IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 8, 2019, accepted May 24, 2019, date of publication May 30, 2019, date of current version June 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2920010

Intra-Balance Virtual Machine Placement for Effective Reduction in Energy Consumption and SLA Violation

AL-MOALMI AMMAR[1], JUAN LUO[1], ZHUO TANG[1], (Member, IEEE), AND OTHMAN WAJDY[2]
[1]School of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[2]School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei 230026, China

Corresponding author: Juan Luo (juanluo@hnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672220, in part by the Key Scientific and Technological Research and Development Plan of Hunan Province under Grant 2017GK2030, and in part by the National Defense Basic Research Plan under Grant JCKY2018110C145.

ABSTRACT Cloud computing has emerged as one of the most important technological revolutions globally. However, the rapid growth of cloud computing has imposed a massive financial burden and resulted in environmental side effects due to excessive energy consumption. The high power consumption is not only attributed to the size of data centers but also to the ineptitude of resource usage. Most of the extant research has focused on reducing power consumption by an aggressive VM consolidation, which leads to the violation of the service level agreement (SLA). Furthermore, the unbalanced resource consumption exacerbates the unavailable wasted resources that are referred to as unavailable resource fragmentation. In this paper, we propose the use of a balanced resources consumption algorithm called BRC-IBMMT in order to enhance the efficiency of resource consumption while achieving an acceptable balance between conflicting correlation objectives of power consumption as well as SLA violation. The extensive simulation results of different types of workload validate and lend credence to the significance of the proposed method in reducing power consumption and SLA violation of the cloud data center.

INDEX TERMS Balanced resource placement, VM consolidation, energy consumption, cloud computing.

I. INTRODUCTION

Cloud computing is a practical way of using the network to access powerful servers on cloud data centers to process, store, or manage data, as well as using applications online. This technology has revolutionized the world of information technology and streamlines the means of providing network services. Cloud providers offer different models of cloud service such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [1]. There are different objectives for both providers and consumers regarding these services.

IaaS model, which is the focus of this study, is a technique of providing physical resources such as compute resource, storage, and network components to the customers. By adopting the mechanism of pay-as-you-use in IaaS model, it enables convenient on-demand provisioning of cloud resources as the customers typically pay on a per-use basis [2]. Service providers and customers have conflicting objectives in cloud computing. While service provider invests in cloud computing to increase the profitability, which faces the huge budget of operational costs and negative environmental side effects, customers are looking forward to using cloud computing for excellent service at a lower price.

Power consumption has a large contribution to the operating cost of cloud centers by the cooling system and physical equipment. As reported from Microsoft [3], the power cost of the physical resources (e.g., CPU, memory, network, etc.) is nearly 15% of the total cost, which indicates the importance of reducing the power consumption of the cloud data center. Besides, most of the data centers use only less than 50% of its resource [4] which results in a massive amount of resource wastage and power consumption [5]. Therefore, consolidating VMs by virtualization technology on a low number of Physical Machines (PM) helps service providers to reduce the operating cost.

The associate editor coordinating the review of this manuscript and approving it for publication was Nafees Mansoor.

VM consolidation increases the physical resource utilization in order to minimize the number of active hosts, which significantly reduce the power consumption. At the same time, cloud service providers should ensure QoS, which is represented in service level agreement (SLA). As a result of consolidating multiple VMs on the same PM, power consumption is effectively reduced, but aggressive consolidation has negative effects on other metrics. Depending on resources consumption, VMs have different behavior in resource consumption, whereas the diversity of applications running on VMs result in resources competition. Hosting the same type of resource consumption of VMs (e.g., intense CPU, intense memory, intense I/O, or intense network) increases the resource contention and result in unbalanced resource consumption [6]. As an instance, because of scarcity of CPU, which is produced by consolidating multiple VMs of intense CPU on the same PM, the VM placement may be blocked of hosting a new VM resulting in resource wastage [7], [8].

Based on the aforementioned observation, contradictive objectives make the purpose of minimizing power consumption and ensuring SLA an open issue with a lot of room for improvement [9], which remains a challenge in previous works. Such conflicting objectives hard to efficiently satisfied as high resources utilization leads to low QoS. Therefore, Cloud providers need efficient placement strategy, which can make a balanced trade-off between low power consumption and high QoS. Most of the research solve server consolidation by classic bin-packing algorithms. However, because of the continuous and dynamic change of workload in the cloud; which cause VM migrations, classic bin-packing algorithms are not particularly applicable to server consolidation.

The assumption of that bins are empty is contrary to server consolidation where the servers have running VMs. While the capacity of bins continues constant, the resource demands from VMs dynamically change; which increase the VM migrations and resource fragments. Moreover, unbalanced resource consumption has a negative effect on power consumption and resource utilization; where the unbalanced PMs are more sensitive for workload change, which increases the live migration, resource wastage, performance degradation, and power consumption. Consequently, it is necessary, when designing server consolidation and VM selection algorithms, to consolidate VMs in a minimum number of active PMs which satisfy the VMs' requirements and minimize the resource fragmentation.

In this paper, we propose a balanced resource consumption algorithm called (BRC-IBMMT). The algorithm effectively balances the consumption of the resource and finds an acceptable balance between contrary correlation objectives of power consumption and SLA violation. Two algorithms for VM deployment and selection called Balanced Resource Consumption (BRC) and Imbalance VM with Minimum Migration Time (IBMMT), are proposed. Both algorithms consider CPU, RAM, bandwidth (BW). VM deployment algorithm (BRC) leverages the efficiency of resource utilization and minimize resource fragmentation while VM selection policy (IBMMT) enhances the balance of resource consumption and reduces the migration cost. The main contribution of this paper can be summarized as follows.

1) Balanced resource consumption algorithm called (BRC-IBMMT) is proposed
2) An effective algorithm called (BRC) is proposed to balance resource consumption and minimize the resource wastage.
3) A new VM selection policy (IBMMT) that leverages the balance of the resource consumption and minimizes the migration time is presented.
4) Intensive experiment of different types of workload with different settings is conducted to test the efficiency of proposed algorithms.

The following sections of this paper are organized as follows. Section 2 outlines related works, while section 3 illustrates system model and metrics definitions. The distributed model and its proposed algorithms are presented in section 4. Experiment setup and simulation results are revealed in section 5. Finally, the conclusion is addressed in section 6.

II. RELATED WORKS

Most of the related studies considered the VM allocation as Bin packing problem and applied greedy heuristic models. Well-known research used simple greedy heuristics, while others used the optimization of greedy heuristics [6], [10]–[13]. However, most of these approaches have depended on one-dimensional resource and did not consider multi-objective goals as maximization of resource utilization, minimization of live migration cost and energy consumption.

In [14], VMs were initially submitted according to the least power consumption placement algorithm, and then four algorithms for overload detection with three algorithms for selecting VM migration were proposed. The combinations of these algorithms produced good results for desired objectives in cloud computing. However, these algorithms depended on one-dimensional resource and often overused or wasted other resources. Moreover, these algorithms focused on power consumption and SLA, but failed to consider intra-balance of the resource utilization to decrease the resource wastage. A recent study was compiled by Song et al. [15] who modeled resource allocation as an online bin packing problem. The study devised an algorithm called Various Size of Bin Packing Algorithm (VISBP) which is capable of dealing with the change in VM size during the runtime. The results showed that VISBP, among other algorithms mentioned in the paper, has a good balance and detection technique. However, SLA or impact of migration cost had not been discussed.

Ammar *et al.* [16] proposed an anti-overload model by using a preemptive detection of overload and then migrating depending on Single Exponential Smooth (SES) prediction technique. The work made a significant reduction in SLA violation and number of VM migration. However, algorithms considered the only CPU and caused wastage of other resources. Besides, this work did not improve the VM placement to minimize power consumption. Han *et al.* [17] proposed an algorithm for VM placement called Remaining Utilization-Aware (RUA). The algorithm conducted live migration in case of overload situation and replaced the VM depending on the remaining capacity of PM. However, the authors only consider the CPU as the main resource type to place the VM. At the same time, the algorithm left significant capacity on every PMs.

Zhao et al. [18] proposed two algorithms of VM placement, Least-Reliable-First (LRF) and Decreased-Density-Greedy (DDG) to increase cloud provider's revenue by reducing the cost of SLA violation. Algorithms have produced results on the revenue of cloud providers under different conditions. However, the ideal conditions are by no means guaranteed in reality. Moreover, the study considered the VM instances in the form of one dimension which produces the resource wastage. Hao *et al.* [19] proposed a generalized resource placement methodology to allocate the VM demands with additional constraints on data center location, service delay guarantee. This study focusing on how to choose the VM allocation with minimum delay and data traffic among the cooperative VMs. However, the proposed research does not consider the efficient use of resource utilization or power consumption.

In [20] the authors adopt grey wolf optimizer to solve VM optimization into an optimum number of active hosts. As a result of submitting the VMs to the minimum number of PMs, the power consumption had been reduced to the minimum. However, they only studied initial VM placement, in which the SLA violation was not taken into the consideration of VM placement.

Zhao *et al.* [21] proposed power-aware and performance-guaranteed (PPVMP) method. In this methodology, the authors uses a non-linear energy model in VM placement as a bi-objective problem and solved by the ant colony optimization. The results have shown a reasonable reduction in the power consumption of the data center. However, they only studied the static VM placement at the homogeneous PMs data center. Moreover, they did not discuss the VM migration, different workload, resource utilization or SLA violation.

Placement methods are still facing some challenges in term of power consumption and SLA violation. In this paper, we attempt to enhance the efficiency of resource utilization to reduce power consumption and present good QoS.

III. SYSTEM MODEL

A. VIRTUAL MACHINE PLACEMENT (VMP)

VM placement is the process of choosing the most suitable host to accommodate the VM according to resource requirements. VMP tackles the persistent requests of hosting VMs, and extra user demands such as cancellation and resizing the VM at runtime. VM requests include a diversity of multiple resource requirements such as CPU, memory (RAM), and network bandwidth (BW). Similarly, the physical machine has a variety of multiple resource capacities. The concern for the placement process is how to accommodate the arrival VMs on sufficient PMs depending on the consideration of multiple resources.

VMP from the perspective of the provider should place as much VMs requests as possible on the same PM. As a consequence, the number of operating PMs is minimized, and power consumption is decreased. These subsequent results would lead to increased profitability and decreased power consumption. In contrast, saturated PMs drove the system to SLA violations and performance degradation that influences QoS. Consequently, in this paper, the VMP aims to conduct compound goals to satisfy both cloud provider and customer by balancing between the conflicting objectives. Most of the researchers tackle VMP as bin packing problem. In the following, we illustrate the description of the bin packing problem for VMP.

B. MODELING VM PLACEMENT OF MULTI-DIMENSIONAL RESOURCE TYPES

The problem of allocating some requests of VMs to the least number of sufficient PMs in a balanced manner can be formed as a multi-dimensional bin packing problem which is NP-hard complete problem [22]. Meanwhile, the preset threshold must not be violated to ensure the QoS. In this paper, we described the capacity of $P_j \in \{P_1, P_2, ..., P_m\}$ as a cube where CPU, RAM, and BW form the cube's sides. In the same way, the VM $V_i \in \{V_1, V_2, \ldots., V_n\}$ requirement was described as a small cube where CPU, RAM, and BW form the cube's sides. The resource types are represented as RT. The PM Resource Capacity (RC) denotes the total capacity of PM where $RC_j^k = \{RC_j^{cpu}, RC_j^{RAM}, RC_j^{BW}\}$, and the Utilized Capacity (UC) of PM denotes the occupied capacity of PM P_j where $UC_j^k = \{UC_j^{cpu}, UC_j^{RAM}, UC_j^{BW}\}$. Likewise, each VM V_i Resource Requirement (RR) denotes the demanded resources where $RR_i^k = \{RR_i^{cpu}, RR_i^{RAM}, RR_i^{BW}\}$, and Resource Utilized (RU) of VM $RU_i^k = \{RU_i^{cpu}, RU_i^{RAM}, RU_i^{BW}\}$. The ratio of the utilized capacity of P_j for each type is calculated as the summation of RUs for VM V_i. Symbols used in this paper are defined in Table 1.

$$UC_j^k = V_{ij} \times P_j \times \frac{\sum_{i=1}^{n} RU_i^k}{RC_j^k}, \quad \forall\, k \in RT,\ V_{ij} = 1 \quad (1)$$

where $V_{i,j}$ is submitted to one and only one PM $V_{ij} = \begin{Bmatrix} 1 \ if \ V_i \ is \ submitted \ to \ P_j \\ 0 \qquad Otherwise \end{Bmatrix}$

C. LIVE MIGRATION COST

Live migration is an essential technique for placement optimization. Live migration enables transfer of VMs among hypervisors on PMs without suspension or prolonged downtime. Live migration is used to resize VM, avoid overload, QoS requirement, or maintenance at the run time. Although this technique enhances resource management to scale up or down depending on the demand, VM migration

TABLE 1. Description of symbols used in paper.

Symbol	Description
NRT	Number of resource types
RC_j^k	Resource Capacity of k type of resource for server j
UC_j^k	Utilized Capacity of k type of resource for server j
RR_i^k	Resource Requirement of k type of resource for VM i
RU_i^k	Resource Utilization of k type of resource for VM i
MT_{vi}	Migration Time of VM i
$BW_{spMigration}$	Specified Network bandwidth for migration
PD_{vi}	Performance Degradation by VM i
SLAVTAH	SLA violation time for Active servers
VT_j	Violation Time of server j
$activeT_j$	Active Time of server j
VMPD	Performance degradation of VM migration
EPD_i	Estimated Performance degredation of migrated VM i
RW_j	Resource Wastage of server j
$RemC_j^k$	Remaining Capacity of k type of resource for server j
PUC_t^{cpu}	Prediction of Utilized Capacity of cpu at time t
α	Smoothing factor where $0 < \alpha < 1$
T_j	CPU Threshold of server j
BF_j	Balance Factor of server j
ERU_j^k	Estimated Resource Utilization of k type for server j
$max(U)$	percentage value of the maximum utilized capacity
UL	Upper probability limit
LL	Lower probability limit
IBF_j	Imbalance Factor for server j
$IBmax(U)$	Percentage value of the maximum utilized capacity
MMT_{mu}	Ratio of utilized memory to CPU
E_{CPU}	Energy Consumption of CPU
E_{RAM}	Energy Consumption of RAM
E_{RW}	Power needed to conduct read or write command
E_{Back}	Power Consumption of Active or Standby memory

takes time, creates additional overhead on the involved PMs, and produces an adverse impact on QoS and SLA.

There are two primary performance metrics to describe and measure the performance of live migration. The first metric is migration time which refers to the time of transferring all pages of VM memory (the original memory and the dirty/modified memory), and the CPU status. Due to the large size of VM memory which is usually much larger than the status of CPU, the migration time of CPU status is trivial.

The second metric is performance degradation during the migration process. Therefore, it is essential to keep the number of VM live migration minimized.

The popular way of live migration is known as pre-copy, Voorsluys *et al.* [23] explored the impact of live migration and formulated a way to model it. Consequently, we calculated the migration time and the performance degradation during the migration operation for VM as shown in this equation:

$$MT_{v_i} = \frac{RU_i^{RAM}}{BW_{spMigration}}, \tag{2}$$

$$PD_{v_i} = 0.1 \times \int_{t0}^{t0+MT_{v_i}} RU_i^{cpu}(t)\, dt, \tag{3}$$

D. SERVICE LEVEL AGREEMENT (SLA)

SLA is the quality of expected service defined in the contract between the cloud provider and its customers. This contract may contain the performance indicators, availability, response time, financial penalties, or target values [24]. SLA violation occurs when one or more of service level objectives (SLOs) are dissatisfied. There are many reasons for SLA violation, such as aggressive consolidation, improper sizing of VMs or poor elasticity solutions [25]. Consequently, the public, private, and hybrid cloud providers should supply VMs with modern elasticity. Beloglazov and Buyya [14] formulated two metrics to measure the SLA violations in an IaaS environment. The first measurement is the average of violation time during the active PM time (SLAVTAH). The second metric is the performance degradation migration due to VM's migration (VMPD):

$$SLAVTAH = \frac{1}{m}\sum_{j=1}^{m}\frac{VT_j}{activeT_j} \tag{4}$$

$$VMPD = \frac{1}{n}\sum_{i=1}^{n}\frac{EPD_i}{RU_i^{cpu}} \tag{5}$$

Both AVT and PDM metrics are independent. Consequently, SLA violation is a multiplying of AVT and PDM.

$$SLAV = SLAVTAH \times VMPD. \tag{6}$$

E. RESOURCE WASTAGE MODEL

Data centers' servers are rarely fully utilized, which increases the number of active PMs. VM placement concerning only one resource type leaves various amounts of residual capacity on other resources. This remaining resource knows as resource fragmentation in which unutilized resource wasted in the unavailable PMs. Consequently, the placement strategy should utilize the available resource efficiently for all resource types. The high resource wastage indicates the inefficiency of resource utilization which restricts the ability to exploit residual capacities for coming requests. Therefore, the efficient allocation should optimize resource utilization or leave the residual resources balanced for more opportunities for hosting new VMs on the same PM. Depending on the description mentioned above, we formulated a mathematical formula to quantify the resource wastage of PM which is the extent of the model of a multi-dimensional resource [26], [27]. The total amount of resource usage by PM P_j is estimated as the total resource used by all VMs running on it. We calculated the resource wastage as a percentage value. The high percentage of Resource Wastage (RW) indicates inefficient use of the resource or imbalance placement which saturates one resource dimension and prevents from placing more VMs on PM. The average resource wastage of major resource is calculated as described in equation (7).

$$RW_j = \frac{1}{NRT} \times \sum_{k \in RT} RemC_j^k,$$

where

$$RemC_j^k = (1 - UC_j^k) \tag{7}$$

The minimum percentage value of resource wastage indicates the efficiency of resource utilization.

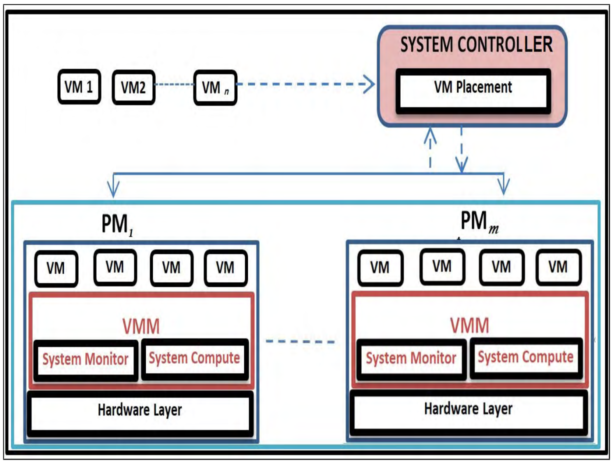


FIGURE 1. Distributed system model.

IV. THE PROPOSED DISTRIBUTED MODEL

In this section, we give the details of our model that aims to maximize resource utilization and balanced resource consumption. The model not only involved placing the VMs on sufficient PMs but also set them in a balanced way. We have employed the distributed model for an elastic and extensive framework which fits the cloud computing paradigm. The three main components of the model are:

1) **System controller** which is responsible for making global management decision such as VM placement. It is worth mentioning that the system controller is installed on PM which does not host any VM. The system controller receives requests of customer allocation demands and then places them efficiently on sufficient PM in a balanced way for multiple resources. Furthermore, the controller is also responsible for remapping VMs to other PMs during the migration while retaining the system in a balanced manner.
2) **System compute** is installed on all PMs which are responsible for hosting VMs. The system compute is deployed as a part of VMM to detects the overload or the under load utilization. Besides, the system compute selects the VMs for migration in case of PM overload.
3) **System monitor** is distributed on all PMs, which collects the resource utilization data, then stores it locally to be used by algorithms. Figure 1 depicts the distributed model for the cloud data center.

A. PROPOSED STRUCTURE AND ALGORITHMS

We concentrate on IaaS consisting of a large amount of PMs. These PMs have different capacities of CPU, RAM, and BW. Besides, the storage of VMs and its data is within shared storage as Network Attached Storage (NAS) to enable live migration. These VMs should be distributed by the efficient method to satisfy both the cloud provider and customers. VM placement and consolidation are two steps to place VMs in the right place that satisfy the management policies and to provide a better service at a lower cost. We try to settle VM in the right place to keep the system stable and minimize the VM migration. The method of hosting VM in a suitable place can be divided into five substeps:

1) Placing VM in an appropriate place according to requirements.
2) Consolidating VMs on a minimum number of PMs depending on the current VMs utilization.
3) Detecting the overload situation so that some VMs are migrated to mitigate the overload under the threshold.
4) Detecting the underload situation, so all VMs are moved to other active PMs and shut down these PMs.
5) Relocating the migratory VMs from the overload and underload PMs into the right place.

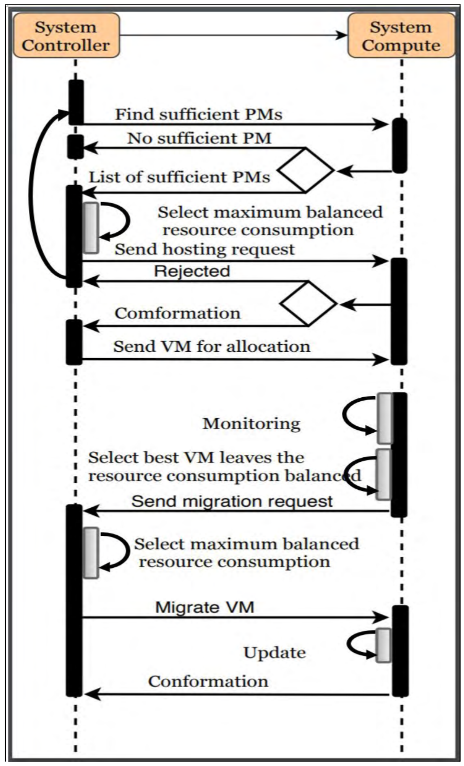


FIGURE 2. Distributed interaction components.

In this section, we introduce two algorithms to address VM deployment and selection. The placement algorithm has to place the arrival VM requests in a balanced way depending on multi-dimensional resource types, rather than depending on one resource dimension. Furthermore, another algorithm is invoked to mitigate PM load in the case of threshold violation resource. Thresholds policy is vital to restrain the violation of SLA and guarantee a good QoS to customers. Eventually, when an overload has been detected, it is eschewed by migrating VMs to calm the load under the violated threshold. Figure 2 illustrates the interaction components of distributed algorithms to achieve intra-balance placement and VMs selection.

B. HOST OVERLOAD DETECTION

The power consumption can be reduced efficiently through consolidation of VMs on a minimum number of active PMs and shutting down other PMs. However, aggressive consolidation or placement leads to an increase of SLA violations resulting in poor service [28]. The flexibility of the virtualization technique which allowed performing a live migration of VMs between the supervisors solved the problem of the overloaded host. The host overload detection should be accurate enough to minimize the number of live migration and estimate the overload before it occurs.

In this paper, the overload has been detected depending on the adaptive threshold policy. The host is considered as an overloaded PM if the estimated resource utilization exceeds the adaptive threshold. In the following, we explained the method of calculating the resource utilization of multi-dimensional resources, and the predicted approach which uses the forecast technique called single exponential smooth (SES) [29]. Besides, we explained the method of calculating the adaptive threshold.

1) RESOURCE UTILIZATION PREDICTION

We employed a time series data model called SES to make a prediction. This technique has been applied in our previous work which gave good results [16]. SES is a kind of weighted moving average sequence data that we used to forecast the resource utilization. The time sequence of the observation started from zero and ended at time t, where t is the time for the final observation value. The resource utilization data collected into the two-dimensional array. The X rows present the length of data collected in t time, and Y column presents the CPU utilization. SES prediction technique used the list of observed resource utilization of CPU to calculate the predictive value. The expression for SES is given by the formula 8.

$$PUC_{t+1}^{cpu} = PUC_{t}^{cpu} + \alpha(UC_{t}^{cpu} - PUC_{t}^{cpu}) \tag{8}$$

The formula can be expressed as in formula below:

$$PUC_{t+1}^{cpu} = \alpha \times UC_{t}^{cpu} + (1-\alpha) \times PUC_{t}^{cpu} \tag{9}$$

SES uses all the previous historical data which makes it more stable and uniform. The smooth value α presents the sensitivity of the model to the frequent data changes over the observed time. If the collected data of resource utilization has a unified response rate, the smooth value α is a constant value close to zero, but if observation values have fluctuated rates, the smooth value α is close to one. In our experiment, we set the α to 0.7 because of great fluctuating of workload data.

The initial of predictive value defined as PUC_{1}^{cpu} which initialized into real resource utilization value if the length of the collected data series is more than 15, which is the experimental value. If the length of the collected data series is less than 15, the PUC_{1}^{cpu} is calculated as the average value of the previous observations.

$$PUC_{1}^{cpu} = \begin{Bmatrix} \sum_{i=0}^{l} \frac{UC_{i}^{cpu}}{l}, & l < 15 \\ UC_{1}^{cpu}, & l \geq 15 \end{Bmatrix} \tag{10}$$

2) AN ADAPTIVE UTILIZATION THRESHOLD

The static threshold is unsuitable for a dynamic workload environment such as cloud computing. The cloud system threshold should be an auto-adjusted value capable of automatically changing its value depending on the hosted application's workload. We can auto-adjust the upper utilization threshold T_j for each PM_j by preserving the amount of spare capacity between lower and upper limits (LL and UL) as shown in the equation 11.

$$T_j = 1 - |(UL - LL) \times UC_{j}^{cpu}| \tag{11}$$

where UL and LL are the upper probability limit and low probability limit respectively. The PM is considered as an overloaded PM if the predictive value is more than the upper threshold. Therefore, some VMs would be preemptively migrated to avoid the overload situation. In our experiment, we set the UL and LL to 85 and 90, respectively.

C. BALANCE FACTOR FOR VM PLACEMENT

The cloud providers' goal is to maximize resource utilization which reduces the total energy consumption of the cloud center. Consequently, the better way to maximize resource utilization is to submit the VMs to the PMs in a balanced way depending on multi-dimensional resources. Figure 3 clarifies two examples of assigning the different size of VMs to the same PM. Figure 3a illustrates imbalanced allocation, and Figure 3b shows the balanced allocation. The capacity of PM for CPU, RAM, and BW represented as a cube. The required resources of CPU, RAM, and BW of three VMs are shown as small cubes. The residual capacity on PM represents the big cube that refers to the remaining capacity after accommodating three VMs.

Obviously in Figure 3a, the VMs hosted on this PM are incompatible, so it leaves the PM with an unuseful remaining capacity due to the saturation of CPU. The deficiency of CPU capacity restrains the opportunity of placing a new VM. In contrast, the residual capacity in the figure 3b is balanced due to the good strategy of choosing the compatible VMs which give a good opportunity to host new VMs. The algorithm estimated resource utilization of adding VM to PM before hosting depending on available information for PMs capacities and requirement of VMs.

The Balance Factor (BF) and the estimated resources utilization are calculated as equation 12:

$$BF_j = \sqrt{\frac{\sum_{k \in RT} (ERU_{j}^{k} - max(U))^2}{NRT}},$$

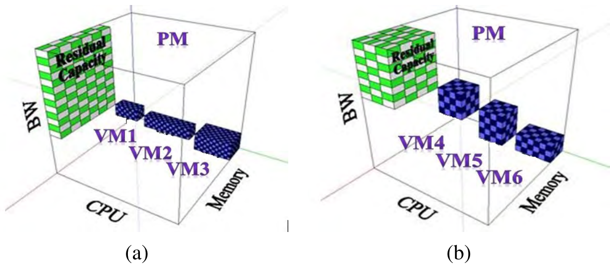


FIGURE 3. Balance and Imbalance resource allocation of three VMs into a single PM. (a) Imbalance allocation. (b) Balance allocation.

where

$$ERU_j^k = \left(RU_j^k + RR_i^k\right), \quad \forall\, k \in RT$$
$$max\,(U) = Max(ERU_j^k) \tag{12}$$

BF_j is a percentage value between [0,1], where value close to zero indicates efficient balance, and value close to one indicates that one or two types of resource might be saturated. Consequently, we used the Balance Factor (BF) in our placement algorithm to submit the VM to PM, where the minimal BF ensures that the residual capacity is balanced for a better chance of placing new VMs on the same PM.

D. IMBALANCE FACTOR FOR VM SELECTION (IBF)

The imbalance factor of VM selection is the method of choosing VM to be migrated in the case of threshold violation. The imbalance factor is a percentage value between [0, 1] which determines the VM for migration, equation 13. The best choice to select VM for migration among the list of VMs is the lower value of IBF in which the PM machine would be more balanced after the VM migration. Applying this method leads the whole system to be balanced shortly and maintains a balanced residual capacity to accommodate more VMs.

$$IBF_j = \sqrt{\frac{\sum_{k\in RT}\left(UC_j^k - IBmax\,(U)\right)^2}{NRT}}$$

where

$$UC_j^k = \left(UC_j^k - RU_i^k\right), \quad \forall\, k \in RT$$
$$IBmax\,(U) = Max(UC_j^k) \tag{13}$$

E. BALANCED RESOURCE CONSUMPTION (BRC)

The algorithm is deployed on a system controller which is accountable for accommodating the arrival of VM's requests in a tidy balanced way to balance the resource consumption. Due to the different instances of which include an intensive CPU and intensive RAM, accommodation of one type of instance on the same PM would waste the capacity of other resources, increase the contention of resource usage, and augment the operating expenses. However, BRC is an accurate method of accommodating the VMs on PMs to balance resource consumption.

A multi-dimensional vector introduces the capacity of PM resources and requirement of VMs; those dimensions present the amount of resource for the major types of resources such as CPU, RAM, and BW. The system controller receives the hosting requests with the resource requirements, and the requests are sorted by their intra-balance of resource requirements, then the scheduler starts to scan the available resources on all operating servers of the cloud data center to find the sufficient PM to host the customer's VMs. Among the sufficient PMs, the algorithm starts to estimate the load of transferring VM to PM and verifies if the PM would become overloaded. If so, the algorithm excludes the PM from the potential hosting list.

Furthermore, the algorithm starts to classify the optimal choice among the active hosts by the BF and resource consumption, which introduces the best balance of the minimum residual capacity. Resource consumption is the average resource utilization for all types of resource. The algorithm looks for the minimum ratio between BF and resource consumption. The minimum ratio of BF and resource utilization indicates that the allocation policy chooses the maximum balanced resource consumption. This method increases the opportunity of hosting more VMs into the same PM and exploits the resources to their maximum potential. If there is no active host under these conditions, the algorithm submits the VM into a new PM. In that case, the number of active hosts and power consumption would be decreased. Moreover, the balanced resource consumption will decrease the resource fragmentation and minimize the need for VM migration. The pseudo-code of the algorithm is exhibited in the algorithm (1). The complexity of the algorithm is $n \times m$, where n is the number of VMs and m is the number of PMs in the data center.

F. IMBALANCE VM WITH MINIMUM MIGRATION TIME SELECTION ALGORITHM (IBMMT)

The consolidation process optimizes resource utilization and enhances the power consumption depending on the current use of VMs. However, the aggressive consolidation increases the SLA violations which cause a poor QoS. Consequently, the consolidation process should be allowed to the determined limit to keep spare space which can be used in the case of VMs extension or unreasonable load change. As a result, the overload situation is expected, and some VMs should be migrated to reduce SLA violation. The overload detection is defined depending on the preset threshold, the predicted value of the overloaded PM is compared with the threshold to discover the overload before its occurrence. The prediction technique minimizes the time of SLA violation in which the migration starts before the actual overload occurs. Moreover, using a dynamic threshold is suitable for such an environment as a cloud data center to avoid unnecessary VM migrations. Strictly, the system starts to observe resources to restrain the threshold violation. If PM faces the overload situation, the VM selection is an urgent procedure to select VM for migration. The selection of VM migration has a direct impact on the state of the source and destination. Where the size and number of VM migrations affect the performance of

Algorithm 1 BRC

```
input  : VMList , PMList
output : allocated_PM
VMList.SortDescendingIntra-balance( );
for each VM in VMList do
    minRBC ← MAX;
    allocatedPM ← Ø;
    for each PM in PMList do
        if host suitable for VM then
            if !isHostOverUtilizedERU_j^k(PM, VM) then
                if RU_j^cpu > 0 then
                    BF ← BF_j /*eq(12);
                    RW ← *eq(13);
                    RatioBFRC ← BF/(1 − RW_j);
                    if RatioBFRC < minRBC then
                        minRBC ← RatioBFRC;
                        allocated_PM ← PM;
                    end
                end
            end
        end
    end
    if allocated_PM = Ø then
        allocated_PM ← newPM;
    end
    if allocated_PM ≠ Ø then
        PlaceVM toallocated_PM;
    end
end
Return: allocated_PM
```

the cloud system. The VM selection can be conducted for multiple objectives, but the main objective is to relieve the overload and minimize the migration time. However, the VM selection can also target additional objectives such as balancing resource consumption. IBMMT algorithm aims to remain the resource consumption balanced, and minimizes the migration cost. The negative impact of the live migration cost on cloud system imposes some rules on selecting migratory VMs. Therefore, choosing the most suitable VM is the critical process for the model to handle before conducting the migration to mitigate SLA violation and performance degradation. In our algorithm, VM selection is achieved for multiple goals such as balancing resource consumption and minimizing the downtime. Therefore, the migration must be done fast to avoid performance degradation and maintain the balance of resource consumption. In the case of threshold violation, the algorithm looks for all VMs which comply with the condition of reducing the load under the threshold policy (if the difference of current PM utilization and the violated threshold value is less than the VM utilization). Then, the algorithm starts to classify them depending on IBF and the proportion of memory and CPU consumption of VM. The algorithm strives to find the VM which leaves the resource consumption balanced and has the least proportion of memory and CPU consumption at the same time. Migrating the proper VM reinforces the balance of resource consumption and minimizes the migration time; which improves the QoS. In our algorithm, both factors have the same importance. Therefore, we have used a weighted sum of the two objectives and give them the same weight according to equation (15):

$$MMTmu = \frac{RU_i^{RAM}}{RU_i^{CPU}} \tag{14}$$

$$\begin{aligned} muIBF = &\frac{IBF_i - IBF^{max}}{(IBF^{max} - IBF^{min}) \times 2} \\ &+ \frac{MMTmu_i^{RAM} - MMTmu^{max}}{(MMTmu^{max} - MMTmu^{min}) \times 2} \end{aligned} \tag{15}$$

The pseudo-code for the algorithm is exhibited in algorithm (2). The complexity of the algorithm is $n \times l \times m$, where n is the number of VMs and l is the number of accepted VMs that can take the load under the threshold, and m is the number of the overloaded PMs in the data center.

V. EVALUATION METHODOLOGY

A. EXPERIMENT SETUP

In this section, we evaluate the impact of the intra-balance of resource consumption on power consumption and SLA violations. Due to the difficulty of reproducing results within the real infrastructure, we have used a simulated method to assess the proposed solution in comparison to other algorithms. One of the best simulation platforms to simulate IaaS is cloudsim [30]. Cloudsim supports many features of cloud management options such as provisioning on-demand resources, power awareness solutions, and dynamic workloads. We used the 3.1 version toolkit of the cloudsim. We compared our algorithm with the same version of VM placement problem algorithms. The beloglazov [14] algorithm is the default placement algorithm in Cloudsim which has conducted with a different policy of VM selection and overload detection. The placement algorithm chooses the least power consumption to host the VM. Moreover, the authors find the best performance come with local regression for overload detection which was adapted for all following comparative algorithms. Guazzone *et al.* [11] algorithm is a type of nonlinear programming algorithm which sorted PMs to the best-fit -decreasing heuristic depending on multiple factors. Active PMs are sorted decreasingly according to the free CPU capacity. If it is a tie, the minimum power consumption is chosen. Chowdhury *et al.* [12] algorithm has the opposite behavior of Beloglazov algorithm where the highest difference of power consumption after hosting VM is preferred. Shi *et al.* [13] proposed two algorithms (SHI-PU, SHI-AC). SHI-PU algorithm sorted the PMs decreasingly according to the CPU utilization and choose the highest utilized PM. SHI-AC algorithm sorted the PMs decreasingly according to absolute capacity and choose the biggest PMs. RUA [17] submits the VMs to maximum CPU utilization with keeping spare space to guarantee the SLA. For the sake of

Algorithm 2 IBMMT

input : PMlist, VMList, T_j
output : VMmigratedList
for *each PM in PMList* **do**
  $Best\_muIBF \leftarrow max$;
  $bestchoiceVM \leftarrow \emptyset$;
  while $UC_j^{cpu} > T_j$ ***or*** $PUC_{t+1}^{cpu} > T_j$ **do**
    VMList.SortDescendingutilization( );
    for *each VM in VMList* **do**
      if $UC_j^{cpu} - T_j < RU_i^{cpu}$ **then**
        $IBF_i \leftarrow IBF_j / * eq(13)$;
        $MMTmu_i \leftarrow eq(14)$;
        $AcceptedvmList.add(VM)$
      end
      else
        $bestchoiceVM \leftarrow VM_i$;
        Break;
      end
    end
    if $bestchoiceVM = \emptyset$ **then**
      for *each VM in AcceptedvmList* **do**
        $IBF^{max} \leftarrow Max(IBF_i)$;
        $IBF^{min} \leftarrow Min(IBF_i)$;
        $MMTmu^{max} \leftarrow Max(MMTmu_i)$;
        $MMTmu^{min} \leftarrow Min(MMTmu_i)$;
        $muIBF \leftarrow muIBF * eq(13)$;
        if $muIBF_i < Best\_muIBF$ **then**
          $Best\_muIBF \leftarrow muIBF_i$;
          $bestchoiceVM \leftarrow VM_i$;
        end
      end
    end
    $UC_j^{cpu} = UC_j^{cpu} - RU_i^{cpu}$;
    $VMmigratedList \leftarrow bestchoiceVM$;
    $VMList.remove(bestchoiceVM)\}$;
  end
  if $RU_j^{cpu} < LT_j^{cpu}$ **then**
    $VMmigratedList \leftarrow PM.vmList$;
    $TurnoffPM$;
  end
end
Return: VMmigratedList

simplicity, all these algorithms are experimented in a single data center considering two types of resource, CPU, and RAM. However, our model can easily be extended to multiple resource types.

1) WORKLOAD

To evaluate algorithms, we have used a real system workload trace. The workload trace helps to reproduce realistic data rather than synthetically-generated data. Two types of real workload were used. The first type of workload is Google Cluster data which contains the resource and workload data [31]. The data of 1200 PMs have been collected in 29 days and publicly published. For each PM, the workload of VMs is included of jobs which contain multiple tasks. Also, PM data is included in these data. The second type of workload is Bitbrains which contains the workload of 1750 VMs taken every five minutes [32]. These data of workload includes the resource consumption of CPU and RAM, as well as network and disk.

TABLE 2. Power consumption at different utilization.

CPU Utilization (%)	Power consumption (W) HP Proliant G4	HP Proliant G5
0	86	93.7
10	89.4	97
20	92.6	101
30	96	105
40	99.5	110
50	102	116
60	106	121
70	108	125
80	112	129
90	114	133
100	117	135

2) ENERGY CONSUMPTION MODEL

We adopt the power model proposed in [33]. The power consumption is calculated as a summation of the energy consumption of both CPU and RAM. The different behavior of power consumption results in a different proportion of power consumption. Two types of servers have been used to accurately describe the power model which drive from SPECpower benchmark (http://www.spec.org/powerssj2008/ - Table (3)). The power consumption of RAM is calculated as a summation of background power and operational power. Background power of RAM is measured as a function of the CPU utilization. Operation power of RAM is determined as a product of RAM bandwidth and the energy consuming for operating read/write on RAM.

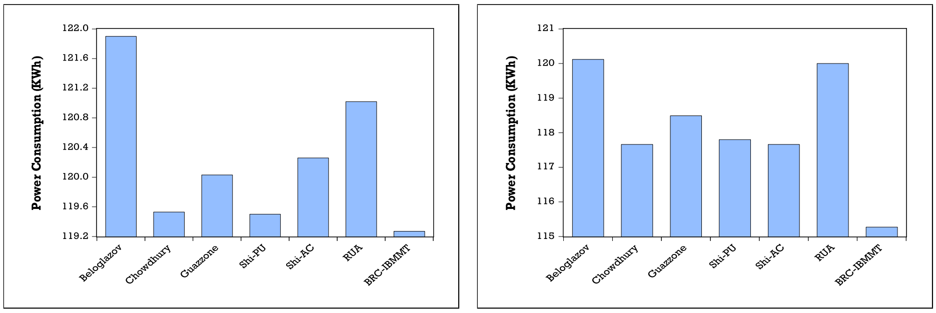
$$E = E_{CPU} + E_{RAM} \tag{16}$$

$$E_{RAM} = E_{RW} + E_{Back} \tag{17}$$

3) EVALUATION METRICS

Several metrics are used to compare the efficiency of the algorithms. These metrics should evaluate the objectives of the proposed methodology accurately. Consequently, the following metrics are appropriate to assess the efficiency of the algorithms compared to other related works:

1) **Power consumption** measures the effect of the proposed model to the reduction of energy consumption in the cloud system, which calculated as shown in section.
2) **SLAVTAH (SLA violation time per active PM)** shows the effect of the overload in the availability of service. It specifies the proportion of unavailability service time to the total time of service.



(a) (b)

FIGURE 4. Power consumption of homogeneous environment. (a) Google. (b) Bitbrains.

TABLE 3. Homogeneous configuration of PMs.

Hosts	CPU type	Frequency(GHz)	Cores	RAM (GB)
HP Proliant G4	Intel Xeon 3040	1.86	2	4

2) **SLAVTAH (SLA violation time per active PM)** shows the effect of the overload in the availability of service. It specifies the proportion of unavailability service time to the total time of service.
3) **VMPD (VM Migration Performance Degradation)** presents the result of VM migrations in the performance of running VMs.
4) **ESV(Energy SLA Violation** according to the negative correlation between power consumption and SLA violation, this metric reflects the trade-off between power saving and SLA violation, while the low value indicates the satisfying reduction of both factors.
5) **Number of VM migration** shows the number of VM migrations, as the small number of VM migrati reflects good stability and availability of the cloud s tem. However, not only the number of VM migrati contribute to SLA violation, but also the size and re-allocating time of migrated VMs.

3) **VMPD (VM Migration Performance Degradation)** presents the result of VM migrations in the performance of running VMs.
4) **ESV(Energy SLA Violation** according to the negative correlation between power consumption and SLA violation, this metric reflects the trade-off between power saving and SLA violation, while the low value indicates the satisfying reduction of both factors.
5) **Number of VM migration** shows the number of VM migrations, as the small number of VM migrations reflects good stability and availability of the cloud system. However, not only the number of VM migrations contribute to SLA violation, but also the size and re-allocating time of migrated VMs.

B. SIMULATION RESULTS AND ANALYSIS

In the following, we give the results of all experiments which conduct to ensure the performance of the proposed algorithm compared to others. The algorithms have been tested in different environmental settings of the cloud data center (homogeneous, heterogeneous, and different size of VMs).

TABLE 4. Four types of VM.

VM Type	CPU (MIPS)	RAM (GB)
HighCPU medium instance	2500	0.85
Extra large instance	2000	1.7
Small instance	1000	1.7
Micro instance	500	0.61

1) HOMOGENEOUS EVALUATION

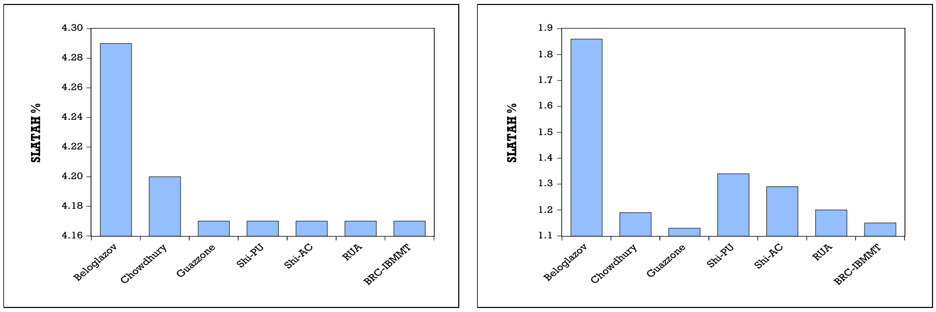
In this section, we assess the efficiency of our algorithms comparing with different algorithms. The compared algorithms are named according to the name of the first author. Table (3) shows the homogeneous specification of PMs. Table (4) shows the VMs types according to Amazon EC2. In this experiment, we have used the same type of PMs and different type of VMs.

Fig (4) shows the performance of all algorithms concerning power consumption for Google Cluster and Bitbrains workload respectively. Considering both cases of power consumption, our algorithm has achieved a remarkable reduction in power consumption. In contrast to other algorithms, our algorithm takes into the consideration of VM placement the balanced consumption of resources, which improves the resource utilization and contributes significantly to power reduction, where the same number of the VM has been placed into less number of active hosts. In the case of google cluster workload, as shown in Fig (4a), the power consumption is reduced from 121.09 KWh in Beloglazov's algorithm to 119.25 KWh in BRC-IBMMT algorithm, and from 120.12 KWh in Beloglazov's algorithm to 119 KWh in BRC-IBMMT algorithm, in case of BitBrain workload, as shown in Fig (4b).

According to Fig (5), which shows the performance of all algorithms in term of SLAVTAH for Google Cluster and Bitbrains workload. BRC-IBMMT succeeded to reduce the power consumption, which was expected to produce an increase in SLA. Even though the negative correlation between the reduction of power consumption and SLA,

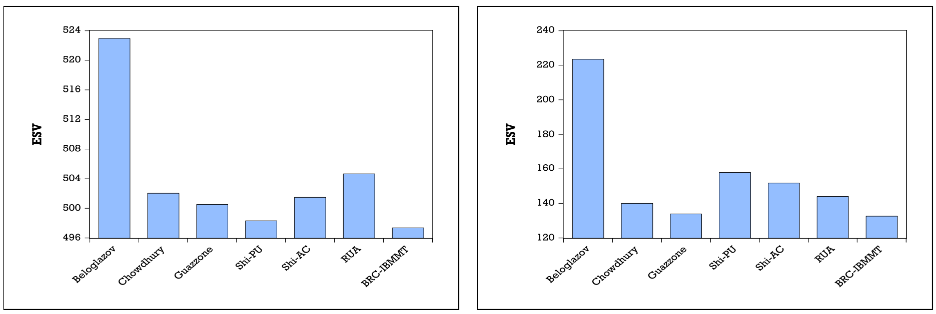
TABLE 5. Simulation result of homogeneous data center.

	Power (KWh)		SLAVTAH %		VMPD %		ESV		VM migrations	
	Google	Bitbrains	Google	Bitbrains	Google	Bitbrains	Google	Bitbrains	Google	Bitbrains
Beloglazov	121.9	120.12	4.29	1.86	0	0.02	522.951	223.4232	1092	3815
Chowdhury	119.53	117.66	4.2	1.19	0	0.02	502.026	140.0154	826	3402
Guazzone	120.03	118.49	4.17	1.13	0	0.02	500.5251	133.8937	792	2976
Shi-PU	119.5	117.8	4.17	1.34	0	0.02	498.315	157.852	784	2919
Shi-AC	120.26	117.66	4.17	1.29	0	0.02	501.4842	151.7814	836	3306
RUA	121.02	120	4.17	1.2	0	0.02	504.6534	144	805	3125
BRC-IBMMT	119.27	115.27	4.17	1.15	0	0.01	497.3559	132.5605	722	2856



(a) (b)

FIGURE 5. SLAVTAH of homogeneous environment. (a) Google. (b) Bitbrains.



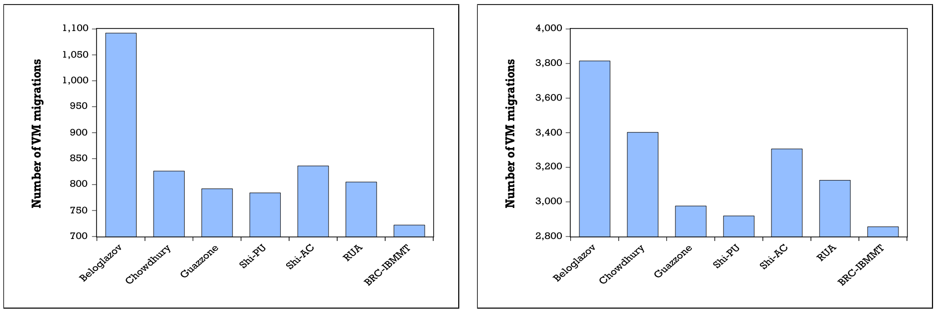
(a) (b)

FIGURE 6. ESV of homogeneous environment. (a) Google. (b) Bitbrains.

BRC-IBMMT algorithm reduces the SLA during the total time of service compared to other algorithms. As a result, balanced resource consumption in BRC-IBMMT causes a steady reduction in both power consumption and SLA time in active hosts. Moreover, BRC-IBMMT reduces the competition for resources, which achieves a significant reduction in performance degradation as shown in the table(5). As can be seen from the results, BRC-IBMMT registered the lowest value in both cases. In BitBrain as an instance, BRC-IBMMT lessens the SLATAH 38% compared to Beloglazov algorithm.

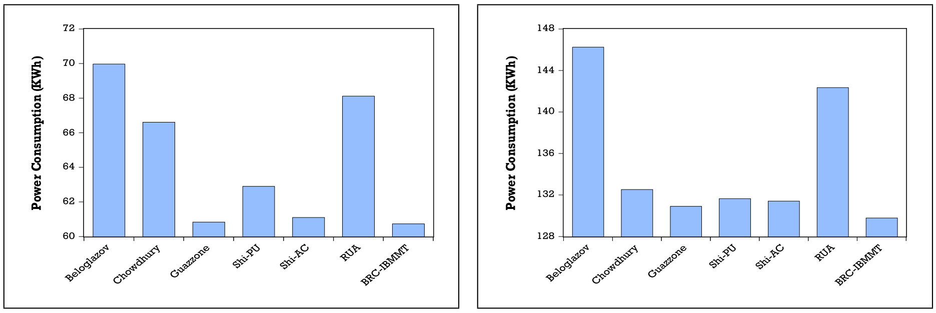
Although aggressive consolidation reduces power consumption, it increases SLA violations due to the negative correlation between them. However, the balanced consumption of resources contributes to a balanced reduction of both objectives. To prove the stability of algorithms in this context, ESV measured the system performance of all algorithms to show the trade-off between power consumption and SLA violation. Fig (6) shows the performance of all algorithms with respect to ESV for Google Cluster and Bitbrains workload respectively. Our algorithm gains the lowest value of ESV compared with other algorithms which reflect the efficiency of our algorithms when it targets both objectives. As an instance, BRC-IBMMT decreases the ESV 4 % compared to Beloglazov algorithm, as shown in Fig (6a) in case of Google Cluster workload, and 40 % compared to Beloglazov algorithm, as shown in (6b), in the case of BitBrain workload.

The number of VM migrations influences the availability of service for the customers, as a high number of VM migrations leads to a high downtime of service. However, not only the number of VM migrations is the cornerstone in the availability of service, but also the size of migratory VMs. Consequently, live migration should be considered according to the migration time and performance degradation.



(a) (b)

FIGURE 7. Number of VM migrations of homogeneous environment. (a) Google. (b) Bitbrains.



(a) (b)

FIGURE 8. Power consumption of heterogeneous environment. (a) Google. (b) Bitbrains.

Fig (7) shows the number of VM migrations for Google Cluster and Bitbrains workload, respectively. In contrast to other algorithms, which depend on one type of resource during the placement, it is clear from the figure that, balanced placement of BRC-IBMMT algorithm has contributed to the stability of the system and reduced the need to migrate VMs in case of overload PMs. In addition, the VM selection policy, which enriches balanced placement and takes into account the size and impact of migration in performance degradation, contributes to reducing the number of migrations. As shown in Fig (7a), BRC-IBMMT decreases the number of VM migrations 33% compared to Beloglazov algorithm in case of Google workload, and 25 % in the case of BitBrain workload, as shown in Fig (7b).

In summary, It is clear that our algorithm effectively reduces the value of all evaluation metrics in both cases. However, it was more noticeable in the case of BitBrain workload due to the diversity of loads in this case, which clarifies the effect of balanced resource consumption of BRC-IBMMT algorithm in all evaluation metrics.

2) HETEROGENEOUS EVALUATION

Most of the real data centers are a heterogeneous environment. Data centers contain a hundred thousands of different types of PM. Therefore, to evaluate the performance of the algorithm in the real simulated data center, we have to evaluate them in such an environment. In this section, we assess the efficiency of all algorithm in a heterogeneous environment. Table (6) shows the heterogeneous specification of PMs. In this experiment, we have used the heterogeneous types of PMs and VMs.

TABLE 6. Heterogeneous configuration of PMs.

Hosts	CPU type	Frequency(GHz)	Cores	RAM (GB)
HP Proliant G4	Intel Xeon 3040	1.86	2	4
HP Proliant G5	Intel Xeon 3075	2.66	2	4

It can be seen from Fig (8) that BRC-IBMMT shows an efficient reduction in power consumption for both types of workload. However, it is more noticeable for BitBrain workload case. A heterogeneous environment with a diversity of workload makes the case of placement more complicated, although, BRC-IBMMT shows an effective reduction in power consumption compared to other algorithms, which do not consider the balanced consumption of resource. Therefore, placing VM depending on a single resource type increases the fragmentation of residual resources. In this case, most of the residual resource are not useful to allocate new demands. As an instance, in the case of Google Cluster workload, BRC-IBMMT gets the best result, which is followed by the algorithm of Guazzone and Shi-AC. In the case of BitBrain, BRC-IBMMT reduces the power consumption 13 % compared to Beloglazov algorithm.

Fig (9) shows the performance of all algorithms with respect to SLAVTAH for Google Cluster and Bitbrains

TABLE 7. Simulation result of heterogeneous data center.

	Power (KWh)		SLAVTAH %		VMPD %		ESV		VM migrations	
	Google	Bitbrains	Google	Bitbrains	Google	Bitbrains	Google	Bitbrains	Google	Bitbrains
Beloglazov	69.97	146.25	12.41	4.65	0	0.08	868.3277	680.0625	7391	13317
Chowdhury	66.61	132.53	10.41	2.05	0	0.02	693.4101	271.6865	2473	5787
Guazzone	60.83	130.91	11.77	2.28	0	0.03	715.9691	298.4748	1960	4646
Shi – PU	62.9	131.65	11.34	2.77	0	0.04	713.286	364.6705	2028	5110
Shi – AC	61.1	131.41	11.66	2.18	0	0.03	712.426	286.4738	2275	5297
RUA	68.12	142.35	11.42	1.92	0	0.02	777.9304	273.312	2602	6022
BRC-IBMMT	60.73	129.78	11.44	1.88	0	0.02	694.7512	243.9864	1921	4446

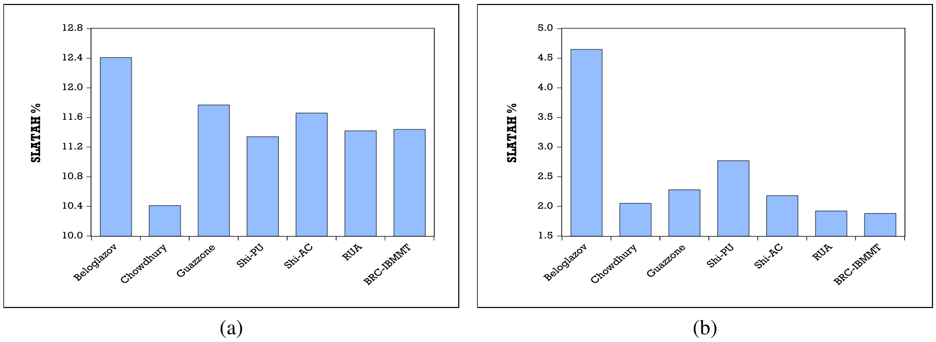


FIGURE 9. SLAVTAH of heterogeneous environment. (a) Google. (b) Bitbrains.

workload, respectively. As shown in Fig (9), our algorithm achieves a good reduction in SLAVTAH besides the power consumption although the negative correlation between them. BRC-IBMMT places the VMs in a tidy balanced way, which increases the resource utilization and prevents the saturation on a single type of resource. Fig(9a) and Fig (9b) shows the reduction for both types of workload. As an instance, in the case of Bitbrain workload, BRC-IBMMT drops the value of SLATAH from 4.65% in Beloglazov algorithm to 1.88%. Fig (10) emphasizes the efficiency of our algorithm, where BRC-IBMMT gets the best result for both types of workload as shown in Fig(10a), and Fig (10b). This result is expected because of the good reduction in both metrics of power consumption and SLA.

As can be seen from Fig (11) that BRC-IBMMT significantly reduces the number of VM migrations. For Bitbrains workload, as an instance, BRC-IBMMT decreases the number of VM migrations from 7391 in Beloglazov algorithm to 1921 in case of Google Cluster, and from 13317 to 4446 in case of BitBrain. Table (7) shows a summary of the result of the heterogeneous environment.

In conclusion, BRC-IBMMT achieves a good balance between the contradictory objectives. In the case of a heterogeneous environment, it was obviously the effect of BRC-IBMMT's balanced placement strategy and selection policy in maximizing the resource utilization and decrease the SLA violation at the same time.

3) DIFFERENT SIZE OF VM EVALUATION

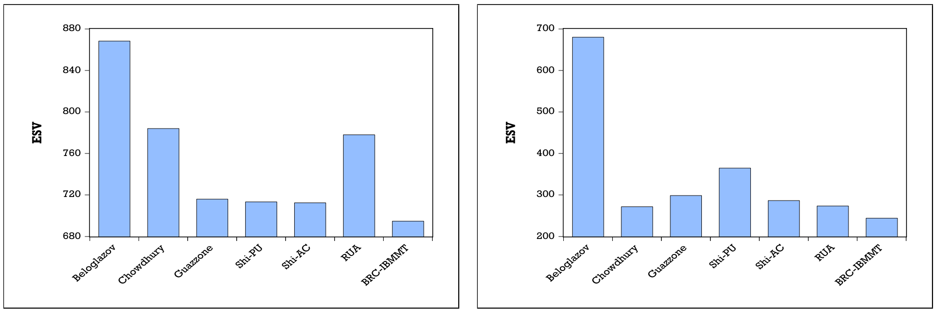
Since our algorithm considers the intra-balance of resource utilization during the placement and selection of VM, it was necessary to evaluate the performance of algorithms for different sizes of VM. In this test, we have changed the size of the VM into three sizes (Small, Normal, and Big). Table (8) shows the specification of the different sizes of VM.

TABLE 8. Different size of VM configuration.

VM type	CPU (MIPS)	RAM (MB)
Small	450,900,1800,2250	413,670,1540
Normal	500,1000,2000,2500	870,1740,630
Big	550,1100,2200,2750	920,1840,750

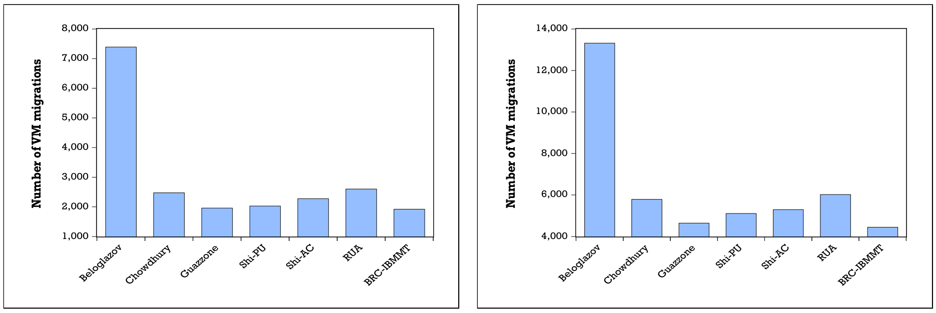
In general, the power consumption is increased gradually with the increasing size of VMs for all algorithms. However, the lowest power consumption level is registered with the algorithm of BRC-IBMMT, as shown in Fig (15). In addition, it was observed that the BRC-IBMMT algorithm performance was better with a smaller size of VMs. The reason for this is that BRC-IBBMT considers the intra-balance of resource utilization during the placement and selection of VM. Therefore, the small size of the VMs leads to low fragmentation of resource. As an instance, BRC-IBMMT reduces the power consumption by 10% compared to Beloglazov algorithm in case of small size VMs, and 8% in case of the big size of VMs.

Fig (13) shows the performance of all algorithms in term of SLAVTAH with respect to the size of VMs. By the same manner, the value of SLATAH is increased with the increase of VM sizes for all algorithms. However, BRC-IBMMT achieves a significant reduction in SLATAH specifically with the small size of VMs. It can be seen also from Fig (13) that BRC-IBMMT decreases the value of SLAVTAH about 22%, 11%, and 10% compared to the best



(a) (b)

FIGURE 10. ESV of heterogeneous environment. (a) Google. (b) Bitbrains.



(a) (b)

FIGURE 11. Number of VM migrations of heterogeneous environment. (a) Google. (b) Bitbrains.

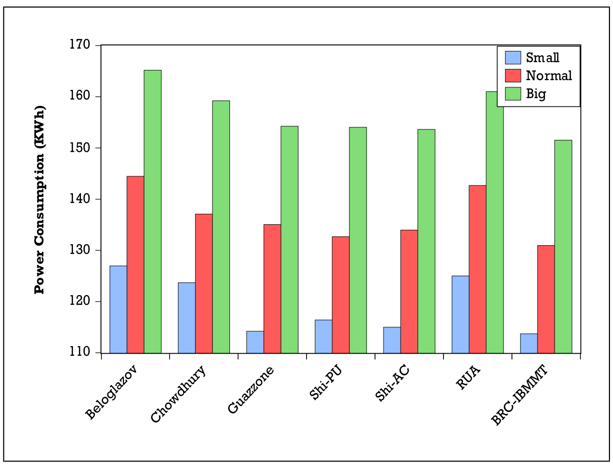


FIGURE 12. Power consumption of different size of VM.

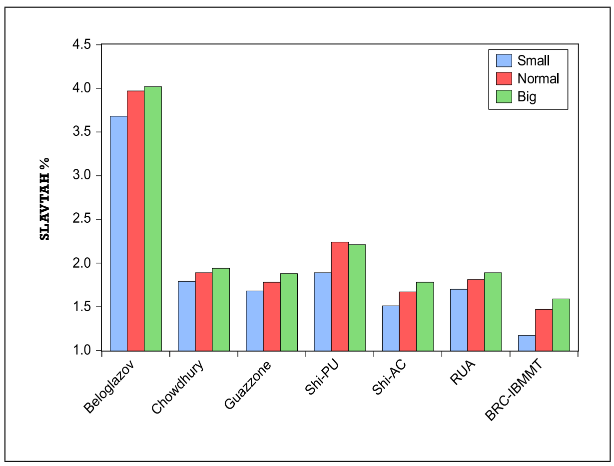


FIGURE 13. SLAVTAH of different size of VM.

algorithm of Shi-AC for three groups of VM size, respectively. For the VM performance degradation, BRC-IBMMT algorithm shows a significant reduction. We can see from the Fig(14) that BRC-IBBMT decreases the VMPD about 50% compared with other algorithms in case of small size VM, and about 33% in case of both sizes of normal and big VMs. The rational reason for that is BRC-IBMMT decreases the number of VM migration and minimizes the migration time efficiently. The superiority of BRC-IBMMT algorithm is clearly presented in Fig (15), which shows the result of ESV metric for all VM sizes. BRC-IBMMT gets the best result for all VM sizes especially with the small size of VM.

Fig (16) shows the number of VM migrations for different sizes of VM. The number of VM migrations is decreased with the increase in the size of VM. For all cases, BRC-IBMMT decreases the number of VM migration efficiently for all groups. According to the results of Fig (16), the best reduction in the number of VM migrations comes with the big size of VM, where BRC-IBMMT decreases the number of VM migration about 13% compared with the best result of Guazzone algorithm.

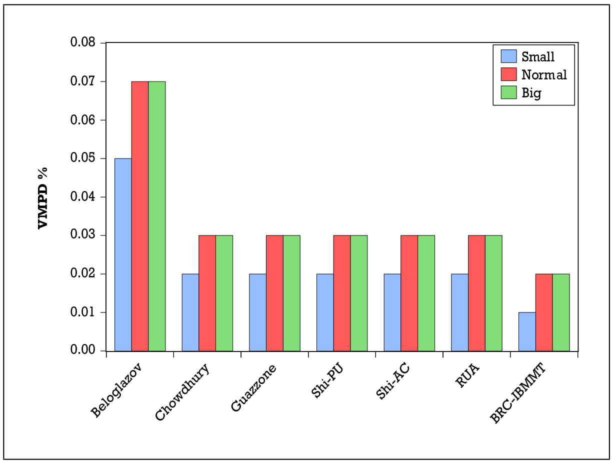


FIGURE 14. SLAVTAH of different size of VM.

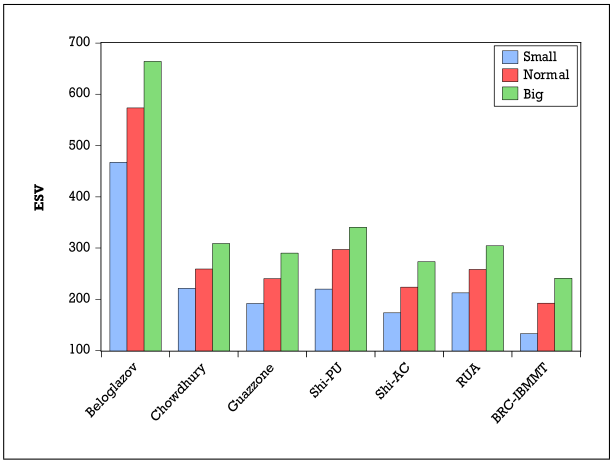


FIGURE 15. ESV of different size of VM.

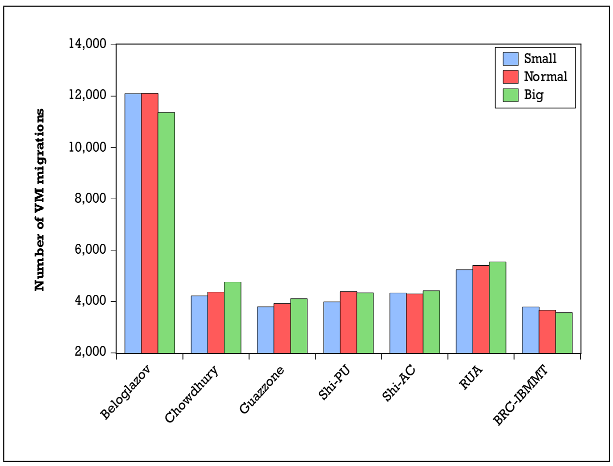


FIGURE 16. Power consumption of different size of VM.

At the end of our evaluation, we can draw the following conclusion: 1) Our algorithm has succeeded to minimize the fragmentation of residual capacity. 2) Our algorithm has succeeded to produce a well-balanced reduction between the negative correlation metrics (power consumption, SLA violation). 3) The size of VM plays the main role in our algorithm, as the best result has appeared clearly by the small size of VM. 4) Our algorithm is more effective in the heterogeneous environment compared with other algorithms which do not consider the intra-balance of resource utilization.

VI. CONCLUSION

Power consumption in the cloud data center has a negative correlation with the SLA violation. While VMs are consolidated in the minimum number of active PMs to efficiently exploit the available resource, resource fragmentation is one of the reasons for resource wastage because of the saturation on single resource restrains the consolidation process. Therefore, we proposed (BRC-IBMMT) algorithm which considers the balance of resource consumption during the VM placement. Because of the balanced resource consumption which leads to efficient resource utilization, BRC-IBMMT succeeded to minimize the waste of the resource on the unavailable PMs. Moreover, the VM selection policy enriches the balance of resource consumption and minimizes the migration cost. The proposed algorithm produced an acceptable balance between the contradictive objectives of minimizing power consumption and SLA violation. The experimental results of different types of workload show that (BRC-IBMMT) algorithm outperform other algorithms in terms of reducing power consumption and SLA violation. Besides the reduction in power consumption and SLA violation, our work is expected to contribute to the sustainable environment by reducing the excessive emission of carbon dioxide. Even though, BRC-IBMMT has provided a practical solution to make an acceptable balance between the objectives of investment in the cloud data center for both providers and customers. But still, the algorithm needs to be tested in a real cloud infrastructure with the inclusion of more resource types. Consequently, future work is intended for the algorithms to be implemented in the most famous cloud computing platform, known as OpenStack.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Commun. ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.

[2] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in *Proc. 10th IEEE Int. Conf. High Perform. Comput. Commun.*, Sep. 2008, pp. 5–13.

[3] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, Jan. 2009.

[4] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 732–794, 1st Quart., 2016.

[5] L. Sharifi, L. Cerdà-Alabern, F. Freitag, and L. Veiga, "Energy efficient cloud service provisioning: Keeping data center granularity in perspective," *J. Grid Comput.*, vol. 14, no. 2, pp. 299–325, Jun. 2016.

[6] X. Li, Z. Qian, S. Lu, and J. Wu, "Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center," *Math. Comput. Model.*, vol. 58, nos. 5–6, pp. 1222–1235, Sep. 2013.

[7] L. Yin, J. Luo, and H. Luo, "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4712–4721, Oct. 2018.

[8] J. Luo, W. Song, and L. Yin, "Reliable virtual machine placement based on multi-objective optimization with traffic-aware algorithm in industrial cloud," *IEEE Access*, vol. 6, pp. 23043–23052, 2018.

[9] W. Yue and Q. Chen, "Dynamic placement of virtual machines with both deterministic and stochastic demands for green cloud computing," *Math. Problems Eng.*, vol. 2014, Jan. 2014, Art. no. 613719.

[10] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proc. 8th Int. Workshop Middleware Grids, Clouds e-Sci.*, Dec. 2010, p. 4.

[11] M. Guazzone, C. Anglano, and M. Canonico, "Exploiting VM migration for the automated power and performance management of green cloud computing systems," in *Energy Efficient Data Centers* (Lecture Notes in Computer Science), vol. 7396, J. Huusko, H. de Meer, S. Klingert, and A. Somov, Eds. Berlin, Germnay: Springer, 2012.

[12] M. R. Chowdhury, M. R. Mahmud, and R. M. Rahman, "Study and performance analysis of various VM placement strategies," in *Proc. IEEE/ACIS 16th Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput.(SNPD)*, Jun. 2015, pp. 1–6.

[13] L. Shi, J. Furlong, and R. Wang, "Empirical evaluation of vector bin packing algorithms for energy efficient data centers," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2013, pp. 9–15.

[14] 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 Comput., Pract. Exper.*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.

[15] W. Song, Z. Xiao, Q. Chen, and H. Luo, "Adaptive resource provisioning for the cloud using online bin packing," *IEEE Trans. Comput.*, vol. 63, no. 11, pp. 2647–2660, Nov. 2014.

[16] A.-M. Ammar, J. Luo, and Z. Tang, "An anti-overload model for openstack based on an effective dynamic migration," *KSII Trans. Internet Inf. Syst.*, vol. 10, no. 9, pp. 4165–4187, Sep. 2016.

[17] G. Han, W. Que, G. Jia, L. Shu, and A. Jara, "An efficient virtual machine consolidation scheme for multimedia cloud computing," *Sensors*, vol. 16, no. 2, p. 246, 2016.

[18] L. Zhao, L. Lu, Z. Jin, and C. Yu, "Online virtual machine placement for increasing cloud provider's revenue," *IEEE Trans. Services Comput.*, vol. 10, no. 2, pp. 273–285, Mar./Apr. 2017.

[19] F. Hao, M. Kodialam, T. V. Lakshman, and S. Mukherjee, "Online allocation of virtual machines in a distributed cloud," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 238–249, Feb. 2017.

[20] A. Al-Moalmi, J. Luo, A. Salah, and K. Li, "Optimal virtual machine placement based on grey wolf optimization," *Electronics*, vol. 8, no. 3, p. 283, Mar. 2019.

[21] H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang, and Q. Zheng, "Power-aware and performance-guaranteed virtual machine placement in the cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 6, pp. 1385–1400, Jun. 2018.

[22] S. Martello, D. Pisinger, and D. Vigo, "The three-dimensional bin packing problem," *Oper. Res.*, vol. 48, no. 2, pp. 256–267, Apr. 2000.

[23] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *Proc. IEEE Int. Conf. Cloud Comput.*, Dec. 2009, pp. 254–265.

[24] J. A. Wickboldt, R. P. Esteves, M. B. de Carvalho, and L. Z. Granville, "Resource management in IaaS cloud platforms made flexible through programmability," *Comput. Netw.*, vol. 68, pp. 54–70, Aug. 2014.

[25] Z. Xiao, Q. Chen, and H. Luo, "Automatic scaling of Internet applications for cloud computing services," *IEEE Trans. Comput.*, vol. 63, no. 5, pp. 1111–1123, May 2014.

[26] J. Xu and J. A. B. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *Proc. IEEE/ACM Int. Conf. Green Comput. Commun. Int. Conf. Cyber, Phys. Social Comput.*, Dec. 2010, pp. 179–188.

[27] 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," *J. Comput. Syst. Sci.*, vol. 79, no. 8, pp. 1230–1242, 2013.

[28] A. Varasteh and M. Goudarzi, "Server consolidation techniques in virtualized data centers: A survey," *IEEE Syst. J.*, vol. 11, no. 2, pp. 772–783, Jun. 2017.

[29] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting Methods and Applications*, 2008.

[30] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in *Proc. Int. Conf. High Perform. Comput. Simulation*, Jun. 2009, pp. 1–11.

[31] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, and M. A. Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis," in *Proc. 3rd ACM Symp. Cloud Comput.*, Oct. 2012, p. 7.

[32] S. Shen, V. van Beek, and A. Iosup, "Statistical characterization of business-critical workloads hosted in cloud datacenters," in *Proc. 15th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2015, pp. 465–474.

[33] P. H. P. Castro, V. L. Barreto, S. L. Corrêa, L. Z. Granville, and K. V. Cardoso, "A joint CPU-RAM energy efficient and SLA-compliant approach for cloud data centers," *Comput. Netw.*, vol. 94, pp. 1–13, Jan. 2016.