosts (VMs placement). The efficient VMP is conducive to efficient resource provisioning and power saving.

Known as a VM assignment problem, VMP is critical to the efficient resource provisioning. Resource allocation in cloud computing is demonstrated in Fig. 1. The VMP accomplishes the map function between the VMs and PMs with two steps. At the first step, all cloud tenants’ requests are encapsulated into different VMs. The next step, several models and policies which based on different optimization objectives are utilized to assign the specific VM to the selected PM to accomplish the placement. VMP can be divided into two categories: online provisioning and batch provisioning [11]. The former receives the requests and places them immediately. The latter collects requests to form a group and places them under several constraints.

Several works formulate VMP as a variable size bin packing problem [12], [29] where PMs are conceived as bins and VMs as items. Therefore, the classical bin-packing algorithms should be modified to apply in the VM consolidation problem for three main reasons [13]: (a) the multi-dimensional resources (e.g. CPU, Memory etc.); (b) the different bin sizes (e.g. the heterogeneous servers); (c) multi-objective optimization functions (e.g. energy, load balance etc.) with SLA constraints and QoS requirements. In fact, the multi-dimensional bin packing has great difference with multi-capacity bin packing, which can be seen from the illustration in Fig. 2. The details are reported in Sect. 3.1.

Many of heuristic approaches are proposed to solve the VMP such as various greedy algorithms: First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD), Best Fit Decreasing (BFD), where they do not provide the global optimum solutions. The Ref. [10] proposed several metrics to rank servers by considering an adaptive upper bound based on a statistical analysis of historical CPU data. The Median Absolute Deviation method (MAD), Interquartile Range method (IQR), Local Regression method (LR) and Robust Local Regression (LRR) have been proposed to estimate the overload thresholds of CPU utilization. And they also proposed three different virtual machine selection policies: Minimum Migration Time policy (MMT), Random Selection policy (RS) and the Maximum Correlation policy (MC). The proposed metrics only use the current CPU utilization as the main criterion to decide VMs’ migration destinations. Reference [14] proposed a Modified Best Fit Decreasing (MBFD) algorithm by sorting the VMs in the decreasing order and PMs in the increasing order of their capacity. The limitation of MBFD is only single objective considered and could not accommodate the scalable situations of data centers.

The Ref. [23] concentrated on the predictive value based on the local regression of historical data, and it divided the status of the hosts into three categories: under-utilized, properly-utilized and over-utilized. Based on the LR-MMT introduced in [10], they combined with the requested MIPS of the VMs, and (or) the number of VMs when SLA violation last occurred, the SLA violations were greatly reduced and achieved better energy conservation. The reduction of the energy consumption is only with a minimum of 0.15%, and a maximum of 14.12% compared to the LR-MMT. But the decision method of thresholds is not given, which will affect the energy consumption as well as the SLA metrics.

The Ref. [24] improved the framework proposed in [10], and refined the criteria whether the host is over-loaded. Partly similar to [23], it divided the under-loaded hosts into more fine-grained states: UH, UM and UL. The new framework reduces the number of VM migrations by a quarter, and the energy consumption reduction is 1.15% on average compared to that in [10].

Some other works used bio-inspired and nature-inspired algorithms, such as PSO [12], [42], GA [16], ACO [5], [17], BBO [18], Firefly [19] etc. Sharma [12] focused on the key goals of multi-objective VM allocation based on Particle Swarm Optimization (PSO) and VM migration to reduce the energy consumption, resource
wastage and SLA violations. Gao et al. [17] proposed a multi-objective ant colony based VMs allocation at the homogeneous data center which is not realistic in fact. However, the intelligent evolutionary algorithms are sensitive to the parameters, and need artificial adjustment according to the system status. Therefore, the cost of parameter optimization is high, and it’s a long time to achieve the better trade-off between multiple objectives.

Several works paid attention to the predictive frameworks, and variety of host overloading methods [20]–[22], [25] have been proposed. Z. Xiao et al. [21] proposed the dynamic resources allocation using VMs in cloud data center. The limitation of the Xiao’s work is that if the load is not predicted appropriately, SLA violation is suffered. Li et al. [22] developed a Bayesian network-based estimation model for live VM migration. The Ref. [25] proposed an adaptive fuzzy threshold based manner to detect the over-loaded hosts and under-loaded hosts, which assisted to achieve energy and performance tradeoff.

In this paper, we consider not only the heterogeneity of physical machines, but also the heterogeneity of virtual machine types. From the perspective of remaining available resources, we design efficient VM selection strategy and placement policy based on the RAC model to save energy.

3. Problem Formulation

In this paper, PMs, physical servers and hosts are alternatively used with the same meaning in different situations. To facilitate expression, some assumptions have been made and shown below:

a) The IaaS Cloud environment is assumed and the tenants lease slices of the hardware of the datacenter provided by the cloud service provider. Instances of tenants are isolated each other under the assistance of virtualization.

b) The service requests are received from the cloud tenants are encapsulated into the service instances, where they perform in the forms of VMs. All VMs need a fixed amount of cloud resource (i.e. CPU and memory) for a specified amount of time.

c) The cloud consists of heterogeneous servers, which conforms to the reality of data centers. A shared cloud storage resources system is employed in the cloud which facilitates data sharing and transport between PMs. So network bandwidth and storage requirements of VMs are out of this paper’s scope.

3.1 VM Placement and Multi-Capacity Bin Packing

Server consolidation in cloud data center is usually treated as a variable bin packing problem, in which PMs are considered as bins and VMs as items. As the bin packing problem is NP-hard, so it can be solved with heuristic methods such as Best Fit Decreasing (BFD) algorithm. The solution of the algorithm uses no more than $11/9 \cdot OPT + 1$ bins, where $OPT$ is the number of bins provided by the optimal solution [27]. The multi-capacity bin packing (MCBP) problem [28] is very similar to the classical multi-dimensional bin packing (MDBP) problem, with the exception that bins are non-homogeneous. The difference is explicated in Fig. 2.

In MDBP, the item can be put into a bin only when geometric space is enough whatever how many swaps happened between horizontal and vertical dimensions (see Fig. 2 (a)). But any portion of the horizontal or vertical capacity can be used by only one item in MCBP (see Fig. 2 (b)). In MCBP, once the resource is utilized or occupied by one VM, the resource space cannot be reused by any other VMs at the same time [29], only if the instance is terminated or migrated out. VMs are dynamically time-varied objects, tightly packing them with traditional bin packing heuristics may lead to the “stability of placement” [30] concerns. In this research, virtual machine placement problem will be formulated as multi-capacity vector bin packing (MCBP).

3.2 Formulation of the MCBP for VMP

Supposing there are $n$ items to be packed into at most $m$ bins, and requiring to use as few bins as possible with each item’s dimensional request is not exceeded the corresponding dimension capacity of that bin. Some related definitions are given below:

Definition 1 (Bins): Let $B = \{B_1, B_2, \ldots, B_m\}$ be a set of $m$ heterogeneous bins ($|B| = m$) where the sizes of bins are identical or distinct, the capacity of bin $B_i$ is defined as a d-dimensional vector $C_i = \{C_{i,1}, C_{i,2}, \ldots, C_{i,d}\}$, where $C_{i,k}$ is the $k$-th dimension resource capacity and $C_{i,k} > 0$ for all the $k = \{1, 2, \ldots, d\}$.

Definition 2 (Items): Let $X = \{X_1, X_2, \ldots, X_n\}$ be a set of $n$ items, where the items are required to be packed into as few bins as possible without exceeding the bin’s capacity. An item $X_j$ in $X$ can be represented as a d-dimensional vector, $X_j = (r_{j,1}, r_{j,2}, \ldots, r_{j,k}, \ldots, r_{j,d})$ where $r_{j,k}$ is the $k$-th dimension requirement of the $j$-th items and for $\forall k \in \{1, 2, \ldots, d\}$.

Definition 3 (Mapping Function): We define a mapping function $f$: $\{1, \ldots, n\} \rightarrow \{1, \ldots, m\}$, such that $f(j) = i$, $\forall j \in \{1, 2, \ldots, n\}$, $i \in \{1, 2, \ldots, m\}$. Formally, a mapping is feasible if all items have been successfully mapped to bins: $\forall j$, $f(j) \neq \phi$, and the combined demand of resources are within the bin capacities. For $\forall k \in \{1, 2, \ldots, d\}$, $\sum_{j=1}^{n} r_{j,k} x_{j,i} \leq C_{i,k}$ where $x_{j,i}$ stands for the item $X_j$ whether mapped to $B_i$, if yes, $x_{j,i} = 1$; otherwise 0.

Definition 4 (Solution): A solution of the multi-capacity bin packing problem is a feasible mapping that can be represented as $S = \{S_1, S_2, \ldots, S_t\}$, for $\forall t \in \{1, 2, \ldots, m\}$, and $i \in \{1, 2, \ldots, m\}$ where $S_t$ can be represented $S_t = \{p_{1,t}, p_{2,t}, \ldots, p_{q,t}\}$ where $p_{q,t}$ represents that the item $q$ can be packed into $B_t$. The packing decision is subject to some constraints as following shows:

$$\sum_{j=1}^{n} r_{j,k} x_{j,i} < C_{i,k} \quad (1)$$


\[ \sum_{i=1}^{m} x_{ji} = 1 \quad (2) \]

\[ x_{ji} = \{0, 1\}, \forall j \in \{1, 2, \ldots, n\}, \forall i \in \{1, 2, \ldots, m\} \quad (3) \]

The first constraint means the items requirement in each dimension should not exceed the corresponding capacity of the bin, while the second and third constraints ensure that each item must be accommodated in a single bin.

4. Energy-Aware Resource Selection and Allocation

The system architecture is presented in Fig. 3. The data center consists of many server clusters. Moreover, the number of physical servers in any cluster is bounded because the cluster has limited hardware resources, peak power in terms of maximum power supply that it can consume, and the peak network bandwidth that it can use.

We assume a data center consists of several small computing clusters, each of which consists of two or more PMs, and manages the VMs located on it with its own service manager (we named it as local cluster manager). This scenario is analogous to the cluster schedulers in Google’s. Similarly to OpenStack, each cloud’s cluster has a cloud orchestrator. Furthermore, a virtualization hypervisor is employed, where they work together to create VM instances with different specifications, allocate them to tenants and enable instances available for tenants with QoS requirements. Finally, a shared cloud storage system is employed to save data [31], which can facilitate data sharing between all PMs and easily provide users with a shared cloud storage resource and enable live migration of VMs rapidly.

4.1 Power Model

With the virtualization technologies, IaaS cloud providers provide a resource selection interface based on abstract computational units (e.g. EC2 compute unit). Having considered the estimated peak usage of their workloads, cloud tenants usually rent computational units more than what they really need, which resulted in cloud providers having to deal with massive hardware deployments.

The main energy consumption in data centers comes from the computing nodes [32], which are determined by hardware efficiency. Energy consumption of computing node mainly comes from the components such as CPU, Memory, storage systems and enabled network interface cards. Most studies [33]–[35] have shown that the power consumption by servers can be accurately described by a linear relationship between the power consumption and CPU utilization, even when DVFS is applied. In general, given a CPU utilization \( u \in [0, 1] \), the power consumed by the server can be denoted as:

\[ P(u) = P_s + P_d \times u \quad (4) \]

Where \( u \) is the percentage of CPU utilization, and \( P_s \) refers to static power consumption which is independent of workload. As \( P_s \) reflects the idle server’s high energy consumption, it becomes the main motivation for efficient server provisioning [36]. \( P_d \) refers to the dynamic power consumption that mainly depends on a specific usage scenario, clock rates, I/O activity, short-circuiting current and switched capacitance [37]. Let utilization \( u(t) \) be the function of time \( t \), and the Eq. (4) can be denoted as:

\[ P(u(t)) = P_s + P_d \times u(t) \quad (5) \]

**Definition 5.** For each \( \text{vm}_j \in \text{PM}_i \) we define the CPU utilization of \( \text{PM}_i \) as the ratio of the CPU resources allocated to the VMs to the total CPU capacity during the time slot \( t \) period:

\[ u_i(t) = \sum_{\text{vm}_j \in \text{PM}_i} \frac{\text{vm}_j^{cpu}(t)}{\text{Res}_j^{cpu}} \quad (6) \]

Where \( \text{vm}_j^{cpu}(t) \) stands for the CPU utilization of \( \text{vm}_j \) at \( t \) time slot; \( \text{Res}_j^{cpu} \) is the CPU capacity of \( \text{PM}_i \); and \( J \) is the number of VMs running on \( \text{PM}_i \), respectively. The total energy \( E_i \) consumed by \( \text{PM}_i \) at time period \( t \) can be defined as:

\[ E_i = \int P(u_i(t))dt \quad (7) \]

Given a Cluster with \( N \) PMs, the total energy consumed \( E \) can be expressed as formula (8) as shown below:

\[ E = \sum_{i=1}^{N} \int P_s + P_d \times \sum_{\text{vm}_j \in \text{PM}_i} \frac{\text{vm}_j^{cpu}(t)}{\text{Res}_j^{cpu}} dt \quad (8) \]

4.2 Residual Available Capacity Model

Power consumption in heterogeneous systems depends greatly on the type of processors used in the server farm. Since CPU is the largest energy consumer [35]–[37] of the server, migrating VMs from one PM to another has a positive impact on reducing the energy consumption, by influencing the CPU load of the server. Power efficiency reflects on how much useful work produced by the server for a
given power consumption [38]. The higher CPU utilization of PM, the better power efficiency. The utility function describes the satisfaction for a certain service obtained by the tenant [46], combined with the criteria of the utility function [47], we use the exponential function to carve the PM’s energy efficiency.

\[
U(x) = \begin{cases} 
0, & 0 < x \leq \theta \\
2^{-\alpha - \theta} - 1, & x > \theta 
\end{cases} \quad (9)
\]

Where \( \alpha \) and \( \theta \) affect the sharpness and offset of the function, respectively. Let \( x \) be the normalized value which belongs to \([0, 1]\), and we can use the utility function to measure the power efficiency level (e.g. \( \alpha = 5, \theta = 0 \)). If \( x \) be the utilization of resource (e.g. CPU utilization), the larger PM’s CPU utilization is, the better PM’s power efficiency gets.

Due to the complex and dynamic workload of the cloud system, different available resources have varied influences on the physical servers. In [39], the remaining resource of physical host \( L_i \) is defined to represent the performance power of physical host \( L_i \):

\[
L_i = aL_{cpu}^i + bL_{mem}^i 
\]

\[
a + b = 1 \quad (10)
\]

Where \( L_i \) represents the remaining computing power of the physical host \( i; L_{cpu}, L_{mem} \) are the remaining CPU resource and the remaining memory resource, respectively. \( a, b \) are the CPU weight value of \( L_i \), and the memory weight value of \( L_i \), respectively.

It’s obvious that physical servers are treated as multi-dimensional resource units in the IaaS cloud scenario. Different dimension resource has different importance in different scenes. For example, the application requests may be classified as computing-intensive, data-intensive and I/O-intensive. Therefore, it is necessary to take all the dimensional resources into account. We extend the Eq. (10), and use the residual available capacity (RAC) to measure the physical servers’ load ability when the utility of servers is identical. RAC value reflects the utility level of resource utilization of the physical server \( pm_i \). The definition of RAC is defined in Eqs. (12)–(13) as follow:

\[
RAC(pm_i) = \sum_{d=1}^{D} \lambda_d rac^d_{pm_i} \quad (12)
\]

\[
\sum_{d=1}^{D} \lambda_d = 1, \quad 0 \leq \lambda_d \leq 1 \quad (13)
\]

Where \( rac^d_{pm_i} \) represents the residual available capacity of \( pm_i \) on dimension \( d \) that can be utilized and allocated to the VMs, \( \lambda_d \) is the dimensional weight of \( pm_i \) on dimension \( d \), and \( d \in \{1, 2, \ldots, D\} \). Parameter values are obtained by BP Neural Network (BPNN) with system history data [39].

The general BPNN consists of three layers: the input, hidden and output layers. Particularly, BPNN has one or more hidden layer, thus allowing the networks to model complex functions [40]. Let the number of nodes in the input, hidden and output layers be \( m, s \) and \( n \), respectively. Let \( X \in R^{n \times m} \) be the system history data, where \( n \) is the number of weights to be learned. And \( x_p \in R^m \) is an \( m \)-dimensional input vector, which consists of system workload level, system performance parameters and system physical resource utilization in \( D \) dimensions. The output vector of the hidden layer \( h_p \), and the output vector of the output layer \( y_p \) are [41]:

\[
h_p = f(W_1 \times x_p + \theta_h), \quad y_p = f(W_2 \times h_p + \theta_o)
\]

Where \( f(\cdot) \) is the activation function, \( W_1 \in R^{s \times m} \) is the weight matrix between the input and hidden layers, and \( \theta_h \in R^s \) is the threshold vector; \( W_2 \in R^{n \times s} \) is the weight matrix between the hidden and the output layers, and \( \theta_o \in R^n \) is the threshold vector.

4.3 Proposed Algorithms

Server consolidation has several steps (see Sect. 2), we mainly concentrated on two aspects: (a) Select the most suitable VM from the over-loaded hosts to migrate, and we named this policy as Minimized Square Root VM Selection (MISR). (b) Based on the RAC model, find new placement from the under-utilized hosts for the VM to migrate, we named it Balanced Selection (BS) policy.

The flowchart of server consolidation is shown in Fig. 4. First of all, the threshold based heuristic algorithm is used to select the overloaded hosts. Secondly, the novel policy named MISR for selecting the VMs from over-utilized hosts is applied. Thirdly, we propose a novel policy for optimizing the VMs allocation based on resource-aware capacity utility model to form the migration map such as <vm_id, pm_id> pairs. Finally, VMs allocation module is requested to complete the server consolidation process.

4.3.1 Host Overloading Detection and Underutilized Host Selection

In the virtualized data centers, different types of applications share the physical resources. On the consideration of comparing with other adaptive dynamic thresholds, the static utilization threshold to detect the over-utilized hosts is utilized. If the utilization rate of PM exceeds a predefined threshold (e.g. 80%), the server is considered to be overloaded.

If the host utilization is lower than the threshold, the host is regarded as under-loaded. And it may be chosen as
the destination PM for the VMs to migrate. We try to migrate the VMs from current host to another one that remains under-loaded after placement. Once all the VMs are migrated out from the current host, it is switched off. If not, the host is kept active. The process is iteratively repeated for all active under-loaded hosts.

### 4.3.2 Minimized Square Root VM Selection Policy

In this section, we propose a novel policy for selecting the VM from over-utilized hosts. We choose the VM which has the minimal impact on load to migrate out, where the impact is defined by the RAC model. The proposed VM selection strategy named Minimized Square Root (MISR) is based on the dimensional weight factors under their resource requirements constrains. Furthermore, we use MISR policy to accomplish the VM selection process, under the assumption that the network bandwidth is enough stable and has no changes.

We try to find the minimum value with different dimensional resource requirements. All the VMs resources are normalized first to make sure that all the dimensional resources are transformed to \([0, 1]\) and remove the differences in different forms of units. The equation used is given below:

\[
\bar{z}_k = \frac{z_k}{z_{k\text{max}}}
\]

Where \(z_k\) and \(z_{k\text{max}}\) refer to the normalized value and the maximum value on dimension \(k\), respectively. After that the dimensional difference will be eliminated.

For the consideration of multi-dimensional resource allocation situation, it is necessary to consider the distance differences between PMs. As we considered the resource request of the VM and the resource of the PM as multi-dimensional vectors, and need a metric to measure the distance between the resources of the PMs and the original point in multi-dimensional spaces. So we use Euclidean distance as the metric for measuring the abilities of physical servers to serve VMs, which is similar to the Ref. [42] which uses Euclidean distance to determine the ability of server’s energy efficiency.

**Definition 6.** For the physical server \(PM_i\), the \(Ed(i)\) value is defined as the Euclidean distance between \(PM_i\) and the origin point of \(N\)-dimensional space as defined below:

\[
Ed(i) = \sqrt{\sum_{k=1}^{N} (x_k - 0)^2}, \quad \forall k \in [1, \ldots, N]
\]

Where \(x_k\) is the utilization of the physical resource of the physical server \(PM_i\) on dimension \(k\). And \(k\) denotes the resource such as CPU, memory, disk and bandwidth.

Having considered the fact of that, different dimension resources may become bottleneck resource of the system under the dynamic workload, and weight method is applied to alter and affect the results. Therefore, the Eq. (15) can be rewrite as follow:

\[
Ed(i) = \sqrt{\sum_{k=1}^{N} (\omega_k x_k - 0)^2}, \quad \forall k \in [1, \ldots, N]
\]

Where \(\forall k, \omega_k \geq 0\) and \(\sum_{k=1}^{N} \omega_k = 1\).

In this paper, we concentrate on designing energy efficient VM selection policy and VM placement based. On the reality of the cloud system, the parameter weight in Eq. (16) can be given according to the experience. Without loss of generality, we treat each dimensional resource equally in this paper.

After the data is converted to the normalized data, we traverse all the VMs listed in the VM Migration List, and try to find the VM which has the minimum value of \(Ed(i)\). If not, no VM is selected to be the most suitable one. The complexity of the Algorithm 1 is \(O(n^m)\), where \(n\) is the number of PMs in the OverloadedHostList and \(m\) is the number of VMs in the MigratableVmsList that have to be allocated.

### 4.3.3 VM Allocation Policy

The RAC based VM allocation named Balanced VM new placement Selection (BS) is given in Algorithm 2. For each VM in the migration list, we first sort the vms in migrationMap list with a descending order (line 1). For each VM in the migration list, we compare the two PM’s utility function value with RAC model, and sort the PMs in a descending order based on the allocated MIPS of the PM (line 3). Then the variable of the allocated host is initialized to be NULL (line 4), for all the host involved in, it check the host whether in the excluded hosts list (line 6). Next, we check the host whether is suitable for the current VM and make sure the host will not become the over-utilized host after allocation (line 7-9). If all the constraints are satisfied, the allocated host value will become the current host after allocation (line 7-9). If all the constraints are satisfied, the allocated host value will become the current host after allocation (line 7-9). Once the loop process is finished, the current VM and the allocated host are added to the migration map if the

---

**Algorithm 1** Minimized Square Root VM Selection (MISR).

**Input:** HostList, VMList, parameter \(a\), parameter \(b\)

**Output:** MigrationVmsSet, a map set of \(<\text{vm}_id, \text{sourcehost}_id>\) pairs

1. **For each host in OverloadedHostList do**
   2. **While(True)**
      3. \(MinMetric \leftarrow \text{MaxValue}\)
      4. MaxVmMips\(\leftarrow\)findMaxmumVmMips(host)
      5. MaxVmRam\(\leftarrow\)findMaxmumVmRam(host)
      6. MigratableVmsList\(\leftarrow\)getMigratableVms(host)
      7. **For each vm in MigratableVmsList do**
         8. \(if\) vm.isInMigration \(then\) continue
         9. normalMips\(\leftarrow\)vm.getMips()/MaxVmMips
         10. normalRam\(\leftarrow\)vm.getRam()/MaxVmRam
         11. Metric \(\leftarrow\) SQRT((a*normalMips)^2+(b*normalRam)^2)
         12. If Metric < MinMetric then
            13. MinMetric \(\leftarrow\) Metric
            14. vmToMigrate \(\leftarrow\) vm
      15. **EndIf**
      16. **EndFor**
    17. host.updateVmList
    18. MigrationVmsList.add(vmToMigrate, host)
    19. **if** !isOverutilizedHost(host) **then** Break
    20. **EndIf**
    21. **EndWhile**
    22. **Return** MigrationVmsSet

\[
Ed(i) = \sqrt{\sum_{k=1}^{N} (\omega_k x_k - 0)^2}, \quad \forall k \in [1, \ldots, N]
\]
value of allocated host is altered (line 11-13). Once all the VMs in the migration set have been traversed, the process is finished and the migration map is returned (line 15).

The complexity of sort VMs in descending order is \(O(m \times \log m)\). The complexity of sort PMs in descending order is \(O(n \times \log n)\). So the complexity of the algorithm 2 is \(O(m \times \log m + n \times \log n)\), where \(n\) is the number of PMs in the HostList and \(m\) is the number of VMs in the MigrationVmsSet that have to be allocated.

### 5. Experiments and Analysis

#### 5.1 Experiment Setup

Since IaaS is the targeted system, it is natural to evaluate the proposed resource allocation algorithms on a large-scale cloud datacenter infrastructure. Due to the difficulty to conduct repeatable large-scale experiments on real infrastructure, simulations have been chosen as the realistic way to evaluate the performance of the proposed algorithms. The CloudSim[43] is a toolkit for modeling and simulating cloud computing, which provides essential classes for describing cloud computing such as computational resources, virtual machines, cloud users, and management policies. Therefore, we use CloudSim to evaluate the proposed approaches, where it can ensure the repeatability and reproducibility of experiments.

We build a data center with 800 heterogeneous PMs and 2 types of physical servers based on HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores with 1860 MHz, 4 GB) and HP ProLiant ML110 G5 (Intel Xeon 3075, 2 cores with 2660 MHz, 4 GB). In the data center, half of PMs are HP ProLiant ML110 G4 servers and the other half consists of HP ProLiant ML110 G5 servers. The energy consumption data of HP ProLiant ML110 G4 and G5 are provided by SPEC[45].

Four types of VM specifications are used: Micro (500 MIPS, 613 MB RAM), Small (1000 MIPS, 1740 MB RAM), Medium (2000 MIPS, 1740 MB RAM) and Large (2500 MIPS, 870 MB RAM), which are based on Amazon EC2 to simulate the heterogeneous requests.

#### 5.2 Workload

We conduct our experiments on real workload traces, which is publicly available workloads from the CoMon project [44], a monitoring infrastructure for PlanetLab. We have randomly chosen 10 days’ data as our experiment dataset from the workload traces collected during March and April 2011. In the dataset, each VM’s workload trace is stored in a single file while each day’s workload traces are stored in a directory, which consists of number of VMs’ workload traces. The VMs’ workload trace usage data is reported every 5 minutes from thousands of VMs which come from more than 500 places around the world. The characteristics of the VMs’ workload traces in the PlanetLab are presented in Table 1.

#### 5.3 Benchmark

To evaluate the performance of the BS-MISR, five benchmarks were utilized: (1) Static Threshold and Minimum Migration Time policy (THR-MMT), (2) Interquartile Range and Maximum Correlation policy (IQR-MC), (3) Median Absolute Deviation and Maximum Correlation policy (MAD-MC), (4) Robust Local Regression and Random Selection policy (LRR-RS) and (5) Local Regression and Minimum Migration Time policy (LR-MMT). The benchmarks have better performance in some aspects based on experimental verification. For example, the LR-MMT has the best performance [10], the THR-MMT has the lowest SLA V value, while IQR-MC, MAD-MC, and LRR-RS have fewer VM migrations.

For the benchmarks, the utilization threshold is set to 0.8 for all the algorithms, and the safety parameter for MAD is set to 2.5, for LRR is set to 1.2, and for IQR is set to 1.5, respectively. The parameter details of algorithms are presented in Table 2. For the sake of fairness, all the experiments are simulated several times, and the reported results are the
averaged results.

5.4 Performance Evaluation Metrics

To evaluate the efficiency of the proposed approach, four evaluation metrics are utilized: energy consumption, SLA violations (SLAV), energy-SLA violations (ESV), and the number of migrations. These metrics are also used in the Refs. [10], [13], [15] and [23]–[25].

**Energy consumption**: The datacenter’s total energy consumption is first considered. The energy consumption of physical servers mainly depends on the utilization of the CPU, memory, disk and network card. The power data of the servers used in the experiments are measured by SPEC benchmark [45].

**SLA Violations (SLAV)**: Quality of service (QoS) requirements are usually given in the form of SLAs [10]. So SLA Violations (SLAV) is a very important indicator of QoS for data centers. The SLAV metric [10] defined in Eq. (17) is utilized to measure the performance, where SLAVO refers to SLA Violations due to Overutilization and the PDM refers to SLA Performance Degradation due to Migrations as defined in Eq. (18).

\[
SLAV = SLAVO \cdot PDM \tag{17}
\]

\[
SLAVO = \frac{1}{M} \sum_{i=1}^{M} \frac{T_{S_i}}{T_{a_i}}, \quad PDM = \frac{1}{N} \sum_{i=1}^{N} \frac{C_{d_k}}{C_{r_k}} \tag{18}
\]

Where \( M \) is the number of PMs, and \( T_{S_i} \) is the total time that the PM has experienced the CPU or memory utilization of 100% leading to an SLA violation. \( T_{a_i} \) is the total time of the PM being the active state. \( N \) is the number of VMs, and \( C_{d_k} \) is the estimate of the performance degradation of the VM \( k \) caused by migrations; \( C_{r_k} \) is the total CPU capacity requested by the VM \( k \) during its lifetime. And we estimate \( C_{d_k} \) as 10% of the CPU utilization in MIPS during all migrations of the VM \( k \).

**Energy and SLA Violation (ESV)**: So as to minimize energy consumption and SLA violations, combined metric ESV [10] is employed and shown below:

\[
ESV = Energy \cdot SLAV \tag{19}
\]

**The Number of VM Migrations**: Live VM migration is a costly operation that involves the amount of CPU processing, memory blocks copy and transfer time cost, and bandwidth cost between the source and destination. And VM migration consumes non-negligible energy [10]. The more the number of migrations, the greater the negative impact on performance.

5.5 Results and Analysis

For the convenience of expression, the workload data are marked from A to K with chronological order, which is identical to Table 1.

The energy consumption metric is showed in Fig. 5. Seen from the Figure, the lower power consumption it has, the better performance of algorithm it shows. The proposed BS-MISR has the lowest energy consumption compared with the benchmark solutions. The BS-MISR has the least energy consumption while the THR-MMT has the largest energy consumption.

The energy consumption of BS-MISR reduced 32.56% to 40.13%, with an average of 36.35% reduction, compared to the LR-MMT. The energy reduction in [23] is 9.38% compared with the LR-MMT. The results demonstrate that the BS-MISR has better energy conservation compared with the benchmarks.

The ESV metric is showed in Fig. 6. The smaller ESV metric it is, the better result it has. Seen from the figure, the proposed BS-MISR, THR-MMT and the LR-MMT solutions always have smaller results. Especially, the ESV of BS-MISR reduces 28.97% to 41.48%, compared with LR-MMT. As \( ESV = Energy \cdot SLAV \), the better result of BS-MISR in the ESV mainly due to the excellent energy conservation by comparing with the SLAV metric, which is shown in Fig. 8.

The number of VM migrations is shown in Fig. 7. The more the number of migrations, the greater negative im-
The BS-MISR has the least number of VM migrations in total. The number of VM migrations of BS-MISR is smaller than the benchmarks in most of the conditions while the THR-MMT has the largest number of VM migrations on each workload day. Especially, compared with the number of VM migrations of the LR-MMT, the BS-MISR reduces 10.65% at most, 24.41% at least, with an average of 18.56% reduction. Furthermore, compared with the benchmarks, the number of VM migrations of the BS-MISR cuts down 7.06% on average, while it reduces 18.8% (compared to the LR-MMT) at most, and cuts down 2.22% at least (compared to the MAD-MC).

Figure 8 demonstrates the SLA V metric results. The smaller SLA V metric value it is, the better excellent result it has. The proposed solution BS-MISR almost has the same level SLA violations compared with LR-MMT which is the best optimization combination proposed in [10], while the THR-MMT has the best performance. The SLA VO and PDM metric are shown in the Fig. 9 and Fig. 10, respectively. Combined with the Fig. 8, we can make a conclusion that the PDM metric is the main factor which influences the SLA metric.

The explanations can be found from the perspective of BS-MISR itself. First of all, the MISR algorithm pays attention to calculate the Euclidean distance for all dimensional resources only. Secondly, the BS algorithm employs the sort mechanism to select the VM with maximum CPU request and place it to the underutilized host with higher CPU utilization. After that, the selected host is still keeping underutilized state after placement. Therefore, BS-MISR does not assure the selected VM has the minimum migration time.

As shown in Eq. (18), the PDM metric has two factors, the total CPU capacity requested by the VM during its lifetime, and the cost of migration which is estimated as 10% of the CPU utilization in MIPS. As discussed above, the BS-MISR selecting the VM do not assure the minimum migration time, which is also to say, the BS-MISR always try to select the VM with larger CPU request, so the PDM metric of BS-MISR is much higher than the MMT-style policies. The results shown in Fig. 10 can be found as an evidence.
Fig. 11  Energy consumption, ESV, SLAV and the number of VM Migrations comparison.

Fig. 12  The average number of shutdown hosts, the average time of host being active and the number of active hosts comparison.

5.6 Further Discussion and Analysis

For the sake of clarity, the normalized method is utilized to demonstrate the performance differences for the proposed BS-MISR and the benchmarks.

Figure 11 shows the comparison of the four metrics. The maximum value method (see Eq. (12)) is employed to normalize all the metrics. For all the evaluation metrics, the smaller value it is, the better result it has. Seen from the figure, the proposed BS-MISR has the smallest value in energy consumption, ESV metric and the number of VM migrations compared with the benchmarks. Only one exception, it’s the SLAV. Next, we will give a glance on the other three metrics, which are the average number of host shut down (termed as hostShutdown), the average time of host active (termed as avgTimeHostActive) and the number of active hosts (termed as activeHostNum) as shown in the Fig. 12.

In particular, with the same number of VM migrations assumption, the larger number of host shut down does not signify the better server consolidation effect. Because im-

proper server consolidation results in frequently shut down and power-on the servers. Generally, the higher average resource utilization, meanwhile, the larger the value of avgTimeHostActive, the better performance of energy saving. Furthermore, the fewer the number of activeHostNum it has, the better the effect of server consolidation.

As shown in the Fig. 12, the BS-MISR has the least activeHostNum, at the same time, it also has the least hostShutdown while the largest avgTimeHostActive, all together, they give us an explanation why it achieves better energy efficiency than the benchmarks.

6. Conclusion

In this paper, the server consolidation with the trade-off between energy and performance in the virtualized data center is studied and validated. Four important metrics of the data center, such as energy consumption, SLA violation, ESV and the number of VM migrations are considered for the VMs consolidation. First of all, the VM placement problem is addressed as a multi-capacity bin packing problem which has been proven is NP-Hard. To achieve the energy-efficiency, a novel resource-aware capacity utility model RAC is set up to guide the VM consolidation process. Moreover, an energy efficient VM selection policy named MISR based on the RAC model is proposed. Furthermore, the VM allocation policy named BS is given to optimize the consolidation. What’s more, the optimization combination BS-MISR is evaluated on the CloudSim with the real workload traces from PlanetLab. Simulation results validate that the proposed BS-MISR is reliable and can significantly reduce the energy consumption compared with the benchmarks while the BS-MISR remaining the SLA violations at a reasonable level.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant NO.61472139). Professor Chunhua GU is the corresponding author of this paper. We sincerely appreciate the anonymous reviewers’ valuable time and comments to assist us to improve this paper.

References

[1] P. Mell and T. Grance, “The NIST definition of cloud computing,” 2011.
[2] J. Koomey, “Growth in data center electricity use 2005 to 2010,” A report by Analytical Press, completed at the request of The New York Times, 2011, 9.
[3] L. Christian, “Projecting Annual new Datacenter Construction Market Size,” Technique report, Microsoft Corp., 2010.
[4] L.A. Barroso, J. Cidiras, and U. Hölzle, “The datacenter as a computer: An introduction to the design of warehouse-scale machines,” Synthesis lectures on computer architecture, vol.8, no.3, pp.1–154, 2013.
[5] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, and H. Tenhunen, “Using ant colony system to consolidate VMs for green cloud computing,” IEEE Trans. Serv. Comput., vol.8,
no.2, pp.187–198, 2015.
[6] S. Homsi, S. Liu, G.A. Chaparro-Baquero, O. Bai, S. Ren, and G. Quan, “Workload consolidation for cloud data centers with guaranteed qos using request reneging,” IEEE Trans. Parallel Distrib. Syst., vol.28, no.7, pp.2103–2116, 2017.
[7] “Pocket gems on google cloud platform,” http://cloud.google.com/customers/pocketgems/.
[8] D. Hagimont, C.M. Kamga, L. Broto, A. Tachna, and N.D. Palma, “Dvfs aware CPU credit enforcement in a virtualized system; //ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, pp.123–142, 2013.
[9] B. Song, M.M. Hassan, A. Alamri, A. A. Alealiyi, Y. Tian, M. Pathan, and A. Almogren, “A two-stage approach for task and resource management in multimedia cloud environment,” Computing, vol.98, no.1-2, pp.119–145, 2016.
[10] A. Beloglazov and R. Buyya, “Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers,” Concurrency and Computation: Practice and Experience, vol.24, no.13, pp.1397–1420, 2012.
[11] M.M. Rahman and P. Graham, “Responsive and efficient provisioning for multimedia applications,” Computers & Electrical Engineering, vol.53, pp.458–468, 2016.
[12] N. Sharma and R.M. Guddeti, “Multi-Objective Energy Efficient Virtual Machines Allocation at the Cloud Data Center,” IEEE Trans. Serv. Comput., 2016.
[13] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N.T. Hieu, and H. Tenhunen, “Energy-aware vm consolidation in cloud data centers using utilization prediction model,” IEEE Trans. Cloud Comput., 2016.
[14] A. Beloglazov and R. Buyya, “Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints,” IEEE Trans. Parallel Distrib. Syst., vol.24, no.7, pp.1366–1379, 2013.
[15] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, and H. Tenhunen, “Using ant colony system to consolidate VMs for green cloud computing,” IEEE Trans. Serv. Comput., vol.8, no.2, pp.187–198, 2015.
[16] Q. Wu, F. Ishikawa, Q. Zhu, and Y. Xia, “Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters,” IEEE Trans. Serv. Comput., 2016.
[17] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, “A multi-objective ant colony system algorithm for virtual machine placement in cloud computing.” Journal of Computer and System Sciences, vol.79, no.8, pp.1230–1242, 2013.
[18] Q. Zheng, R. Li, X. Li, N. Shah, J. Zhang, F. Tian, K.-M. Chao, and J. Li, “Virtual machine consolidated placement based on multi-objective biogeography-based optimization,” Future Generation Computer Systems, vol.54, pp.95–122, 2016.
[19] X.-K. Li, C.-H. Gu, Z.-P. Yang, and Y.-H. Chang, “Virtual machine placement strategy based on discrete firefly algorithm in cloud environments.” 2015 12th International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP), IEEE, pp.61–66, 2015.
[20] D. Bui, Y.I. Yoon, E.N. Huh, S. Jun, and S. Li, “Energy efficiency for cloud computing system based on predictive optimization.” Journal of Parallel and Distributed Computing, vol.102, pp.103–114, 2017.
[21] Z. Xiao, W. Song, and Q. Chen, “Dynamic resource allocation using virtual machines for cloud computing environment,” IEEE Trans. Parallel Distrib. Syst., vol.24, no.6, pp.1107–1117, 2013.
[22] Z. Li, C. Yan, X. Yu, and N. Yu, “Bayesian network-based Virtual Machines consolidation method,” Future Generation Computer Systems, vol.69, pp.75–87, 2017.
[23] E. Oikonomou, D. Panagiotou, and A. Rouskas, “Energy-aware Management of Virtual Machines in Cloud Data Centers,” Proc. 16th International Conference on Engineering Applications of Neural Networks (INNS), 2015.
R. Buyya, “CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” Software: Practice and Experience, vol.41, no.1, pp.23–50, 2011.

[44] K.S. Park and V.S. Pai, “CoMon: A mostly-scalable monitoring system for PlanetLab,” ACM SIGOPS Operating Systems Review, vol.40, no.1, pp.74–83, 2006.

[45] SPEC power benchmarks, Standard Performance Evaluation Corporation, http://www.spec.org/benchmarks.html#power

[46] S. Li, D. Yang, Y. Qin, and H. Zhang, “Network cross-layer mapping based in utility maximization,” Journal of Software, vol.22, no.8, pp.1855–1871, 2011.

[47] X. Shi, and K. XU, “Utility maximization model of virtual machine scheduling in cloud environment,” Chinese Journal of Computers, vol.36, no.2, pp.252–262, 2013.

Yaohui Chang received his M.S. degree in Computer Science from Northeastern University (NEU) of China in 2011. Now he is a PhD candidate of Computer Science at East China University of Science and Technology (ECUST). His current research interests include cloud computing, scheduling optimization and big data.

Chunhua Gu received his M.S. and Ph.D. degrees from East China University of Science and Technology (ECUST). Professor and PhD supervisor in the School of Information Science and Engineering, ECUST. Senior member of China Computer Federation. His research interests include cloud computing and internet of things.

Fei Luo received his M.S. and Ph.D. degrees from Huazhong University of Science and Technology (HUST) in 2004 and 2008, respectively. Associate professor in the School of Information Science and Engineering, ECUST. His main research interests include cloud computing and distributed computing.

Guisheng Fan received his M.S. and Ph.D. degrees from East China University of Science and Technology (ECUST) in 2006 and 2009, respectively. All in computer science. He is presently a research assistant of the Department of Computer Science and Engineering, ECUST. His research interests include formal methods for complex software system, service oriented computing.

Wenhao Fu received her B.S. degree from East China University of Science and Technology in 2012 in computer science. She is a Ph.D. student in computer science at East China University of Science and Technology. Her current research interests include software engineering and software fault localization.