Mollamotalebi and Hajireza *Journal of Cloud Computing: Advances, Systems and Applications* (2017) 6:16
DOI 10.1186/s13677-017-0086-z

Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH Open Access

Multi-objective dynamic management of virtual machines in cloud environments

Mahdi Mollamotalebi* and Shahnaz Hajireza

Abstract

The management strategy of a data center needs access to sufficient resources in order to handle different requests of applications, while minimizing the consumed energy. With regard to high and varying resource demands, Virtual Machines (VM) management necessitates dynamic strategies. Dynamic management of VMs includes VM placement and VM migration. The management approach presented in this paper aimed to reduce the energy consumption and the violation of service level agreements (SLA) simultaneously in data centers. The simulation results indicate that proposed approach improved the VM management 40% compared to the previous single-goal approaches based on the energy consumption and SLA violation rates.

Keywords: Cloud computing, Dynamic management, Energy consumption, SLA violation, Virtual machine placement

Introduction

Nowadays, computing tends to handle large-scale data centers and provides the resources for client applications as pay-per-use. The data center managers attempt to reduce the consumed energy while providing the required resources to user applications.

The resource allocation can be handled statically by assigning the peak number of required resources to the application. However, such allocation may lead to over-provisioning [1], which results in wasting of data centers' resources. Virtualization technology enables a physical machine to host multiple virtual machines. Each virtual machine can handle different client applications. Even if the peak amount of demanded resources is allocated to the application, still some resources of physical machine may be underutilized. Resource utilization can be improved by allocating only necessary resources to handle the typical demands. However, this may result in resource access competition between VMs in high demand conditions.

The applications with high and variable requirements necessitate frequent changes in their VMs [2] and dynamic management. By such manner of management, a running VM is migrated to another physical host (live migration). In a typical VM management solution, there are following operations as coordinated: (1) VM Placement (Allocation): the placement of a VM to a host machine in response to a VM creation request; (2) VM Relocation: the migration of VMs from a host when their overall resource requirements exceed the available resources of host; and (3) VM Consolidation: the migration of VMs from an under-utilized host so that the machine powers off in order to reduce the costs.

This paper focuses on two important issues in dynamic management of virtual machines: SLA violations reduction, and energy consumption reduction. These issues are often in contrast because reducing the energy consumption is usually reached through in-use (powered on) host reduction. But this leads potentially to more SLA violations. This is because that lower energy consumption can be achieved by placing as many VMs as possible on each host. However, sudden increasing of workload on a host may result lack of resources and subsequently SLA violations. Conversely, reducing the SLA violations typically necessitates VMs to be spread across more number of hosts, often each having a significant amount of unused resources. This manner potentially leads to more energy consumption. Designing a management strategy able to achieve both of the above

* Correspondence: mmahdi2@live.utm.my
Department of Computer, Buinzahra Branch, Islamic Azad University, Buinzahra, Iran

goals is difficult, as getting better performance considering one of the goals typically leads to degradation of performance towards the other goal. The management approaches often focus on a single goal, or prioritize the goals as a primary goal and some secondary goals [1, 3, 4, 5].

This paper proposes two single-goal approaches, Energy-Reduction to reduce the energy consumption and SLAV-Reduction to reduce the SLA violation rates. It then presents a double-goal approach, ESDR, achieving the above two goals simultaneously. The remainder of this paper is organized as follows: Related work section reviews the related work of VM Placement, VM Relocation, and VM Consolidation. The details of proposed approaches are described in The proposed management approaches section. Implementation and evaluation section presents and discusses the simulation results of the proposed approaches, and Conclusion and future work section concludes the paper.

Related work

The efficient allocation of VM to host machines while ensuring sufficient access to computing resources for applications (a requirement of quality of service), has been the subject of much attention in recent years. The reviewed approaches in literature are categorized as *static* and *dynamic* allocation. In static allocation, VMs' service requests are issued as fixed, or they are variable but a one-time mapping of some VMs into empty hosts is performed.

Cardosa et al. [6] developed a VM placement algorithm that leveraged the CPU allocation features min, max, and shares which are presented in modern hypervisors. The algorithm aimed to make balance between VMs' performance and overall energy consumption. Four techniques were provided for VM placement exploiting the inherent features of virtualization technology (Max, Min, and Share). Setting a min metric for a VM ensures that it receives at least the specified minimum amount of resources when it powered on. Also, setting a max metric for low-priority applications ensures no overflowed resource usage and keeping the resources available for high-priority applications. The metric Share enables distributing the resources between contending VMs. Lack of continuous optimization of data center during the live migration of virtual machines can be mentioned as the weakness of this algorithm.

Speitkamp and Bichler [7] proposed a static server consolidation approach that analyses the data mathematically to characterize variations of real-world workload traces. Linear programming was used for optimal mapping of VMs and host machines; however, the approach was not easily adaptable to changes of data centers' workloads.

Stillwell et al. [8] worked on mapping a set of static workloads into hosts to optimize the VMs' resource allocation in terms of fairness. They analyzed different algorithms, and indicated that a vector (or multi-dimensional) bin packing algorithm [9, 10, 11] is able to reach almost optimal result for the problem.

Bobroff et al. [1] presented the First Fit Decreasing heuristic algorithm that periodically re-calculates mapping of VMs to a set of empty hosts, based on forecasts of VMs' demands. The aim was to keep average number of active hosts as minimum while having probabilistic SLA guaranty for VMs. But the relationship between several resources such as CPU and I/O are not considered in the algorithm. It acts as static while considering variable demands to re-calculate VMs mapping periodically.

Khanna et al. [3] showed that the number of servers can be reduced in VM relocation using virtualization technology. VMs and hosts can be selected for migration by developing a mathematical optimization model. They presented a heuristic method that sorts the VMs as ascending on CPU and memory utilization to minimize the migration costs. Then hosts list is sorted as ascending on remaining capacity (i.e. available resources) to maximize resource utilization. After that, the least loaded VMs (in terms of CPU and memory usage) are migrated to the highest loaded hosts. The First Fit heuristic is used as selection algorithm with the primary goal of energy consumption reduction. The authors considered neither additional sorting strategies, nor the impact of their heuristic on the number of migrations issued.

Verma et al. [5] relied on a First Fit Decreasing heuristic that to place VMs in the most power efficient servers. The main goal was reducing the migration costs and energy consumption in data centers while meeting the QoS requirements; despite no reduction in SLA violations. The presented system is able to work in a heterogeneous environment that supports virtualization, and VMs are allocated to a physical machine in terms of consumed energy.

Wood et al. [7] presented a management system named *Sandpiper* using the First Fit Decreasing heuristic [12, 13, 14]. The hosts are sorted in an increasing order based on resource utilization, the workload is balanced among the hosts, and SLA violations are reduced subsequently. Also, monitoring and detection of hotspot servers are performed automatically.

Moreover, a new mapping to physical and virtual resources is applied and essential migration in virtualized data centers is set up.

Gmach et al. [15] developed a fuzzy logic based controller as part of their proposed management system. The controller does not distribute migrations when a host became stressed or underutilized. The appropriate host for migration is the least loaded one with enough resources to fit the VM. However, VM selection for migration is not clearly explained.

Keller et al. [16] studied variants of the First Fit heuristic to address the VM relocation problem. They showed that the order of VMs/hosts migration affects the performance metrics of data centers, such as energy consumption and SLA violations.

Beloglazov and Buyya [17] proposed Best Fit Decreasing heuristic algorithm to deal with both stressed and underutilized hosts. In the case of stressed hosts, the algorithm selects VMs with the least memory allocation to migrate, and selects as much VMs as needed in order to reduce hosts' CPU utilization to a specified value. When hosts are underutilized, all hosts' VMs are selected for migration. The selection of target host is based on a Best Fit Decreasing heuristic. The migrating VMs are sorted as descending on CPU utilization. They are then placed on the appropriate host having minimum energy consumptions.

In [18], Foster et al. attempt to switch between two single goal (minimizing SLA violations or energy consumption) strategies dynamically. They aimed to get better adaption of changing data center conditions. They proposed three strategies to handle dynamic switching between different strategies; however their situation identification process to handle the replacements is time consuming.

Zhen Xiao et al. [19] presented a system that uses virtualization technology to allocate datacenter resources dynamically based on application demands. The system is adapted to green computing by optimizing the number of servers in use. They introduced *skewness* as a metric to measure unevenness for multi-dimensional resource utilization of a server. By minimizing the skewness, different types of workloads are combined. Also, overall utilization of server resources is improved. In overall, the system is able to prevent overloads effectively while saving the energy.

Ferdaus et al. [20] addressed the resource and energy related issues in datacenters by tackling through datacenter level resource management. They focused on high resource wastage and energy consumption. To this aim, they proposed a method using the Ant Colony Optimization (ACO) metaheuristic and a vector algebra-based multi-dimensional resource utilization model. Optimizing network resource utilization is handled by an online network-aware VM cluster placement strategy to localize data traffic among VMs communication. By this manner, traffic load in data center interconnects and subsequently communication overhead in upper layer network switches are reduced.

With regard to the above review of literature, allocation of VMs to physical machines while keeping the access to computing resources as efficient is handled by many of recent researches. The provided allocation approaches are categorized as *static* and *dynamic*. Most of the algorithms aimed to make balance between VMs' performance and overall energy consumption. They try to allocate the minimum amount of resources to low-priority applications and keep the resources available for high-priority ones. With regard to the dynamic nature of cloud environments, some approaches recalculate mapping of VMs to a set of empty hosts periodically based on VMs' demand forecasts. Such approaches are able to minimize the number of active hosts while keeping SLA for VMs in a reasonable level. Choosing VMs and hosts for migration can be performed by a mathematical optimization model. Also, heuristic methods are used to sort VMs based on CPU/memory utilization in order to minimize the migrations and reduce the consumed energy.

The dynamic behavior of cloud environment necessitates having a dynamic and adapted strategy to switch between SLA violation and energy consumption dynamically in VM migrations. Most of the existing algorithms have focused on one aspect of efficiency. This paper provides three dynamic approaches (first one to reduce the energy consumption, the second one to reduce SLA violations, and third one to handle both of energy and SLA violations simultaneously) to manage the virtual machines in data centers. With regard to changes of data center's state, the provided double-goal approach switches between different policies at run-time if it is necessary. SLA violation and power efficiency are measured in order to decide for activate the appropriate approach.

The proposed management approaches

With regard to the dynamic nature of virtualized data centers, designing a management system with different goals is a challenging task. In this section, three dynamic approaches are proposed to manage the virtual machines in data centers. In the following, first the terms and metrics of management approaches are introduced. Then, two single-goal approaches (one to reduce the energy consumption, and another to reduce SLA violations) are proposed. After that, a double-goal approach is proposed achieving the above two goals simultaneously.

i) The status *SLA violation* occurs when resources required by a VM are not available. This status subsequently leads to performance degradation. It is typically defined by the percentage of a VM's required CPU time which is not available currently. ii) The *overall utilization* of a data center is the percentage of CPU capacity that is currently in use by data center. Iii) A limited capacity of CPU's processing capabilities is specified as *CPU_Shares*. In our work, the CPU_Shares assigned to each core, is related to its frequency. For example, the CPU_Shares for a 1GHZ CPU is assigned to 1000. iv) For a host h, the *power efficiency*, per_h, refers to in-use processing per consumed energy, and it is measured by CPU-shares-per-watt (CPU/W). The power efficiency of a single host is calculated by Eq. 1:

$$per_h = \frac{\Phi_h}{\Psi_h} \tag{1}$$

Where Φ_h is the number of CPU_Shares currently is in-use across all cores in the host, and Ψ_h refers to current power consumption of the host in watts. Moreover, an active host consumes a significant amount of power even if it has little or no CPU load (i.e., very low power efficiency). Equation 2 calculates the power efficiency for the entire data center, per_{dc}:

$$per_{dc} = \frac{\Sigma_{\mathrm{h}\in\mathrm{hosts}}\Phi_\mathrm{h}}{\Sigma_{\mathrm{h}\in\mathrm{hosts}}\Psi_\mathrm{h}} \tag{2}$$

Such that *hosts* indicates a collection of all hosts in data center. v) *Maximum Power Efficiency* represents the best amount of power efficiency a host can achieve. It is calculated as power efficiency of the host at maximum CPU utilization. vi) *HostUtilization* refers to the average amount of CPU utilization for all hosts in the state "On". The higher value of host utilization leads to more efficient usage of resources and energy consumption reduction.

This paper supposes that each time a management operation takes place, hosts are classified to three different power states: *power on, power suspended, and power off.* The powered on hosts are further classified to *stressed*, *partially-utilized*, *under-utilized,* and *empty* based on their CPU utilization. The state of hosts may be changed according to workload changes of hosted VMs, or migrations performed by management operations. To do this, two threshold values are used as $stress_{cpu}$ and min_{cpu}. The classification is carried out based on average CPU utilization of hosts. Hosts with average CPU utilization ranged in ($stress_{cpu}$, 1], (min_{cpu}, $stress_{cpu}$], and [0, min_{cpu}) are supposed as stressed, partially-utilized, and under-utilized hosts respectively. Moreover, hosts with no assigned VMs are empty ones; hosts in suspended or power off state are also included in this category.

Energy-reduction and SLAV-reduction approaches

Energy-Reduction and SLAV-Reduction are two single-goal candidates. In the next subsections, the process of VM Placement, VM Relocation and VM Consolidation policies that form the above approaches are explained. The placement management operation runs each time a new VM creation request is received, and selects a host in which to instantiate the VM. Algorithm 1 shows the process of virtual machine placement policy for the proposed Energy-Reduction approach.

Algorithm 1 VM Placement process for Energy-Reduction approach

```
Data: H,VM
1: $H^{!}, H^{+}, H^{-}, H^{\phi}$ ←classify(H)
2: targets ← combine(sort($H^{+}$), sort($H^{-}$), sort($H^{\phi}$))
3: for $h_t \in$ targets do
4:     if has_capacity($h_t$, VM) then
5:         place (VM, $h_t$)
6:         break
7: end
```

The VM Placement policy for Energy-Reduction approach (see Algorithm 1), first classifies *hosts* in appropriate categories (line 1) as stressed ($H^{!}$), partially-utilized (H^{+}), under-utilized (H^{-}) or empty (H^{Φ}). Each host category is then sorted (lines 2). H^{+}, and H^{-} are sorted as descending on maximum power efficiency and CPU utilization respectively. The category H^{Φ} is also sorted as descending on maximum power efficiency and power state respectively. Such sorting ensures focus of placement on power efficiency. Then a list of *targets* hosts is prepared by concatenating *(sort (H^{+}), sort (H^{-}), sort (H^{Φ})).* Finally, following a First Fit approach, the policy assigns VM to the first host in *targets* with enough capacity (lines 3-6). The method *has_capacity (h_t, VM)* checks the ability of host to meet resource requirements indicated in the VM creation request (line 4) without becoming stressed.

The VM Placement policy for SLAV-Reduction approach differs from the Energy-Reduction policy in the way that H^{+} and H^{-} are sorted. H^{+} and H^{-} are sorted as ascending and descending respectively on CPU utilization and then maximum power efficiency. Such sorting causes the placement to be focused on distributing the load across the hosts, and leaving spare resources to handle spikes in resource demand over other considerations.

The VM relocation operation runs frequently over short intervals of time in order to detect stress situations as soon as possible. For both approaches, the interval is set to 10 min. The VM relocation determines which hosts are prone to be in a stress situation. Then it removes such hosts by migrating one VM from a stressed host into a non-stressed one. Algorithm 2 shows the process of virtual machine relocation policy for the proposed approach Energy-Reduction.

Algorithm 2 VM Relocation process for Energy-Reduction approach

```
Data: H
1: $H^{!}, H^{+}, H^{-}, H^{\phi}$ ←classify(H)
2: sources ← sort ($H^{!}$)
3: targets ← combine (sort ($H^{+}$), sort ($H^{-}$), sort ($H^{\phi}$))
4: for h ∈ sources do
5:       success ← FALSE
6:      VM-candidates ← sort($Vm_h$)
7:      i = 1
8:      for $v_i$ ∈VM-candidates do
9:           for $h_t$ ∈ targets do
10:               if has_capacity ($h_t$ , VM) then
11:                  migrate ( h , $v_i$ , $h_t$ )
12:                  i = i+1
13:              success ← TRUE
14:             break
15:        if success then break
16: end
```

As shown in Algorithm 2, the VM Relocation policy for Energy-Reduction approach first classifies *hosts* to appropriate categories (line 1) performing a *stress check* on all hosts. A host is stressed if its CPU utilization has remained above the $stress_{cpu}$ threshold throughout the last CPU load monitoring window. The hosts are categorized as stressed ($H^{!}$), partially-utilized (H^{+}), under-utilized (H^{-}), or empty (H^{ϕ}). Then each host category is sorted (line 3). $H^{!}$ is sorted as descending on CPU utilization, H^{+} and H^{-} *are* sorted as descending on maximum power efficiency and CPU utilization respectively. H^{ϕ} is sorted as descending on maximum power efficiency and power state respectively. After that, a list of *targets* hosts is formed by concatenating *(sort (H^{+}), sort (H^{-}), sort (H^{ϕ})).* Following the First Fit heuristic, one VM is chosen from each host *h* in *sources* and a corresponding *host* in *targets* to which one to migrate VM (lines 4-13). For each host *h* in *sources* (which is stressed), VMs with less CPU load are filtered. The remaining VMs then are sorted as ascending on CPU load (line 6). After that, all VMs are sorted as descending on CPU load, and finally the migration process is launched (line 10).

VM Relocation policy for SLAV-Reduction approach differs from Energy-Reduction in the way of sorting $H^{!}$ and H^{-} ($H^{!}$ is sorted as ascending on CPU utilization and maximum Energy-Reduction efficiency respectively. H^{+} is sorted as descending on CPU utilization and maximum power efficiency respectively). In addition, SLAV-Reduction policy performs a different stress check such that a host is considered as stressed if its last two monitored CPU load values are more than $stress_{cpu}$ threshold, or its average CPU utilization throughout the last CPU load monitoring window exceeds $stress_{cpu}$.

The purpose of VM consolidation policy is to control the load that VM Placement and VM Relocation have distributed across the data center. This is performed by migrating VMs from under-utilized hosts (through suspending or powering them off) to partially-utilized ones. This management operation runs less frequently than VM Relocation. The interval is set to 1 and 4 h for Energy-Reduction and SLAV-Reduction strategies respectively. Algorithm 3 shows the steps of virtual machine consolidation process for the proposed approach Energy-Reduction.

Algorithm 3 VM Consolidation process for Energy-Reduction approach

```
Data: H
1: $H^{!}, H^{+}, H^{-}, H^{\phi}$ ←classify(H)
2: :sources ← sort ($H^{-}$)
3: targets ← combine (sort ($H^{+}$), sort ($H^{-}$))
4:   usedTargets ← Ø
5:   usedSources ← Ø
6:   for h ∈ sources do
7:  if h ∈usedTargetsthen continue
8:  for v ∈ $VM_h$ do
9:       for $h_t$ ∈ targets do
10:            if hasCapacity($h_t$ , v) & ∧$h_t$≠ h & ∧$h_t$∉usedSourcesthen
11:                  usedTargets←usedTargets ∪ $h_t$
12:                  usedSources←usedSources ∪ h
13:                  migrate ( h , v , $h_t$ )
14:                breake
15: end
```

As shown in Algorithm 3, the VM consolidation policy for Energy-Reduction approach first classifies *hosts* (line 1) as stressed ($H^{!}$), partially-utilized (H^{+}), underutilized (H^{-}), and power off *(H^{ϕ}).* Host machines in Power-off state are grouped as Empty. The policy then sorts H^{+} and H^{-} as descending on power efficiency and CPU utilization respectively, and

Table 1 The characteristics of used workload patterns

The workload pattern	Description	Monitoring time period (sec)	Measurement time period (sec)
ClarkNet	Based on the Log file prepared from the access to ClarkNet during two weeks	1,209,400	100
EPA	The file prepared from web Log based on EPA during one day	86,200	100
SDSC	The file prepared from Log center of Santiago supercomputers during one day	86,200	100
Google cluster data	Tracking 7 h the workload of Google cluster with different jobs	22,200	300

creates new lists. Then it forms a list of *targets* hosts by concatenating *sort* (H^+)*, sort* (H^-) (line 3). Afterwards, H^- is sorted again this time as ascending on power efficiency and CPU utilization respectively. Using a First Fit heuristic, the policy attempts to release all hosts *h* in *sources* by migration of their VMs to *hosts* in *targets* (lines 6-14). For each host *h* in *sources*, the policy sorts its VMs as descending on overall resource capacity (memory, number of CPU cores, and core capacity) and CPU load respectively. It is necessary to avoid using a host both as source and target for migrations.

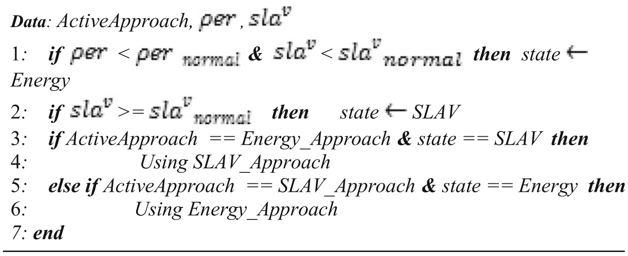
The VM Consolidation policy for SLAV-Reduction approach differs from Energy-Reduction in the way that H^+ and H^- are sorted such that first, H^+ is sorted as ascending on CPU utilization and maximum power efficiency respectively. Also, $H^-$is sorted as descending on CPU utilization and maximum power efficiency respectively. Then, H^- is sorted as ascending on CPU utilization to form the source host machines.

The approaches use different values for $stress_{cpu}$ threshold: Energy-Reduction uses 95% and SLAV-Reduction uses 85%. The lower threshold for the SLAV-Reduction approach allows additional resources to be available for workload variations. Both strategies use the min_{cpu} threshold of 60%. Selected values for the above thresholds have been obtained experimentally and based on the average CPU utilization of host machines. The characteristics of workload patterns are shown in Table 1.

Energy-reduction and SLAV-reduction dynamic run-time replacement (ESDR)

ESDR attempts to meet two objectives simultaneously. With regard to changes of data center's state, ESDR switches between different policies at run-time if it is necessary. It checks data center metrics monitored during different executions in order to determine if current in-use approach (*ActiveApproach*) needs changes. It uses SLA violation and power efficiency ratio metrics to evaluate whether the active approach should be switched. The power efficiency ratio is calculated as the ratio of optimal power efficiency to current power efficiency [21]. The switching approach is triggered when the metric's value related to the goal of the active approach (i.e. sla^v for the SLA violation, *per* for the Power efficiency ratio) is less than the threshold value (sla^v_{normal} or per_{normal}), while the metric related to inactive approach exceeds the threshold value. Algorithm 4 shows the steps used to change active approach in ESDR.

Algorithm 4 Conditions to change the active approach in ESDR



```
Data: ActiveApproach, $per$, $sla^v$
if $per < per_{normal}$ & $sla^v < sla^v_{normal}$ then state ← Energy
if $sla^v >= sla^v_{normal}$ then state ← SLAV
if ActiveApproach == Energy_Approach & state == SLAV then
        Using SLAV_Approach
else if ActiveApproach == SLAV_Approach & state == Energy then
        Using Energy_Approach
end
```

The switching used in ESDR allows the data center to respond to a situation in which performance in one metric has deteriorated. Table 2 shows the default value of ESDR's parameters.

As shown in Table 2, choosing the best running interval as well threshold values (sla^v_{normal} |per_{normal}) is performed based on the results of Energy-Reduction and SLAV-Reduction experiments. The default values are

Table 2 Default values of ESDR's parameters

Parameter	Normal Value	Description	Calculation method
per_{normal}	71.817	Normal value for power efficiency threshold	The average value of energy efficiency in both Energy-Reduction and SLAV-Reduction approaches
sla^v_{normal}	1.644	Normal value for SLA violations threshold	The average value of SLA violation in both Energy-Reduction and SLAV-Reduction approaches
Interval Running Approach	1 h	Time interval for running ESDR approach	—

Table 3 The features of hosts used in data centers

Hosts	Number of CPUs	Number of cores	Min CPU share	Max CPU share	Power consumption at 100% efficiency	Maximum power efficiency	Memory capacity
Large	2	4	2500	20,000	W233	85.84	GB16
Small	2	2	3000	12,000	W258	46.51	GB8

obtained after 12 iterations of Energy-Reduction and SLAV-Reduction process with 5 randomly generated workload patterns. The average of 30% least values of consumed energy and SLA violations are considered as the default values.

Implementation and evaluation

In order to evaluate the performance of proposed algorithms, they are implemented in DCSim [22] simulation environment which is found as a common and efficient simulation tool in the literature. Two metrics, power efficiency (*per*) and SLA violation (sla^v), are considered to evaluate the performance. Making decision only based on the above metrics is difficult because if one approach performs well with respect to SLA violations at the expense of high power (and vice versa), it is hardly possible to conclude about the preferable approach [18, 23]. The decision depends on relative changes in each area as well as the importance assigned to each metric by data center operators according to their business objectives, and costs of energy and SLA violations.

To measure the performance of proposed double-goal approach, an experiment is planned such that the Energy-Reduction and SLAV-Reduction approaches are used as benchmarks. SLAV-Reduction approach provides the bounds for the best SLA violation (sla^v_{best} = $sla^v_{SLAV\text{-}Reduction}$) and the worst power efficiency (per_{worst} = $per_{SLAV\text{-}Reduction}$). The Energy-Reduction approach provides the bounds for worst SLA violation (sla^v_{worst} = $sla^v_{Energy\text{-}Reduction}$) and best power efficiency (per_{best} = $per_{Energy\text{-}Reduction}$). The values for selected approach i (Energy-Reduction, SLAV-Reduction or ESDR), are then used to create the normalized vector v_i, represented as [per_{norm}, sla_{norm}]. The values of sla^v_{norm} and per_{norm} are calculated through the following equations [18]:

$$sla^v{}_{norm} = \frac{sla^v{}_i - sla^v{}_{best}}{sla^v{}_{worst} - sla^v{}_{best}} \tag{3}$$

$$per_{norm} = \frac{per_{best} - per_i}{per_{best} - per_{worst}} \tag{4}$$

$$v_i = (per_{norm}, sla^v{}_{norm}) \tag{5}$$

per_{norm} and sla^v_{norm} indicate normalized power efficiency and normalized SLA violation respectively. It is noteworthy that per_{best} > per_{worst}, but sla^v_{best} < sla^v_{worst}. By having the normalized vector v_i, it is possible to calculate *L^2-norm* and use it as an overall score (*Score$_i$*).

$$Score_i = |vi| = \sqrt{sla^v{}_{norm}{}^2 = + \; per_{norm}{}^2} \tag{6}$$

Equation 6 calculates the score of selected approach i. Lower value for the score is considered as the better one, as it is construed as a smaller distance to the *best* bounds of each metric (defined by sla^v_{best} and per_{best}). The Energy-Reduction and SLAV-Reduction approaches always achieve the score 1, as each one achieves the *best* score for one metric and the *worst* score for the other one. Scores less than 1 indicate that overall performance of the candidate approach has improved relative to the baseline approaches. Different workload patterns of data center are considered, and the average amount of all tests' score is calculated to be used in comparisons.

Experimental setup for evaluation

In order to evaluate the proposed approaches, they are simulated along with similar latest ones using DCSim [22] that is an extensible simulator able to model the multi-tenant virtualized data centers efficiently. Also, DCSim includes virtualization features such as CPU scheduling used in modern Hypervisors, resource allocation and virtual machine migrations. Moreover, it is able to model a continuous and interactive workload that is necessary in experiments.

The simulated data center consists of 200 host machines of two types (*small* and *large*). The *small* hosts are considered as HPProLiant DL380G5, with 2 dual-core 3 GHz CPUs and 8 GB of memory. The *large* hosts are considered as HP ProLiantDL160G5, with 2 quad-core 2.5 GHz CPUs and 16 GB of

Table 4 Properties of virtual machines in the data center

Type of VMs	Number of virtual cores	Min CPUShare	Memory capacity
Small	1	1500	512 MB
Medium	1	2500	512 MB
Large	2	2500	1 GB

Table 5 The best values of ESDR parameters

Parameter	Value	Description
per_{normal}	71.817	Normal value of power efficiency threshold
sla^{v}_{normal}	1.644	Normal value of SLA Violation threshold

memory. Table 3 shows detailed features of the above mentioned hosts used in performed experiments.

The power consumption of both hosts is calculated by SPECPower benchmark [24] and results indicate that maximum power efficiency of *large* hosts (85.84 CPU/W) doubles the *small* host ones' (46.51 CPU/W). Three different sizes are considered for virtual machines in simulated data center as shown in Table 4.

The hosts are able to use a work-conserving CPU scheduler which is available in virtualization technology. This means that each shared CPU that is not used by a VM can be used by other ones. For CPU contention, the shares are assigned to VMs as round-robin until all the shares be allocated. The metrics used by management policies (e.g. host CPU utilization and SLA violation) are measured every 2 min from each host, and evaluated by the policy over a sliding window of 5 measurements.

Workload pattern

Data centers experience a highly dynamic workload, driven by frequent VMs' arrivals/departures, as well as resource requirements of VMs. In this paper, to evaluate the proposed approach, random workload patterns are generated each one having a set of VMs. The VMs are assigned a specific start/stop time and a dynamic trace-driven resource requirement. Each VM is driven by one of the individual traces (*ClarkNet*, *EPA*, and *SDSC* [25]), and two different job types from the *Google Cluster Data* trace [26]. For each workload pattern, incing requests are calculated with 100 s interval. The requests rate is used to define current workload of each VM. The CPU requirement of each VM is calculated through a linear function of current input rate. Each VM starts its trace with a random selected offset time.

The number of VMs in data centers varies during simulation process frequently to form dynamicity of the environment. Within first 40 h of activity, 600 VMs are created and remain as running throughout simulation period in order to keep at least a low level of loads existing. After 2 days of simulation, the rate of new arrival VMs begins to change, and it stops changing after about 1 day. The arrivals are generated as one per day. The total number of VMs in data center is set by a random generated number with normal distribution between the values of 600 and 1600. The reason to choose value 1600 for maximum number of VMs is that beyond this value, the SLAV-Reduction denies admission of some incoming VMs due to insufficient available resources. The simulation continues for 10 days and then the experiment finishes.

Evaluation of per_{normal} and sla^{v}_{normal} in ESDR

Switching between two approaches Energy-Reduction and SLAV-Reduction in ESDR occurs when the metrics related to the goal of active approach (sla^{v} for SLA violations or *per* for power efficiency) is within the normal/acceptable range considering the threshold values (sla^{v}_{normal} or per_{normal}). But the metrics of non-active approach exceeds its normal/acceptable threshold value. ESDR uses $\{sla^{v}_{normal}|per_{normal}\}$threshold values. These values are derived from the experiments performed on different workload patterns for Energy-Reduction and SLAV-Reduction approaches. Table 5 shows the best values of above mentioned parameters.

Discussion and evaluation of the experimental results

The experiments with the same workload patterns are repeated five times for each proposed approach and the average values of results are obtained. Table 6 shows the results of five different workload patterns for each approach.

The columns of Tables 6 are evaluation metrics introduced in Section III-I. The results show measured values of different metrics as well the reported normal values of Energy-Reduction and SLAV-Reduction.

As shown in Fig. 1, the highest and lowest host utilization values result from Energy-Reduction and SLAV-Reduction approaches respectively.

Given that the Energy-Reduction and SLAV-Reduction approaches use different values for $stress_{cpu}$ threshold (95% in Energy-Reduction approach, and 85% in SLAV-Reduction approach), lower threshold allows SLAV-Reduction approach access to more resources while

Table 6 The results of 5 workload patterns for each approach

Approach	Host Unit	Power (KWH)	PwrEff (CPU/W)	SLA	Migration	sla^{v}_{normal}	per_{normal}	Score
SLAV-Reduction	83.505	4531.190	68.811	1.403	57,459	0	1	1
Energy-Reduction	84.686	4094.201	74.695	2.087	37,863	1	0	1
ESDR	84.583	4404.442	70.712	1.476	50,417	0.307	0.514	0.599

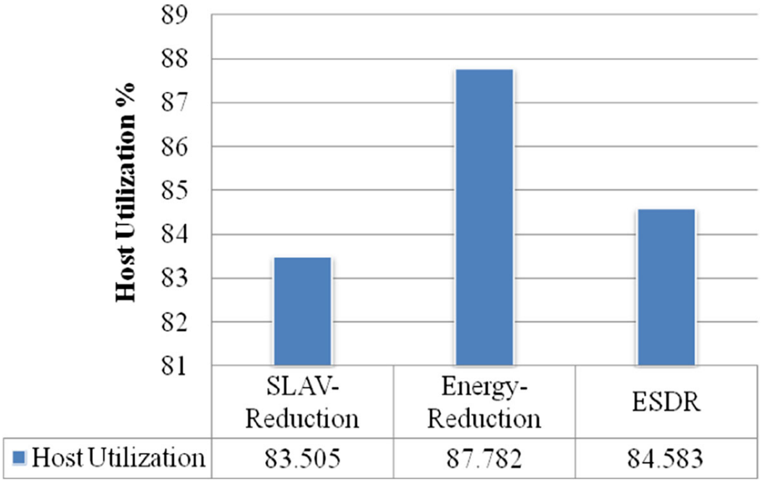


Fig. 1 Comparison of Host Utilization metric in different approaches

workload changes. This subsequently leads to low utilization of the active host. Figure 2 and Fig. 3 compare the metrics power consumption and power efficiency for different approaches.

With regard to primary goal of SLAV-Reduction and Energy-Reduction approaches which are reducing SLA violations and increasing energy efficiency respectively, SLAV-Reduction approach consumes the highest amount of power (its power efficiency is low) and Energy-Reduction approach acts as contrary. Figure 4 compares the SLA violations of three proposed approaches.

Corresponding to power consumption results, Fig. 4 shows that SLAV-Reduction approach has the lowest amount of SLA violations while Energy-Reduction approach suffers from highest amounts of violations. The number of migrations for different approaches is compared in Fig. 5.

The results indicate that SLAV-Reduction approach caused by the highest number of virtual machine migrations while Energy-Reduction approach gained the least ones. SLAV-Reduction approach attempts to reduce the SLA violations through migration strategies, and thus the resource requests of VMs rarely fail. On the other hand, Energy-Reduction approach avoids migrations as much as possible in order to increase the hosts' utilization.

The migration overhead and its effects on SLA violation and host utilization are also investigated. Switching between Energy-Reduction and SLAV-Reduction approaches with different stress thresholds increases the migrations from ESDR to Energy-

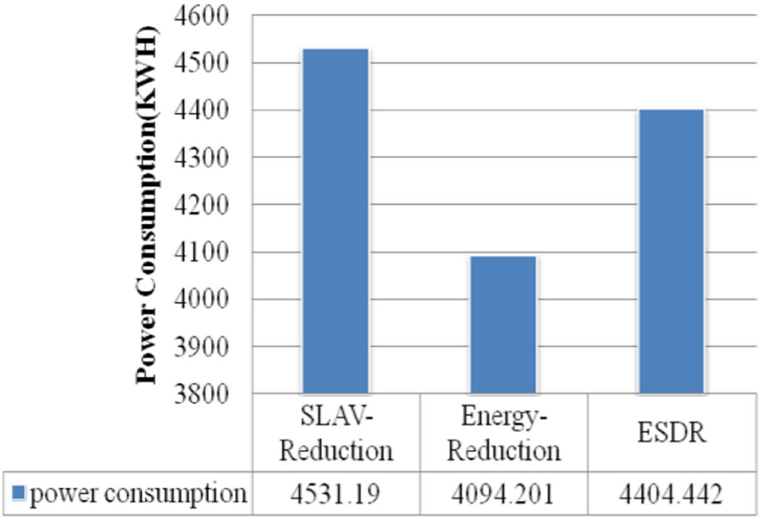


Fig. 2 Comparison of power consumption metric in different approaches

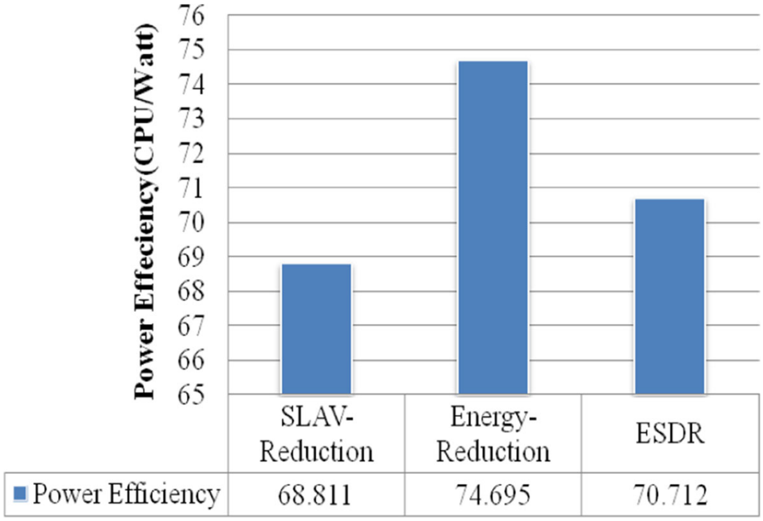


Fig. 3 Comparison of power efficiency in different approaches

Reduction as a side-effect. The value of stressed threshold varies in different approaches. The Energy-Reduction approach uses a large number of hosts efficiently while considering the stress threshold. It switches to SLAV-Reduction approach close the stress threshold, causing a large number of hosts become stressed and migrations increasing.

In order to evaluate the success rate of ESDR approach considering both goals (SLA violation reduction and power efficiency), a new metric is introduced named as score (see Eq. 6). The values obtained for score of different approaches are compared in Fig. 6.

As shown in Fig. 6, double-goal ESDR approach has achieved fewer score compared to single-goal SLAV-Reduction and Energy-Reduction approaches. This means difference reduction for best values of SLA violation and power efficiency. The graphical representation vectors of score can be also useful to show and analyze the results. L2-Norm (Euclidean) vector representation in two-dimensional space is as the unit circle which is shown in Fig. 7.

Both Energy-Reduction and SLAV-Reduction approaches that have been selected as benchmark, gained the score 1. The smaller radius of the circle gets better results because it means that the shorter distance is created for the best values of SLA violation and power efficiency. As shown in Fig. 7, ESDR has gained better score than other single-goal approaches, Energy-Reduction and SLAV-Reduction.

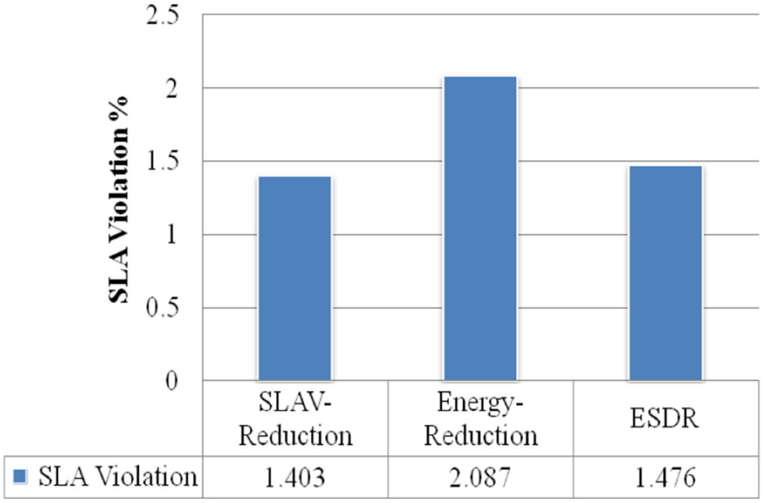


Fig. 4 Comparison of SLA violations for different approaches

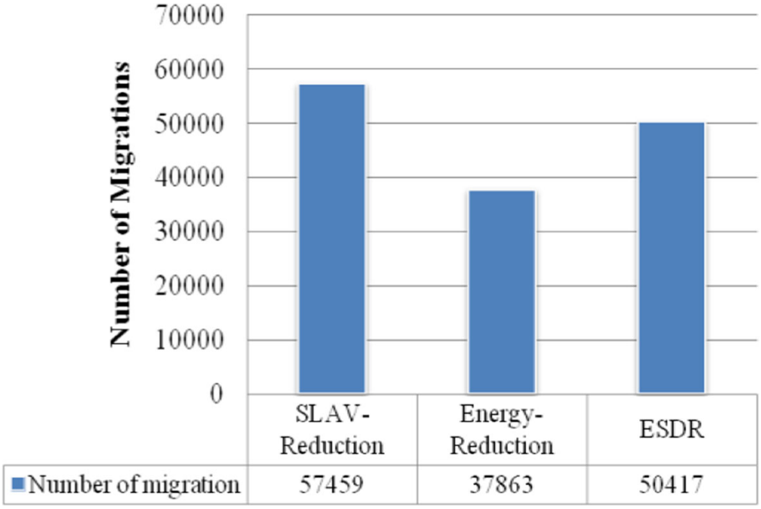


Fig. 5 Comparison of the number of migrations in different approaches

The ESDR improved the score about 40% compared to Energy-Reduction and SLAV-Reduction. This is due to regular calculations of SLA violations and power efficiencies at specified intervals and then, comparing them with per_{normal} *and* sla^{v}_{normal} parameters.

Conclusion and future work

This paper aimed to manage virtual machines dynamically which is useful in cloud environment data centers. The provided approaches handle both major goals of dynamic management in data centers, maximizing power efficiency and minimizing SLA violations, considering the inherent trade-off between these goals. It is difficult to manage data centers with conflicting goals simultaneously. It becomes worse by lack of an efficient method to do a straightforward comparison based on various metrics. This paper provided two single-goal approaches Energy-Reduction and SLAV-Reduction, and also ESDR as a double-goal approach. The double-goal approach reduces the consumed energy and SLA violations simultaneously.

All approaches are experimented in the same simulation conditions. The experimental results indicated that ESDR handles the mentioned goals more effectively compared to other approaches. ESDR improved the score about 40% compared to Energy-Reduction and SLAV-Reduction.

Some directions can be chosen as future work. One possible plan is to focus on a management approach able to switch between two single-goal approaches considering entire efficiency of data center and according to its current workload. Moreover, this paper relied only on

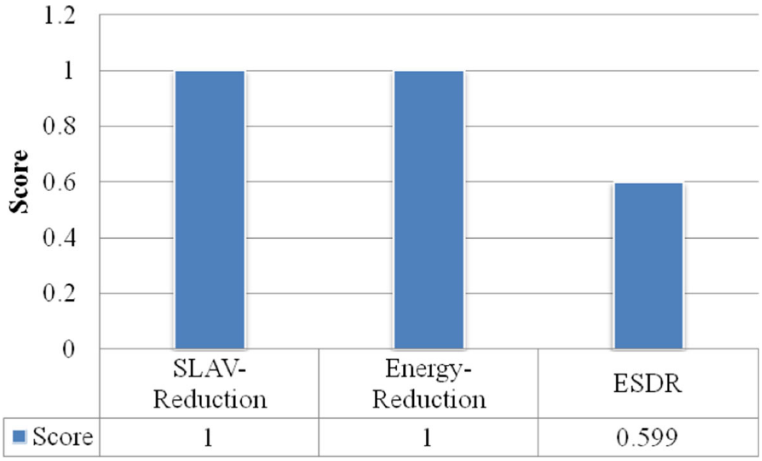


Fig. 6 Comparison of score in different approaches

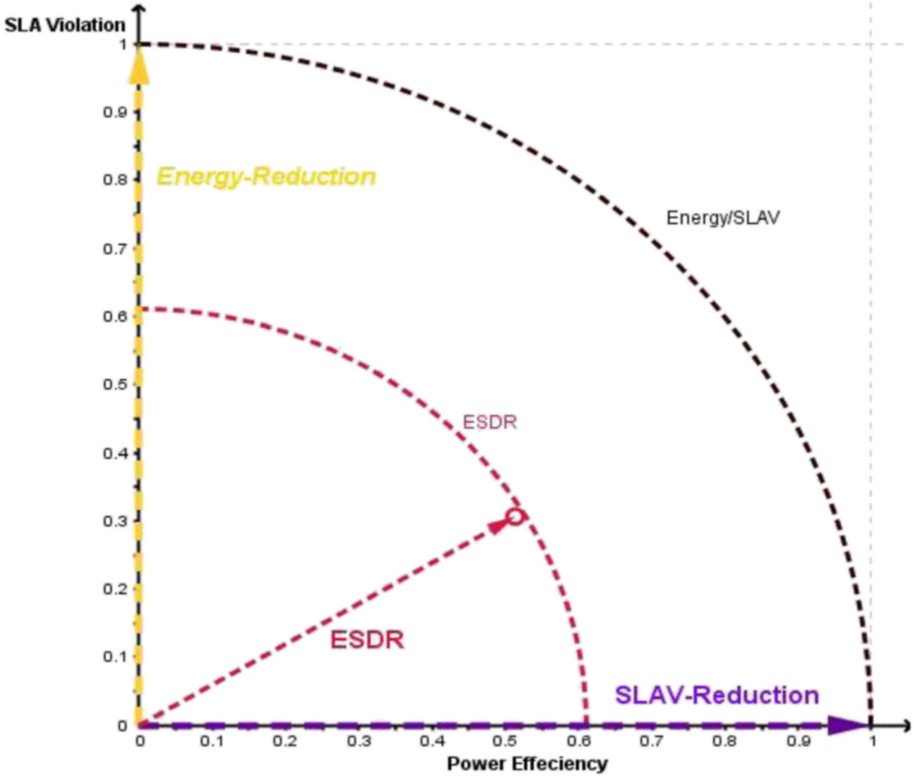


Fig. 7 Euclidean vector representation of scores for different approaches

CPU loads to measure host or VM loads. In future works, the memory and bandwidth loads can be taken into account additionally.

Acknowledgements
We are also thankful to anonymous reviewers for their valuable feedback and comments for improving the quality of the manuscript.

Authors' contributions
This research work is part of SH (second author) dissertation work. The work has been primarily conducted by SH under the supervision of MM (first author). Extensive discussions about the algorithms and techniques presented in this paper were carried between the two authors over the past year. Both authors read and approved the final manuscript.

Authors information
Mahdi Mollamotalebi is an Assistant Professor in the Department of Computer, Buinzahra branch, Islamic Azad University, Buinzahra, Iran. He received his Ph.D. from the University Teknology Malaysia (UTM) in the area of Grid computing resource discovery. He has authored and coauthored several technical refereed and non-refereed papers in various conferences, journal articles, and book chapters in research and pedagogical techniques. His research interests include parallel and distributed system performance, Grid and Cloud computing, and IoT. Dr. Mollamotalebi is a member of IEEE. Shahnaz Hajireza is pursuing towards his M.Sc., Department of Computer, Buinzahra branch, Islamic Azad University, Buinzahra, Iran. Shahnaz's interests are in Cloud computing, Virtualization, and Cloud localization. She has authored and coauthored some technical papers in various conferences, and journal articles.

Competing interests
The authors declare that they have no competing interests.

Received: 6 November 2016 Accepted: 27 June 2017
Published online: 12 July 2017

References
1. Bobroff, N., Kochut, A., & Beaty, K. (2007). Dynamic placement of virtual machines for managing sla violations. In Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on (pp. 119-128). IEEE
2. Kochut, A., & Beaty, K. (2007). On strategies for dynamic resource management in virtualized server environments. In Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS'07. 15th International Symposium on (pp. 193-200). IEEE
3. Khanna, G., Beaty, K., Kar, G., & Kochut, A. (2006). Application performance management in virtualized server environments. In Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP (pp. 373-381). IEEE
4. Wood T, Shenoy PJ, Venkataramani A, Yousif MS (2007) Black-box and gray-box strategies for virtual machine migration. In: NSDI Vol. 7, pp 17–17
5. Verma, A., Ahuja, P., & Neogi, A. (2008). pMapper: power and migration cost aware application placement in virtualized systems. In Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware (pp. 243-264). Springer-Verlag New York, Inc.
6. Cardosa, M., Korupolu, M. R., & Singh, A. (2009). Shares and utilities based power consolidation in virtualized server environments. In Integrated Network Management, 2009. IM'09. IFIP/IEEE International Symposium on (pp. 327-334). IEEE
7. Speitkamp B, Bichler M (2010) A mathematical programming approach for server consolidation problems in virtualized data centers. IEEE Trans Serv Comput 3(4):266–278
8. Stillwell M, Schanzenbach D, Vivien F, Casanova H (2010) Resource allocation algorithms for virtualized service hosting platforms. J Parallel Distributed Comput 70(9):962–974
9. Panigrahy, R., Talwar, K., Uyeda, L., & Wieder, U. (2011). Heuristics for vector bin packing. Research. Microsoft. Com

10. Kou LT, Markowsky G (1977) Multidimensional bin packing algorithms. IBM J Res Dev 21(5):443–448
11. Frenk H, Csirik J, Labbé M, Zhang S (1990) On the multidimensional vector bin packing. University of Szeged. Acta Cybernetica, pp 361–369
12. Yue M (1991) A simple proof of the inequality FFD (L)≤ 11/9 OPT (L)+ 1,∀ L for the FFD bin-packing algorithm. Acta Mathematicae Applicatae Sinica (English Series) 7(4):321–331
13. Dósa G (2007) The tight bound of first fit decreasing bin-packing algorithm is FFD (I)≤ 11/9OPT (I)+ 6/9. In: Combinatorics, algorithms, probabilistic and experimental methodologies. Springer, Berlin Heidelberg, pp 1–11
14. Kao, M. Y. (Ed.). (2008). Encyclopedia of algorithms. Springer Science & Business Media. Springer Berlin Heidelberg. doi:10.1007/978-3-642-27848-8.
15. Gmach, D., Rolia, J., Cherkasova, L., Belrose, G., Turicchi, T., & Kemper, A. (2008). An integrated approach to resource pool management: Policies, efficiency and quality metrics. In Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on (pp. 326-335). IEEE
16. Keller, G., Tighe, M., Lutfiyya, H., & Bauer, M. (2012). An analysis of first fit heuristics for the virtual machine relocation problem. In Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualiztion management (svm) (pp. 406-413). IEEE
17. 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. Concurrency Comput 24(13):1397–1420
18. Foster, G., Keller, G., Tighe, M., Lutfiyya, H., & Bauer, M. (2013). The right tool for the job: Switching data centre management strategies at runtime. In Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on (pp. 151-159). IEEE
19. Xiao Z, Song W, Chen Q (2013) Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Trans Parallel Distributed Syst 24(6):1107–1117
20. Ferdaus, M. H. (2016). Multi-objective virtual machine Management in Cloud Data Centers (Doctoral dissertation, Monash University)
21. Tighe, M., Keller, G., Shamy, J., Bauer, M., & Lutfiyya, H. (2013). Towards an improved data centre simulation with DCSim. In Network and Service Management (CNSM), 2013 9th International Conference on (pp. 364-372). IEEE
22. Tighe, M., Keller, G., Bauer, M., & Lutfiyya, H. (2012). DCSim: A data centre simulation tool for evaluating dynamic virtualized resource management. In Network and service management (cnsm), 2012 8th international conference and 2012 workshop on systems virtualiztion management (svm) (pp. 385-392). IEEE
23. Foster, G. (2013). UTIL-DSS: utilization-based dynamic strategy switching for improvement in data Centre operation (Doctoral dissertation, The University of Western Ontario)
24. (2014) SPECpower ssj2008. Standard Performance Evaluation Corporation. Available on: https://www.spec.org/power_ssj2008/. Accessed 7 July 2017.
25. (2014) The Internet Traffic Archive. Available on: http://ita.ee.lbl.gov/. Accessed 7 July 2017.
26. (2014) Google Cluster Data. Google Inc. Available on: https://github.com/google/cluster-data. Accessed 7 July 2017.