The following publication J. V. Wang, N. Ganganath, C. Cheng and C. K. Tse, "Bio-Inspired Heuristics for VM Consolidation in Cloud Data Centers," in IEEE Systems Journal, vol. 14, no. 1, pp. 152-163, March 2020. is available at https://dx.doi.org/10.1109/JSYST.2019.2900671

Bio-Inspired Heuristics for VM Consolidation in Cloud Data Centers

Jing V. Wang, *Student Member, IEEE*, Nuwan Ganganath, *Member, IEEE*, Chi-Tsun Cheng, *Member, IEEE*, and Chi K. Tse, *Fellow, IEEE*

***Abstract*—In Infrastructure-as-a-Service environments, Cloud data centers employ virtualization technology to host various applications in virtual machines (VMs) and enable application isolation on shared physical resources. Additionally, live VM migration has been adopted to perform load balancing by moving VMs across distinct hosts. However, co-located VMs which show significant positive correlations on their CPU utilization patterns are at a higher risk of triggering overloading events and incurring performance degradation, even when their host is operating below its critical limits. To address this problem, a VM consolidation mechanism inspired by host-switching behaviors in symbiotic associates is proposed in this paper. In the proposed mechanism, hosts and VMs in Cloud data centers represent symbionts in an ecosystem. Two heuristic functions, inspired by host susceptibility and symbiotic coefficient among symbionts, are proposed to yield better resource utilization via VM consolidations. Experiment results demonstrate that the proposed mechanism can achieve reductions in both energy consumption and Service Level Agreement (SLA) violations of Cloud data centers.**

***Index Terms*—Resource management, heuristics, bio-inspired, utilization correlation, VM consolidation.**

I. Introduction

CLOUD computing is recently gaining rapid prominence by pooling resources for on-demand computing. Cloud technology enables users to access a large pool of computational and storage resources on an on-demand basis [1]. Virtualization, the enabling technology of Cloud computing, allows physical resources to be shared by multiple isolated virtual machines (VMs). Efficient virtualization technology makes Cloud applications possible and allows them to proliferate further. However, such soaring demands have led to high energy consumption in Cloud clusters. In 2014, 70 billions kWh of energy was consumed by Cloud clusters in US [2], which has been one of the major sources of carbon dioxide emissions. Therefore, with the unprecedented development of Cloud clusters in both their scale and complexity, their energy consumption has become a key problem that needs to be addressed.

On the other hand, guaranteeing the required Quality of Servive (QoS) of Cloud applications is an essential task for Cloud service providers (CSPs) [3]. The desired level of QoS is expressed in form of Service Level Agreements (SLAs). During a VM consolidation process, the workload of applications on the VMs may vary dynamically. Such fluctuations may cause server overload and will affect the performance of all VMs on the overloaded servers and thus lead to significant SLA violations. Therefore, how to lower the risk of overloading is an important issue that needs to be tackled in maintaining application QoS.

Live migration allows VMs to move across different hosts with virtually no interruption [4]. In the VM consolidation problem, with live migration technology, VMs can be properly consolidated onto physical hosts for better resource utilization and energy saving in Cloud data centers. Excess load will be migrated out from overloaded hosts to under-utilized hosts to eliminate hotspots. However, co-located VMs may trigger overloading incidents if majority of their applications reach their peak utilization level simultaneously [5]. Thus, to avoid potential violations of SLA, correlation information among co-located VMs has to be considered in the VM consolidation process.

Bondings among VMs and hosts in Cloud data centers share a lot of characteristics and features with organisms in natural with symbiotic relationship (*i. e.* parasites and hosts), who are living and evolving together. During the evolutionary process, parasites may switch their hosts if their living environments are not suitable for survival any more [6]. Parasites are more likely to switch to hosts with adequate resources and compatible symbionts during periods of environmental change [7].

Inspired by host-switching behaviors in symbiotic associates, in this paper, a VM consolidation mechanism based on bio-inspired heuristics is proposed to tackle the challenges of energy saving and QoS management in Cloud data centers. In the proposed mechanism, hosts and VMs in Cloud data centers represent symbionts in ecosystems. We propose two heuristic functions based on utilization levels of hosts and Resource Utilization Correlation (RUC) among co-located VMs. The concept of host susceptibility in [8] is adopted here to evaluate hosts' condition according to their utilization levels. Inspired by mutual interactions among symbionts [9], symbiotic coefficient among parasites is adopted to evaluate correlations among VMs. In the proposed mechanism, VMs share resources provided by the physical host to keep its utilization at a relatively moderate level. By considering both hosts' utilization levels and RUC among co-located VMs, the proposed mechanism addresses the VM consolidation problem with the objective of reducing energy consumption while maintaining a low number of SLA violations in Cloud clusters.

The main contributions of this paper are summarized as follows:

- Inspired by host-switching behaviors in symbiotic associates, we propose a bio-inspired heuristics-based VM consolidation mechanism to tackle the challenges of energy saving and QoS management in Cloud data centers.

- We propose two heuristic functions, namely host susceptibility and symbiotic coefficient, to evaluate hosts' condition and correlations among VMs, respectively.
- We propose a VM migration algorithm considering a heuristics-based fitness for optimizing VM reallocations.
- We conducted extensive experiments using CloudSim with real-world workload data. The experimental results show that the proposed mechanism can avoid SLA violations and achieve lower overall energy consumption. Moreover, the proposed mechanism significantly reduces the risk of overloading and avoids resource waste.

The rest of this paper is arranged as follows. Section II discusses some related works. Section III introduces host-switching behaviors in symbiotic associates and preliminaries on the correlations among co-located VMs in detail. In Section IV, formulations of the proposed heuristic functions and their rationales are given. Section V elaborates the details on the proposed VM consolidation mechanism. Details on the experiment setup are described in Section VI. Experiment results and discussions of the proposed VM consolidation mechanism are analyzed in Section VII. Finally, Section VIII gives the concluding remarks.

II. Related Work

This section discusses related work on bio-inspired algorithms and correlation-based methods for VM consolidation problem.

A. Bio-Inspired Algorithms for VM Consolidation

A number of VM consolidation approaches have adopted bio-inspired designs [10]–[15] for better resource management in Cloud data centers. An ant colony based algorithm was proposed by Farahnakian *et al.* in [10] to reduce energy consumption. They introduced a pseudo-random-proportional-rule as an efficient resource management procedure in their ant colony based system. Liu *et al.* [11] also addressed the VM placement problem by minimizing the number of active hosts using the ant colony optimization (ACO) algorithm. They adopted order exchange and migration local search techniques, which swap and migrate VMs between different servers, to enhance searching efficiency. A symbiotic organism search (SOS) algorithm is adopted as an efficient solution to achieve higher system utilization with minimal makespan in [12]. They proposed a discrete SOS algorithm for optimal scheduling of tasks in Cloud data centers. In their mechanism, a mutual benefit factor facilitates the exploration of new regions in the search space, while a parasite vector prevents premature convergence of the system. However, the discrete SOS algorithm in [12] is to schedule multiple tasks in a Cloud data center without considering the problem of VM migration. An energy-efficient virtual resource dynamic integration method was proposed by Wen *et al.* based on an improved genetic algorithm (GA) [13]. In their placement algorithm, the termination condition of GA has been improved to avoid getting stuck at local optimal points. While in [14], the authors adopted GA to forecast the resource utilization and energy consumption in Cloud data centers. The VM placement can then be improved based on the prediction results. A dynamic power-saving resource allocation mechanism was proposed by Chou *et al.* based on a particle swarm optimization (PSO) algorithm [15]. In their PSO algorithm, they considered the energy consumption of both hosts and air conditioners as the fitness function for energy saving. However, the computational complexity of their approaches makes them impractical for real-world applications.

B. Correlation-based Methods for VM Consolidation

As mentioned earlier, correlated VMs co-located on the same host are very likely to impact application performance. To avoid performance degradations, previous works [5], [16]–[20] have considered the utilization correlation information among co-located VMs in VM consolidation processes.

A power management solution was presented by Kim *et al.* in [16] to host scale-out applications in Cloud clusters. First, they conducted comparative analysis on workload characteristics of applications and proposed a cost function to quantify correlations between two selected VMs for server consolidation. Then they determined an optimal voltage to frequency ratio (v/f) for each server according to the estimated cost level of those co-located VMs. They jointly utilized server consolidation and v/f scaling by considering correlation information among VMs to reduce global power consumption. In [5], an *affinity* model was proposed to explore the relationship among VMs based on the predicted utilization values provided by an autoregressive integrated moving average prediction model. In an algorithm proposed in [5], VMs with high affinity will be consolidated together for better resource utilization. In [17], a two-phase multi-objective VM placement scheme was presented by Pahlevan *et al.* for geo-distributed data centers. In the global phase, they exploited data and CPU-load correlations among VMs for clustering VMs based on the holistic knowledge of their characteristics. In the local phase, CPU-load correlation is considered as the only allocation criterion. Their two-phase VM placement scheme aims to tackle the challenges in cost-performance and energy-performance trade-offs. However, the host overloading problem in the VM consolidation process has not been addressed in aforementioned works.

A VM placement scheme was proposed by Wei *et al.* in [18] to guarantee reasonable QoS level. First, an autoregressive integrated moving average model was adopted to predict the future trend. Then, the volatility of the future demand was analyzed based on a generalized autoregressive conditional heteroskedasticity model. Their placement scheme, which takes account of VM correlation, is based on a modern portfolio theory to achieve higher utilization and lower overload ratio. In [19], performance interferences under different combinations of workloads were studied experimentally. Furthermore, a performance interference prediction model was developed to manage application QoS in Clouds based on the application-level and VM-level characteristics of co-located applications. In [20], another interference prediction model was proposed by Zhu and Tung to estimate the application QoS metric. In their proposed model, an influence matrix, which considers

interferences from all types of resources, is presented to estimate the extra resources requested by an application for optimal consolidation configuration. All of these approaches are designed to achieve a desired QoS. However, none of them takes energy consumption into account in the VM consolidation process. Our previous works [21], [22] provided insights on the usage of correlation information as a parameter for decision making in the VM consolidation process.

The work in [23] presented several approaches to tackle an energy-aware scheduling problem. In their power-based methods, the destination host is chosen based on its recent power consumption readings. A similar problem was studied in [24]. A host with the least increase in estimated power consumption after taking up a migrated VM is chosen as the destination for migration. However, this paper considers both host utilization levels and RUC among co-located VMs and proposes a set of solutions for better resource management, including two heuristic functions, global tuning-based hotspot detection, and VM migration algorithm considering a heuristics-based fitness. Our VM consolidation mechanism aims to obtain reasonable trade-offs between energy consumption and SLA violations in Cloud clusters.

III. Background

This section introduces the background on symbiosis and host switching behavior in ecosystems and preliminaries on the correlations among co-located VMs in detail.

A. Symbiosis and Host Switching

The term symbiosis was first used in 1879 to describe the cohabitation behavior between two different biological organisms [25]. To survive, organisms choose to live together in a reliance-based relationship. This kind of symbiotic behavior is ubiquitous in terrestrial, freshwater, and marine communities [26], [27]. Undoubtedly, symbiosis has played an important role in biological evolution in ecosystems.

The generation of biological diversity is accompanied by multiple evolutionary host switches. Host switching is a necessary condition to keep pace in an evolutionary race [28], [29]. A common evolutionary host switching occurs when host utilization capabilities are acquired rapidly or the living environment of parasites is harmed [6]. Thus, parasites may switch to a new host with better fitness and survival advantages.

In general, the process of host switching consists of three basic stages [7], those are (i) Opportunity: It is an essential condition for a parasite to switch to a new host; (ii) Compatibility: After an opportunity presents, parasites and their corresponding hosts should be compatible with each other for cohabitation [6]. It is necessary for parasites to overcome physical barriers (*e.g.* epidermis, exoskeleton, etc.) imposed by the new host without impacting the survival of the species involved. Furthermore, a compatible host should provide adequate resources as a food-source and substrate for parasite survival; and (iii) Conflict resolution: During the process of host-parasite coexistence, conflicts may arise subsequently. The host and parasites should resolve such conflicts for mutual adaptation and better survival.

B. Multiple Correlation Coefficient

In this paper, the multiple correlation coefficient [30], as described in our previous work [22], was adopted to estimate the RUC among co-located VMs. In multiple regression analysis, the multiple correlation coefficient is commonly used to measure the accuracy of the predicted dependent variable. The value of a multiple correlation coefficient varies between 0 and 1. It is 0 if there is no relationship between those variables and 1 if those variables are perfectly correlated.

Assuming that there are n VMs on a host. We denote these co-located VMs using vector $\mathbf{V} = [V_1, V_2, ..., V_n]$. The RUC level of the j^{th} VM toward the other $n-1$ VMs is measured based on their last q CPU utilization observations. We denote the last q observations of the j^{th} VM using vector $\mathbf{y}_j$. Similarly, we denote $\mathbf{X}$ as an augmented matrix contains the q observations of the remaining $n-1$ VMs on the host. The vector $\mathbf{y}_j$ and matrix $\mathbf{X}$ can be expressed as

$$\mathbf{y}_j = \begin{bmatrix} y_{1,j} \\ \vdots \\ y_{q,j} \end{bmatrix}, \ \mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,m} & \cdots & x_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & x_{p,1} & \cdots & x_{p,m} & \cdots & x_{p,n-1} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & x_{q,1} & \cdots & x_{q,m} & \cdots & x_{q,n-1} \end{bmatrix}.$$

Here, variable $x_{p,m}$ represents the p^{th} CPU utilization observation of V_m. The multiple correlation coefficient $R^2_{V_j,\mathbf{V}\setminus V_j}$ for each V_j can then be calculated as

$$R^2_{V_j,\mathbf{V}\setminus V_j} = \frac{\sum_{k=1}^{q}(y_{k,j} - m_{\mathbf{y}_j})^2(\hat{y}_{k,j} - m_{\hat{\mathbf{y}}_j})^2}{\sum_{k=1}^{q}(y_{k,j} - m_{\mathbf{y}_j})^2 \sum_{k=1}^{q}(\hat{y}_{k,j} - m_{\hat{\mathbf{y}}_j})^2},$$

where $\mathbf{V} \setminus V_j$ is the vector representing the VMs on the host except the j^{th} VM. The variables $m_{\mathbf{y}_i}$ and $m_{\hat{\mathbf{y}}_j}$ represent the means of $\mathbf{y}_j$ and $\hat{\mathbf{y}}_j$, respectively. Here, $\hat{\mathbf{y}}_j$ is a vector of predicted values of the j^{th} VM, which can be obtained as

$$\hat{\mathbf{y}}_j = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}_j.$$

where $\mathbf{X}^{\mathrm{T}}$ is the transpose matrix of $\mathbf{X}$ and $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ is the inverse matrix of $(\mathbf{X}^{\mathrm{T}}\mathbf{X})$. In our model, if $(\mathbf{X}^{\mathrm{T}}\mathbf{X})$ is singular, the multiple correlation coefficient of V_j is expressed as its current CPU utilization. In this work, the RUC between the j^{th} VM and other co-located VMs is represented by the corresponding multiple correlation coefficient between both parties.

IV. Heuristic Formulations

In nature, symbiotic organisms live together for sustenance and survival. In Cloud data centers, hosts and VMs are associated with a similar relationship. Similar to host-switching behaviors in symbiotic associates, VMs in Clouds are commonly migrated to different hosts for better performance.

As mentioned earlier, there are three stages in the process of host-switching in symbiotic associates. In the compatibility stage, parasites prefer switching to compatible hosts with adequate resources for better survival. Similar to this phenomenon, VMs in Clouds are preferable to be allocated to hosts with more available resources. Therefore, host utilization level should be considered as a migration criterion. Moreover,

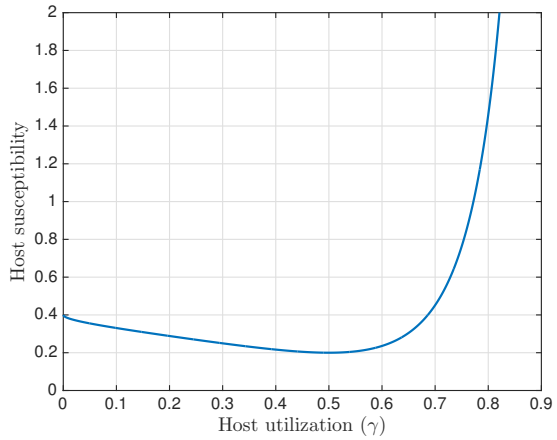


Fig. 1: An illustration of the host susceptibility h_1, with $a = 0.4$, $b = 0.5$, and $c = 0.2$.

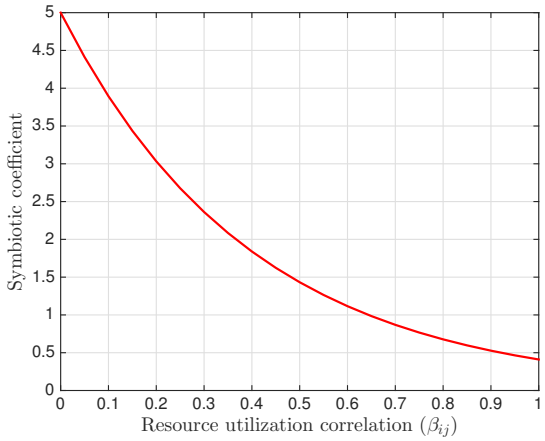


Fig. 2: An illustration of the symbiotic coefficient h_2, with $m = 5$ and $n = -2.5$.

the stage of conflict resolution inspires us to take mutual interactions among co-located organisms into account in the VM consolidation process. In this work, resource utilization correlations are used to represent the interactions among co-located VMs. For a physical host, it is more likely for VMs with high RUC to their co-located VMs to trigger overloading events. Due to the heterogeneity of hosts and VMs, such a problem cannot be completely resolved by imposing static utilization thresholds to control the utilization level of hosts.

Inspired by host-switching behaviors exhibited in symbiotic organisms, we formulate the host utilization level and the RUC among co-located VMs as two heuristic functions [22] to evaluate the state of each host and VM for making allocation decisions. The bio-inspired heuristic functions assign low symbiotic coefficient [9] values to VMs with high correlations in their CPU utilization patterns for co-location avoidance. Conversely, hosts with high utilization levels are considered as susceptible to prevent VMs from migrating onto them. Such kind of hosts may even expel some of their VMs.

A. Host Susceptibility

In nature, a non-immune host is one who has little resistance against a particular organism, thus it is susceptible to be infected by parasites [31]. In contrast, hosts with fewer resources are also susceptible to parasites infection since they have fewer resources to allocate to immune functions or to other defenses against parasites [32]. Similarly, in Cloud data centers, hosts with extreme utilization levels are operating outside their maximum efficiency ranges. Therefore, keeping host utilization at relatively moderate levels is highly recommended. Because of that, we formulate host susceptibility h_1, which corresponds to the utilization level of a host, to evaluate host state as

$$h_1(\gamma) = \frac{(a-c)(1-\sqrt{b})^2}{b}\left(\frac{1}{1-\sqrt{\gamma}} - \frac{1}{1-\sqrt{b}}\right)^2 + c, \quad (1)$$

where $\gamma \in [0, 1]$ is the CPU utilization of a host. Here, γ is highly correlated to the current CPU utilization of co-located VMs. In (1), a represents the intrinsic susceptibility of a host, *e.g.* $h_1(0) = a$. In nature, once a host has been infected, its immune system would be built up. Such hosts would become less susceptible to be infected by parasites but more attractive to their mutualists. Similarly, once a host is utilized in Cloud data centers, it is highly recommended to optimize its utility by encouraging more loads. Because of that, the susceptibility value is being decreased until it reaches a minimum value at a certain point, *e.g.* $h_1(b) = c$. Here, b represents the optimum utilization and c represents the minimum host susceptibility level. In (1), a, b, and c are constants, which should be selected as $a > 0$, $1 \geq b \geq 0$, and $a > c \geq 0$. In this work, they are selected as $a = 0.4$, $b = 0.5$, and $c = 0.2$ to ensure hosts with extreme utilization will have relatively higher susceptibility values. Characteristics of h_1 versus host utilization level γ are illustrated in Fig. 1. Using (1), we define an occupied capacity of an active host i as

$$S(\gamma_i) = \int_0^{\gamma_i} h_1(\gamma) d\gamma. \quad (2)$$

Here, the occupied capacity is used in evaluating host operation level, which its usage will be elaborated shortly.

The rationale behind (1) is that the susceptibility value of a host is high when its utilization is low to avoid unnecessarily provisioning new hosts. Furthermore, the host susceptibility value goes to infinity as its utilization reaches 100% to discourage more loads (parasites) from overloading a host. VMs can therefore use susceptibility as an indicator and try to pick hosts with more available resource and desirable operating environment (*i.e.* those with lower susceptibility values). Details will be explained in the later sections.

B. Symbiotic Coefficient

The level of mutual interactions among parasites is characterized by their symbiotic coefficients [9]. Here, we formulate a symbiotic coefficient (SC) h_2, which corresponds to the RUC among co-located VMs, to evaluate the mutual interactions among VMs. In the proposed mechanism, VMs with high CPU utilization correlations are less likely to be co-located on the same host. Therefore, such VMs will be assigned with low

SC values for co-location avoidance. An exponential function is chosen here such that VMs with significantly high CPU utilization correlations will have lower SC values, which will encourage them to migrate onto different hosts. Here, h_2 is formulated as

$$h_2(\beta_{ij}) = m \exp(n\beta_{ij}), \tag{3}$$

where $\beta_{ij} \in [0, 1]$ denotes the RUC of VM j to other co-located VMs on host i. Here, β_{ij}, which refers to $R^2_{V_j, \mathbf{V}\setminus V_j}$ mentioned in Section III, is related to last q CPU utilization observations of co-located VMs. Parameters m and n are constants, which should be selected as $m > 0$ and $n < 0$. In this work, they are selected as $m = 5$ and $n = -2.5$ to give lower SC values to VMs with higher RUC values. Fig. 2 illustrates SC versus RUC.

C. Capacity Threshold

To evaluate VMs under different conditions, (2) and (3) are integrated to form a global tuning parameter

$$C_{\text{global}} = \int_0^{\mathrm{T}_{ij}} \frac{1}{h_2(\beta_{ij})} h_1(\gamma) d\gamma. \tag{4}$$

Here, T_{ij} is the corresponding utilization threshold for VM j on host i. To keep each VM operated normally, the utilization level of an active host cannot exceed the utilization threshold of its VMs. According to (2), a capacity threshold for each VM j on host i is calculated as

$$S(\mathrm{T}_{ij}) = C_{\text{global}} \times h_2(\beta_{ij}). \tag{5}$$

D. Capacity Margin

The capacity margin of each VM is calculated as

$$\mathrm{M}_{ij} = \begin{cases} S(\mathrm{T}_{ij}) - S(\gamma_i) & \text{if } S(\gamma_i) < S(\mathrm{T}_{ij}), \\ 0 & \text{if } S(\gamma_i) \geq S(\mathrm{T}_{ij}). \end{cases} \tag{6}$$

It is assumed that each VM on host i is assigned with a non-zero margin if the occupied capacity of host i did not exceed the capacity threshold of VM j. In the proposed mechanism, VMs with low h_2 values are assigned with zero margin if they are currently accommodated on a host with an extreme utilization level. Since such states are more likely to trigger overloading incidents, VMs with zero margin are suggested to be migrated to a more suitable host.

E. Fitness

All the available hosts will be evaluated such that suitable hosts can be selected as destination hosts for VM reallocations. In this work, we formulate a heuristics-based fitness function for each host as

$$\text{Fit}(i) = \frac{\mathrm{M}_{ij}}{\left(1 + \left|\frac{\mathrm{d}h_1(\gamma_i')}{\mathrm{d}\gamma_i'}\right|\right)}, \tag{7}$$

where M_{ij} is the capacity margin of VM j if it is migrated to host i. In (7), γ_i' is the utilization level of host i after receiving the migrating-in VM j.

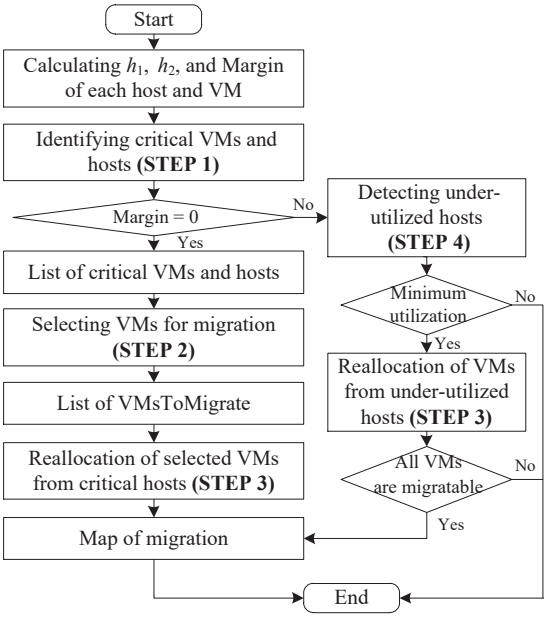


Fig. 3: An iteration of the proposed VM consolidation process.

V. Proposed VM Consolidation Mechanism

In this work, an ordinary Cloud data center with an Infrastructure as a Service (IaaS) model is analyzed. Different applications are allowed to be allocated onto the same physical host for energy saving. Furthermore, VMs can be migrated across the whole data center to yield better utilization. The proposed VM consolidation mechanism is executed in four steps: (1) identifying critical VMs and hosts, (2) selecting VMs for migration, (3) reallocating selected VMs, and (4) detecting under-utilized host. The proposed VM consolidation process is summarized in Fig. 3 and described in details below.

A. Identifying Critical VMs and Hosts

The objective of the first step is to identify VMs and hosts that are regarded as critical. This step is triggered periodically according to the specified interval of the CSP. Whenever the mechanism is invoked, the susceptibility value h_1, the SC value h_2, and the capacity margin of each host and its VMs will be calculated. Here, VMs with zero margin, *i.e.* lower SC values or higher susceptibility values, are considered as critical VMs. In the proposed mechanism, if there exist any critical VM on a host, that host is regarded as critical. As critical hosts often degrade application performance, migrating critical VM(s) away can prevent potential SLA violations.

B. Selecting VMs for Migration

Once critical hosts are identified, one or multiple of their VM(s) will be selected for migration. In the second step, the proposed mechanism selects VM(s) to be migrated according to their migration time. Here, the migration time is calculated based on the amount of RAM being used by the VM divided by the available network bandwidth between source and destination hosts. On a critical host, its critical VM that requires the shortest time for migration will be given the highest priority to

Algorithm 1: The modified BFD with the proposed heuristics

Require: VmsToMigrateList, hostList
Ensure: $migrationMap$

```
VmsToMigrateList.sortDecreasingUtilization()
for vm in VmsToMigrateList do
    maxFitness ← MIN
    allocatedHost ← NULL
    for host in hostList do
        if host has enough resource then
            h1 ← estimated h1
            h2 ← estimated h2
            Margin ← estimated Margin
            if Margin > 0 then
                Fitness ← estimated Fitness
                if Fitness > maxFitness then
                    allocatedHost ← host
                    maxFitness ← Fitness
                end if
            end if
        end if
    end for
    migrationMap.add(vm, allocatedHost)
end for
return migrationMap
```

be migrated. As long migration time can have negative impacts on application performance, such design can lower the chance of having SLA violations. After each selection, the values of h_1, h_2, and the capacity margin of the critical host and its VMs will be updated. This step is executed repeatedly until no more critical VMs can be found on the host.

C. Reallocating Selected VMs

To accommodate the migrated out VMs in the second step, suitable hosts will be selected for VM reallocations. This step is executed in two stages: (1) reallocation of selected VMs from critical hosts and (2) reallocation of VMs from under-utilized hosts. The information on the new VM placement from these two stages constitutes a migration map. VMs are then consolidated according to the migration map.

The problem addressed in this step can be viewed as a bin packing problem with variable bin sizes and costs. In this work, the bin size is representing the available CPU resource of each physical host, items are representing the selected VMs obtained from the second step while costs are corresponding to the heuristics-based fitness values of the selected VMs if they are reallocated onto different hosts. Here, we adopt a modified best fit decreasing (BFD) algorithm together with the proposed bio-inspired heuristic functions introduced in Section IV to solve the bin packing problem. The modified BFD is regarded as efficient as it uses no more than $71/60 \cdot OPT + 1$ bins in its operation (where OPT is the number of bins provided by the optimal solution) [33].

The proposed heuristics can be integrated with the BFD algorithm and it is presented in Algorithm 1. After selecting VMs for migration in the second step, a list of VMs that need to be reallocated is obtained. According to current CPU utilizations of selected VMs in the second step, they are first sorted in a decreasing order in the modified BFD. For each VM on the sorted list, we try to find a host with adequate resources to accommodate it. For each available host, the values of h_1, h_2, and the capacity margin of that host and its VMs (including the sorted VM) after migration will then be estimated. Note that a host, which would become critical after accepting a VM, will not be selected. For each VM, a host that can yield the largest fitness value after migration will be chosen as a destination host, *i.e.* a host with both lower susceptibility value and higher SC value. In summary, the algorithm reallocates a critical VM to a host with the largest margin and a moderate utilization level. This allows critical VMs to choose more capable hosts and avoid co-locating with VMs with similar utilization patterns. If no active host can accommodate the migrated out critical VMs, an inactive host which can yield the largest fitness among switched-off hosts, will be turned on. Once a host is located, the migration will proceed. This step is repeated until all the VMs in the migration list are reallocated. The reallocation process allows hosts to operate at desired utilization levels.

D. Detecting Under-Utilized Hosts

In Clouds, hosts with relatively low utilizations, even being idle, could still reach 70% of their peak power. Therefore, turning off under-utilized hosts is highly recommended for energy saving. Among the active hosts, the host with the minimum utilization will be regarded as the under-utilized host. Note that hosts which have been considered as critical in the first step or have accepted VM(s) in the third step, will not be considered as under-utilized. For an under-utilized host, the proposed mechanism tries to find destination host(s) to accommodate each of its VM(s) and then checks if such a host can place all its VMs on other hosts without introducing new critical host(s). Following the same logic mentioned earlier, the mechanism selects a destination host that can yield the largest fitness value from the available hosts which are better than the source host. The source host is turned off only if all of its VMs can be migrated away. Otherwise, no changes will be applied. This process is then repeated on the host having the next minimum utilization.

Note that the fitness value of a destination host should be larger than that of the under-utilized host in the VM reallocation process. Otherwise, the newly migrated-in VM(s) would trigger overloading on the host assigned and cause unnecessary migrations in the coming rounds.

E. A Worked Example

The rationale of the proposed VM consolidation mechanism can be further elaborated using the following worked example.

Example 1: Consider a Cloud data center with 5 physical hosts $P_1, P_2, \ldots, P_5$, and 10 VMs $V_1, V_2, \ldots, V_{10}$ as shown in Fig. 4. Suppose the global tuning parameter C_{global} is 0.5. For each host, the number inside its bracket indicates its current CPU utilization. For each VM, the numbers inside its bracket

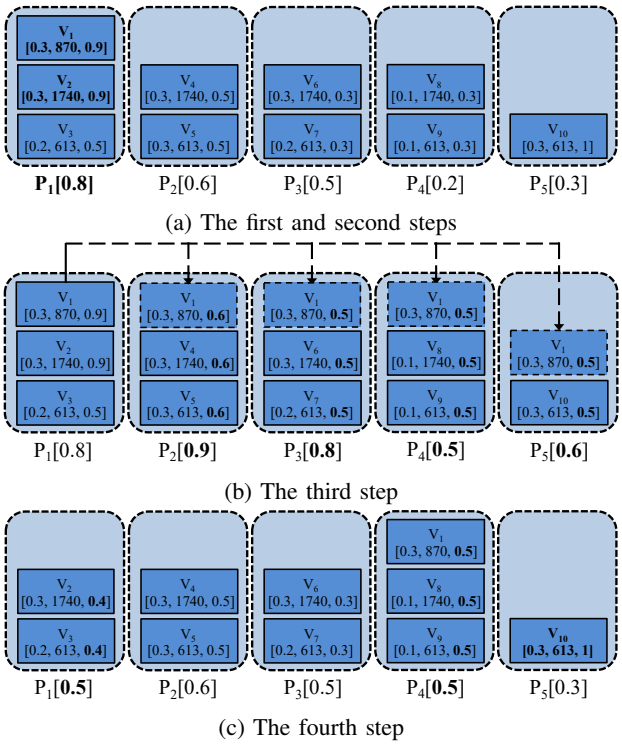


Fig. 4: A worked example of the proposed mechanism.

respectively represent its current CPU utilization, amount of RAM, and RUC level.

For hosts, the proposed mechanism first calculates their M_{ij} for each VM on them. Note that as $M_{11} = 0$ and $M_{12} = 0$, thus V_1 and V_2 will be considered as critical VMs. Therefore, P_1 is regarded as critical. Between these two critical VMs on P_1, V_1 requires shorter migration time. Therefore, V_1 will be selected for migration. After V_1 being chosen, M_{12} and M_{13} will be updated. Since $M_{12} > 0$ and $M_{13} > 0$, after the migration of V_1, no critical host is found in the data center.

By the end of the second step, V_1 is migrated out for reallocation. The proposed mechanism will proceed to its third step and calculates M_{i1} for the remaining 4 physical hosts P_2, P_3, P_4, and P_5. Among the remaining hosts, P_2 is not feasible as it will be regarded as critical after accepting V_1. Among the feasible physical hosts, P_4 can yield the largest fitness value of 0.57895. Therefore, P_4 will be chosen as the destination host for V_1.

After the actual migration of V_1, the mechanism will enter its fourth step. As mentioned earlier, hosts which have been considered as critical in the first step (*i.e.* P_1) or have accepted VM(s) in the third step (*i.e.* P_4) will not be considered as under-utilized. Among P_2, P_3, and P_5, P_5 is regarded as the under-utilized host as it has the minimum utilization of 0.3. For V_{10}, the mechanism will select a destination host that can yield the largest fitness value from the available options (*i.e.* P_2 and P_3) which is better than P_5. If V_{10} can be migrated away without introducing new critical host(s), P_5 will be turned off. Otherwise, P_5 will remain active. The above process will repeat on the host having the next minimum utilization level.

From this example, it can be observed that the proposed mechanism tends to consolidate VMs onto some hosts rather than distributing them across all the hosts evenly. By doing so, some physical hosts can be turned off to save energy. Furthermore, the proposed mechanism tries to drive physical hosts to operate at desired utilization levels and loads them with VMs having different utilization patterns, which avoids triggering further overloading events.

VI. Experiments

To evaluate the efficiency of the proposed VM consolidation mechanism, a series of experiments were conducted in this section. The CloudSim [23] toolkit, which supports modeling of virtualized Cloud data center, was chosen as the experiment platform to implement the proposed mechanism.

A. Experiment Setup

In the experiments, 800 heterogeneous physical hosts were simulated. The simulated data center consists of two types of dual-core servers with equal volumes: HP ProLiant ML110 G4 (1860 MIPS, 4 GB) and HP ProLiant ML110 G5 (2660 MIPS, 4 GB). All hosts were equipped with 1GB storage and 1GB/s network bandwidth. These configuration settings limit the number of VMs on each host. For these physical hosts, their power models were obtained from SpecPower08 [34] correspondingly.

To simulate real world scenarios, four different types of single-core VMs were simulated in the experiments: High-CPU Medium Instance (2500 MIPS, 0.85 GB), Extra Large Instance (2000 MIPS, 1,7 GB), Small Instance (1000 MIPS, 1.7 GB), and Micro Instance (500 MIPS, 613 MB). All of these VMs were modeled to have 2.5 Gigabytes of VM size and 100 Mbit/s of bandwidth individually. In a simulated day, the VM consolidation processes were triggered every five simulated minutes.

B. Performance Metrics

1) Energy Consumption: In the current work, the total energy consumption consumed by all active hosts were measured. In Cloud clusters, high energy consumption will lead to high carbon dioxide emissions and high operational cost. Therefore, the amount of energy consumption is a key measurement to evaluate energy management efficiency. In the experiments, the total MIPS of each host, the allocated MIPS of each VM, and the time are used as input parameters to measure the total energy consumption. Furthermore, other performance indices are needed to give an all-round evaluation in other dimensions such as SLA violations and migration numbers.

2) SLA Violation Metrics: In Cloud clusters, CSPs should satisfy the expected QoS of their subscribers through the negotiated SLA. Here, the SLA, which is defined as a two-sided commitment, is a measurement to evaluate the level of QoS between a CSP and its subscribers. A typical SLA usually comprises several components such as the type of service provided, the desired performance level, rewards, and penalties. However, CSPs will have to pay penalties if the

negotiated SLA is violated, which will increase their operating costs. To measure the level of SLA violation, two metrics in [23] are adopted in the current work: (1) SLA violation Time per Active Host (SLATAH): SLATAH is a metric to measure the percentage of time when active hosts have been fully utilized. It can be calculated as

$$\text{SLATAH} = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{\text{s}_i}}{T_{\text{a}_i}}, \tag{8}$$

where N is the number of physical hosts; T_{s_i} is the total time during which host i has been fully utilized incurring on an violation of SLA; T_{a_i} is the total duration of host i being in the active state; and (2) Performance Degradation due to Migrations (PDM): PDM is a metric to measure the overall degradation of performance due to VM migrations. It can be computed as

$$\text{PDM} = \frac{1}{M}\sum_{j=1}^{M}\frac{C_{\text{d}_j}}{C_{\text{r}_j}}, \tag{9}$$

where M is the total number of VMs in the system; C_{d_j} is the estimated performance degradation of VM j due to VM migrations; C_{r_j} is the total CPU capacity required by VM j during its lifetime. Here, we assume that C_{d_j} equals 10% of the CPU utilization. SLATAH and PDM are equally important but independent to each other. These two metrics are then integrated into a parameter called SLA Violation (SLAV). It is defined as

$$\text{SLAV} = \text{SLATAH} \times \text{PDM}. \tag{10}$$

Here, SLAV measures degradation of performance caused by VM migrations and host overloading. In our experiments, the requested MIPS of each VM, the allocated MIPS of each VM, and the time are used as input parameters to measure the SLAV level.

3) Energy and SLA Violations Metrics: Energy consumption has a conflicted relationship with SLA violations. Energy consumption can usually be decreased at the expense of an increase of SLAV. Therefore, achieving a balanced trade-off between these two conflicting metrics is a major objective of the proposed mechanism. In the current work, we adopt a metric called Energy and SLA Violations (ESV) [23] to evaluate the overall performance of Cloud clusters. It is defined as

$$\text{ESV} = E \times \text{SLAV}, \tag{11}$$

where E is the total energy consumption of a data center. Here, energy consumption and SLAV metrics are combined together for ESV evaluation.

4) Migration Number: Live VM migration is a costly process. During migration, the migrated VMs will occupy some CPU time and network bandwidth on both source and destination hosts. Additionally, such migration may adversely affect the performance of application running on the migrating VM. Therefore, a small migration number is more desirable.

C. Benchmarking Mechanisms

In the experiments, six existing power-based or correlation-based mechanisms were selected for comparison purposes, including two power-based methods in [23] and [24], the three consolidation mechanisms adopting different correlation-based criteria in [21], and the heuristics-based method in [22]. In the experiments, the power-based Local Regression Robust-Minimum Migration Time (LRR-MMT) mechanism in [23] was selected as a referencing benchmark because such mechanism outperforms other existing power-based methods. Here, LRR method, which estimates host utilization level based on its historical utilization values, is an adaptive-threshold method for overloading detection. In the power-based LRR-MMT mechanism, if the estimated CPU utilization of a host is larger than the static utilization threshold, the VM with minimum migration time on such host is chosen for migration. Here, in the power-based LRR-MMT mechanism, the power consumption of a host is adopted as a migration criterion. The host with the least increase in power consumption after receiving the migrated VM will be selected as the primary destination candidate for migration. In the power-based THR-MUG method [24], an upper fixed utilization threshold is set for host overloading detection. On an overloaded host, VM(s) with minimum utilization gap will be selected for migration. The three consolidation mechanisms [21] adopt the correlation of migrated VM(s), the average correlation level (ACL), and the variation of correlation level (VCL) as their criteria separately. In the heuristics-based method [22], a single heuristic h_{T}, which comprises two heuristics h_1 and h_2, is regarded as the migration criterion.

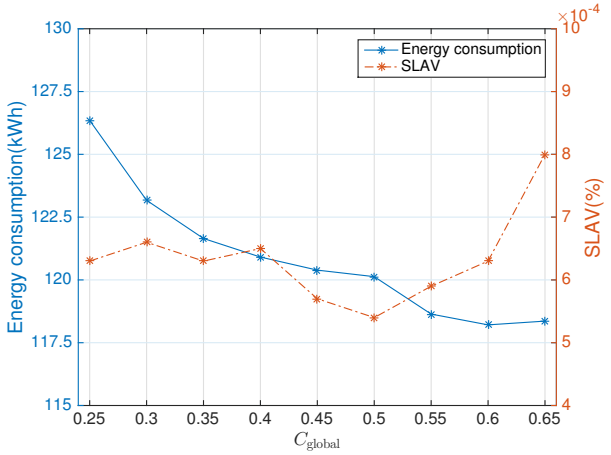
VII. Results and Discussions

In this section, a series of experiments using real-world workload data were carried out and the corresponding results are presented. The workload data is provided by the PlanetLab project [35]. Such project collects CPU usage data from thousands of VMs every five minutes. In the experiments, workload traces randomly chosen from 10 days of the provided data were used.

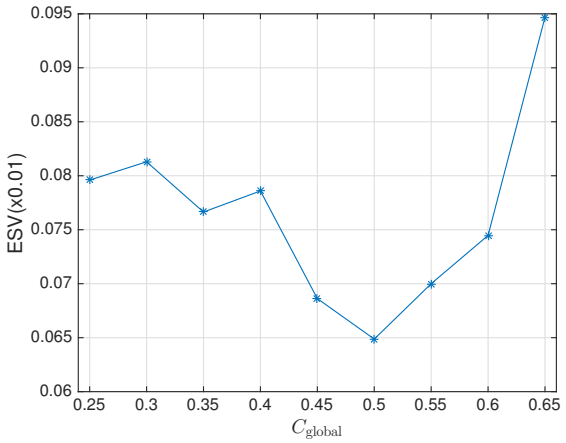
A. Effects of Global Tuning Parameter to the Proposed Mechanism

As mentioned in Section IV, a global tuning parameter C_{global} is required for VM consolidation. An experiment on examining the performance of using different global tuning parameters was carried out using the real-world workload on 03 March, and the results are shown in Fig. 5.

From Fig. 5(b), it can be observed that having $C_{\text{global}} = 0.5$ can yield a minimum level of ESV among the experiment setups under test, which indicates a balanced trade-off between energy and SLAV. Therefore, the same value was used for the remaining experiments. In addition, it is observed that a higher C_{global} value, which refers to a higher threshold for each VM, would allow some hosts to operate at higher utilization levels with more co-located VMs, thus more hosts can be switched off. As a result, the system requires lower total energy consumption. Meanwhile, higher utilization on some hosts implies a higher chance of overloading, which incurs more SLA violations.



(a) Performance metrics versus C_{global}



(b) ESV versus C_{global}

Fig. 5: Results of the proposed mechanism under different global tuning parameters.

The proposed mechanism is executed every five minutes. It can be estimated that the energy consumption decreases with the increase of the global tuning parameter, which concurs with our results. However, having the value of C_{global} being too low, which refers to a small capacity threshold for each VM, triggers over-provisioning and introduces more under-utilized hosts to the system. On the contrary, an extremely high value of C_{global} is also not desirable as it is more likely to trigger overloading incidents. Therefore, the value of C_{global} is suggested to be selected within $[0.25, 0.65]$, which allows VMs with moderate RUC to be co-located and yield a better utilization.

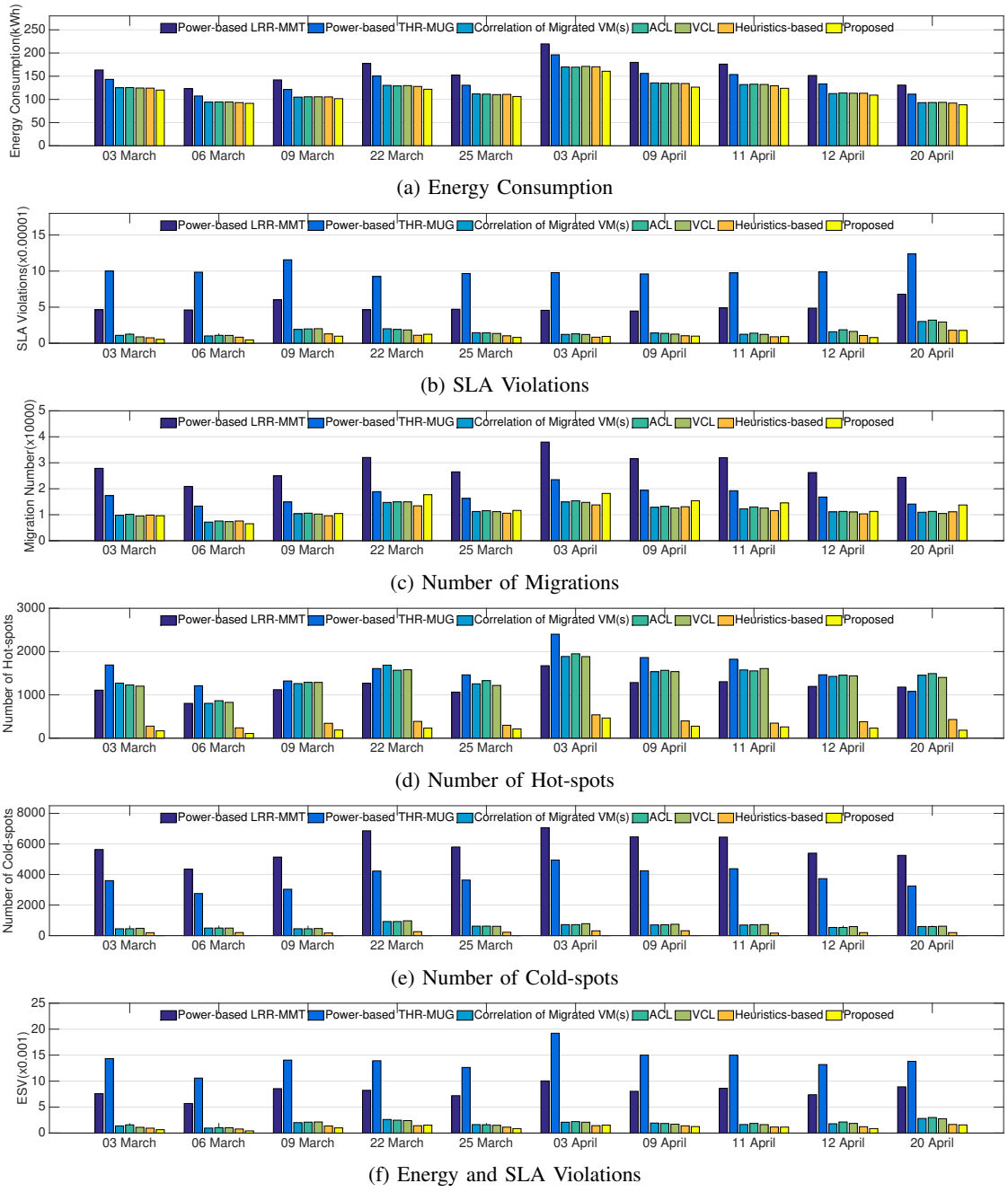
B. Real-World Workload

Fig. 6 shows the experiment results under different consolidation mechanisms. Additionally, the average daily results over 10 simulated days are shown in Table I. The total energy consumption of different VM consolidation mechanisms under test are reported in Fig. 6(a). As observed in Table I, the proposed mechanism can reduce the overall energy consumption by about 4%-29% when comparing with the other methods under test. Fig. 6(b) compares SLAV of Cloud clusters under different VM consolidation mechanisms. The average SLAV of the proposed mechanism is 11%-91% lower than those of other six benchmarking mechanisms. This demonstrates the effectiveness of the proposed VM consolidation mechanism in overload avoidance. VM migrations may trigger violations of SLA, hence it is essential to minimize the migration number whenever possible. As shown in Fig. 6(c), a fewer number of migrations were invoked by the proposed mechanism compared to the power-based LRR-MMT and THR-MUG methods. In addition, the number of hot-spots and cold-spots are reported for comparison. Here, hosts with CPU utilization above 90% or below 25% are considered as hot-spots or cold-spots, respectively. There are two input parameters, the total MIPS of each host and the allocated MIPS of each VM, being used to measure the number of hot-spots and cold-spots. In Fig. 6(d), hot-spots are significantly relieved using the proposed mechanism by prohibiting over-commitment. To a certain degree, the number of cold-spots represents the extent of resource waste. Fig. 6(e) shows that the proposed mechanism mitigates a considerable number of cold-spots to avoid resource waste. Furthermore, a compromise between energy consumption and QoS can be demonstrated by the ESV metric. Systems which are more capable of achieving energy saving and a higher level of QoS, are normally with lower ESV values. The results of ESV in Fig. 6(f) show that the proposed mechanism outperforms other existing mechanisms under test in most cases, which indicates the ability of the proposed mechanism in delivering a better overall performance in Cloud computing environments.

When comparing the number of active hosts under different consolidation mechanisms, it is observed that the amount of active hosts in systems with the proposed mechanism is smaller than those with other six benchmarking mechanisms. Within the active hosts, the number of cold-spots utilized by the proposed mechanism is much lower than other mechanisms under test. This explains the promising energy saving performance of the proposed mechanism as it tends to consolidate VMs onto fewer physical hosts by considering the host utilization levels. In addition, incorporating with the RUC among co-located VMs in the VM consolidation process, the risk of overloading can be lowered. Hence, the proposed mechanism enables better consolidations of VMs with less violations of SLA and hot-spots.

VIII. Conclusion

In this work, a VM consolidation mechanism based on bio-inspired heuristics is proposed. Heuristic functions in the proposed mechanism, which incorporate both the host utilization levels and resource utilization correlations among co-located VMs, are inspired by host-switching behaviors in symbiotic organisms. Under the proposed mechanism, a larger capacity margin and a higher fitness value indicate a more desirable operating environment for VMs and hosts, respectively. The proposed mechanism is implemented and evaluated on CloudSim. The average ESV by the proposed mechanism is 14%-92% lower than that of other benchmarking mechanisms



(a) Energy Consumption

(b) SLA Violations

(c) Number of Migrations

(d) Number of Hot-spots

(e) Number of Cold-spots

(f) Energy and SLA Violations

Fig. 6: Comparison results of the proposed mechanism with other existing mechanisms.

over ten simulated days. Experiment results demonstrate that the proposed mechanism can lower the risk of overloading, reduce SLA violations, and minimize the energy consumption in comparisons with other existing mechanisms under test.

In the future, we plan to extend bio-inspired heuristics to multi-dimensional resources. In addition, we plan to consider the network topology of the Cloud data center in the VM consolidation process to achieve better application performance.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant No. 51777146 and the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University under Project RTMR.

REFERENCES

[1] M. M. Hassan, M. Abdullah-Al-Wadud, A. Almogren, B. Song, and A. Alamri, "Energy-aware resource and revenue management in federated cloud: a game-theoretic approach," *IEEE Systems Journal*, vol. 11, no. 2, pp. 951–961, 2017.

[2] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner, "United States Data Center Energy Usage Report," 2016.

[3] M. Zhang, R. Ranjan, M. Menzel, S. Nepal, P. Strazdins, W. Jie, and L. Wang, "An infrastructure service recommendation system for cloud applications with real-timeqosrequirement constraints," *IEEE Systems Journal*, 2015.

TABLE I: Average daily results over 10 simulated days

VM allocation mechanisms	Energy consumption	Migration number	Hot-spots	Cold-spots	SLAV (x0.00001)	ESV (x0.001)
Power-based LRR-MMT [23]	161.71 kWh	28438	1201	5841	5.02	8.03
Power-based THR-MUG [24]	140.41 kWh	17384	1592	3780	10.18	14.17
Correlation of migrated VM(s) [21]	120.99 kWh	11549	1411	618	1.6	1.89
ACL [21]	121.07 kWh	11899	1426	620	1.69	1.98
VCL [21]	121.00 kWh	11477	1391	647	1.55	1.82
Heuristics-based [22]	120.11 kWh	**11098**	712	448	1.07	1.25
Proposed	**115.03 kWh**	12919	**233**	**3**	**0.95**	**1.08**

[4] Y. Ruan, Z. Cao, and Z. Cui, "Pre-filter-copy: efficient and self-adaptive live migration of virtual machines," *IEEE Systems Journal*, vol. 10, no. 4, pp. 1459–1469, 2016.

[5] X. Fu and C. Zhou, "Predicted affinity based virtual machine placement in cloud computing environments," *IEEE Transactions on Cloud Computing*, no. 1, pp. 1–1, 2017.

[6] E. P. Hoberg and D. R. Brooks, "A macroevolutionary mosaic: episodic host-switching, geographical colonization and diversification in complex host–parasite systems," *Journal of Biogeography*, vol. 35, no. 9, pp. 1533–1550, 2008.

[7] S. B. Araujo, M. P. Braga, D. R. Brooks, S. J. Agosta, E. P. Hoberg, F. W. von Hartenthal, and W. A. Boeger, "Understanding host-switching by ecological fitting," *PLoS One*, vol. 10, no. 10, p. e0139225, 2015.

[8] H. L. Schreiber, M. S. Conover, W.-C. Chou, M. E. Hibbing, A. L. Manson, K. W. Dodson, T. J. Hannan, P. L. Roberts, A. E. Stapleton, T. M. Hooton *et al.*, "Bacterial virulence phenotypes of escherichia coli and host susceptibility determine risk for urinary tract infections," *Science Translational Medicine*, vol. 9, no. 382, p. eaaf1283, 2017.

[9] V. I. Yukalov, E. Yukalova, and D. Sornette, "Modeling symbiosis by interactions through species carrying capacities," *Physica D: Nonlinear Phenomena*, vol. 241, no. 15, pp. 1270–1289, 2012.

[10] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, and H. Tenhunen, "Using ant colony system to consolidate VMs for green cloud computing," *IEEE Transactions on Services Computing*, vol. 8, no. 2, pp. 187–198, Mar. 2015.

[11] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Transactions on Evolutionary Computation*, 2016.

[12] M. Abdullahi, M. A. Ngadi *et al.*, "Symbiotic organism search optimization based task scheduling in cloud computing environment," *Future Generation Computer Systems*, vol. 56, pp. 640–650, 2016.

[13] Y. Wen, Z. Li, S. Jin, C. Lin *et al.*, "Energy-efficient virtual resource dynamic integration method in cloud computing," *IEEE Access*, 2017.

[14] F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, and V. C. Leung, "Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1688–1699, 2018.

[15] L.-D. Chou, H.-F. Chen, F.-H. Tseng, H.-C. Chao, and Y.-J. Chang, "DPRA: dynamic power-saving resource allocation for cloud data center using particle swarm optimization," *IEEE Systems Journal*, 2016.

[16] J. Kim, M. Ruggiero, D. Atienza, and M. Lederberger, "Correlation-aware virtual machine allocation for energy-efficient datacenters," in *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, 2013, pp. 1345–1350.

[17] A. Pahlevan, P. G. Del Valle, and D. Atienza, "Exploiting cpu-load and data correlations in multi-objective VM placement for geo-distributed data centers," in *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2016, pp. 1333–1338.

[18] W. Wei, X. Wei, T. Chen, X. Gao, and G. Chen, "Dynamic correlative VM placement for quality-assured cloud service," in *2013 IEEE International Conference on Communications (ICC)*, 2013, pp. 2573–2577.

[19] R. Chi, Z. Qian, and S. Lu, "Be a good neighbour: Characterizing performance interference of virtual machines under xen virtualization environments," in *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, 2014, pp. 257–264.

[20] Q. Zhu and T. Tung, "A performance interference model for managing consolidated workloads in QoS-aware clouds," in *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, 2012, pp. 170–179.

[21] J. V. Wang, C.-T. Cheng, and C. K. Tse, "Effects of correlation-based VM allocation criteria to cloud data centers," in *2016 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2016, pp. 398–401.

[22] J. V. Wang, N. Ganganath, C. T. Cheng, and C. K. Tse, "A heuristics-based VM allocation mechanism for cloud data centers," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2017, pp. 1–4.

[23] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2012.

[24] X. Wu, Y. Zeng, and G. Lin, "An energy efficient vm migration algorithm in data centers," in *2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES)*. IEEE, 2017, pp. 27–30.

[25] J. Lhotsky, "How important is living together? three facets of symbiosis concept," *Journal of Interdisciplinary Studies*, 2011.

[26] J. Chaston and H. Goodrich-Blair, "Common trends in mutualism revealed by model associations between invertebrates and bacteria," *FEMS Microbiology Reviews*, vol. 34, no. 1, pp. 41–58, 2010.

[27] B. D. Martin and E. Schwab, "Current usage of symbiosis and associated terminology," *International Journal of Biology*, vol. 5, no. 1, p. 32, 2012.

[28] R. Goto, A. Kawakita, H. Ishikawa, Y. Hamamura, and M. Kato, "Molecular phylogeny of the bivalve superfamily galeommatoidea (heterodonta, veneroida) reveals dynamic evolution of symbiotic lifestyle and interphylum host switching," *BMC Evolutionary Biology*, vol. 12, no. 1, p. 172, 2012.

[29] I. Horká, S. De Grave, C. H. Fransen, A. Petrusek, and Z. Ďuriš, "Multiple host switching events shape the evolution of symbiotic palaemonid shrimps (crustacea: Decapoda)," *Scientific Reports*, vol. 6, p. 26486, 2016.

[30] H. Abdi, "Multiple correlation coefficient," *The University of Texas at Dallas*, 2007.

[31] "Principles of epidemiology and microbiology," http://nursing411.org, 2016.

[32] N. R. Council *et al.*, *Critical aspects of EPA's IRIS assessment of inorganic arsenic: interim report*. National Academies Press, 2013.

[33] Y. Sun, J. White, B. Li, M. Walker, and H. Turner, "Automated qos-oriented cloud resource optimization using containers," *Automated Software Engineering*, vol. 24, no. 1, pp. 101–137, 2017.

[34] "Specpower08," http://www.spec.org, 2011.

[35] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, "Planetlab: an overlay testbed for broad-coverage services," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 3, pp. 3–12, 2003.

Jing V. Wang received the B.Eng. degree in electrical engineering and automation from Wuhan University of Technology, Wuhan, China in 2013, and the Ph.D. degree in electronic and information engineering from the Hong Kong Polytechnic University, Hong Kong in 2018. Her current research interests include Cloud Computing and prognostics and health management of lithium-ion batteries.

Nuwan Ganganath (S'09-M'17) received the B.Sc. (Hons) degree with first class honors in electronics and telecommunication engineering from the University of Moratuwa, Sri Lanka, in 2010, the M.Sc. degree in electrical engineering from the University of Calgary, Canada in 2013, and the Ph.D. degree in electronic and information engineering from the Hong Kong Polytechnic University, Hong Kong in 2016. He is currently an Adjunct Research Fellow at the School of Electrical, Electronic and Computer Engineering at the University of Western Australia, Australia. He received the Prize of the President of the International Physics Olympiads (IPhOs) at the 36th IPhO competition in Salamanca, Spain in 2005. He was a recipient of the Mahapola Merit Scholarship from the Sri Lankan Government in 2006, the Hong Kong Ph.D. Fellowship from the Hong Kong Research Grants Council in 2013, and the Endeavour Research Fellowship from the Australia Government in 2016.

Chi-Tsun Cheng (S'07-M'09) received the B.Eng. and M.Sc. degrees from the University of Hong Kong in 2004 and 2005, respectively, and the Ph.D. degree from the Hong Kong Polytechnic University in 2009. He was a recipient of the Sir Edward Youde Memorial Fellowship in 2009 during his Ph.D. studies. From 2010 to 2011, he was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, the University of Calgary. From 2012 to 2018, he was a Research Assistant Professor in the Department of Electronic and Information Engineering, the Hong Kong Polytechnic University. Since June 2018, he has been a Senior Lecturer in the Department of Manufacturing, Materials and Mechatronics, RMIT University, Melbourne, Australia.

Dr. Cheng serves as Associate Editors for the *IEEE Transactions on Circuits and Systems II* and *IEICE Nonlinear Theory and Its Applications*. He served as the Chairperson of the International Workshop on Smart Sensor Networks (IWSSN) from 2012 to 2018. He is members of the IEEE Nonlinear Circuits and Systems Technical Committee and IEEE Power and Energy Circuits and Systems Technical Committee. He served as the track co-chair for the IEEE International Symposium on Circuits and Systems (ISCAS 2018). He is a Youth Startup Sub-com Member (Global Relations Research) of the Hong Kong Electronics and Technologies Association. His research interests include Wireless Sensor Networks, Internet of Things, Industry 4.0 Technologies, Cloud Computing, and Additive Manufacturing.

Chi K. Tse (M'90–SM'97–F'06) received the BEng (Hons) degree with first class honors in electrical engineering and the PhD degree from the University of Melbourne, Australia, in 1987 and 1991, respectively.

He is presently Chair Professor at the Hong Kong Polytechnic University, Hong Kong, with which he served as Head of the Department of Electronic and Information Engineering from 2005 to 2012. His research interests include power electronics, nonlinear systems and complex network applications. He was awarded a number of research and industry awards, including Prize Paper Awards by IEEE TRANSACTIONS ON POWER ELECTRONICS in 2001, 2015 and 2017, RISP Journal of Signal Processing Best Paper Award in 2014, Best paper Award by International Journal of Circuit Theory and Applications in 2003, two Gold Medals at the International Inventions Exhibition in Geneva in 2009 and 2013, a Silver Medal at the International Invention Innovation Competition in Canada in 2016, and a number of recognitions by the academic and research communities, including honorary professorship by several Chinese and Australian universities, Chang Jiang Scholar Chair Professorship, IEEE Distinguished Lectureship, Distinguished Research Fellowship by the University of Calgary, Gledden Fellowship and International Distinguished Professorship-at-Large by the University of Western Australia. While with the Hong Kong Polytechnic University, he received the President's Award for Outstanding Research Performance twice, Faculty Research Grant Achievement Award twice, Faculty Best Researcher Award, and several teaching awards.

Dr. Tse serves and has served as Editor-in-Chief for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II (2016-2019), *IEEE Circuits and Systems Magazine* (2012-2015), Editor-in-Chief of *IEEE Circuits and Systems Society Newsletter* (since 2007), Associate Editor for three IEEE Journal/Transactions, Editor for *International Journal of Circuit Theory and Applications*, and is on the editorial boards of a few other journals. He currently chairs the steering committee for the IEEE Transactions on Network Science and Engineering. He also serves as panel member of Hong Kong Research Grants Council, and member of several professional and government committees.