

Received August 10, 2020, accepted August 20, 2020, date of publication August 28, 2020, date of current version September 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020043

# Evidence-Efficient Affinity Propagation Scheme for Virtual Machine Placement in Data Center

ZHIHUA LI<sup>1</sup>, SHUJIE GUO<sup>1</sup>, LEI YU<sup>2</sup>, AND VICTOR CHANG<sup>3</sup>

<sup>1</sup>Department of Computer Science and Technology, School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi 214122, China

<sup>2</sup>IBM Research, Yorktown Heights, NY 10598, USA

<sup>3</sup>School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough TS1 3BX, U.K.

Corresponding author: Zhihua Li (zhli@jiangnan.edu.cn)

This work was supported in part by the Smart Manufacturing New Model Application Project Ministry of Industry and Information Technology under Grant ZH-XZ-18004, in part by the Future Research Projects Funds for the Science and Technology Department of Jiangsu Province under Grant BY2013015-23, in part by the Fundamental Research Funds for the Ministry of Education under Grant JUSRP211A 41, in part by the Fundamental Research Funds for the Central Universities under Grant JUSRP42003, in part by the 111 Project under Grant B2018, and in part by the VC Research under Grant VCR 0000071.

**ABSTRACT** In cloud data center, without efficient virtual machine placement, the overload of any types of resources on physical machines (PM) can easily cause the waste of other types of resources, and frequent costly virtual machine (VM) migration, which further negatively affects quality of service (QoS). To address this problem, in this paper we propose an evidence-efficient affinity propagation scheme for VM placement (EEAP-VMP), which is capable of balancing the workload across various types of resources on the running PMs. Our approach models the problem of searching the desirable destination hosts for the live VM migration as the propagation of responsibility and availability. The sum of responsibility and availability represent the accumulated evidence for the selection of candidate destination hosts for the VMs to be migrated. Further, in combination with the presented selection criteria for destination hosts. Extensive experiments are conducted to compare our EEAP-VMP method with the previous VM placement methods. The experimental results demonstrate that the EEAP-VMP method is highly effective on reducing VM migrations and energy consumption of data centers and in balancing the workload of PMs.

**INDEX TERMS** VM placement, affinity propagation, compatibility matrix, responsibility, availability.

## I. INTRODUCTION

Cloud data centers are often characterized by high energy consumption when providing stable and reliable cloud services for cloud users. The large amount of running physical machines (PMs) but with low resource utilization is considered the primary reasons for the high energy consumption of cloud data centers [1]. Virtual machine (VM) consolidation is a basic technology for virtual resource scheduling and management at the LaaS layer of a cloud data center. As a key aspect of VM consolidation, VM placement alters the deployment relationship between VMs and PMs, by placing VMs appropriate destination hosts. It reallocates various types of virtual resources in the cloud data center. Efficient VM placement not only provides high-quality and stable services for cloud rental users, but also reduce energy consumption, improve resource utilization, which helps to guarantee quality of service (QoS) of cloud data centers.

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

However, computation tasks submitted by users involve uncertain demands for resources during VM placement. Although virtual resources can be reallocated through VM consolidation, PM resources are typically allocated unevenly and may not have sufficient reservation for future demands, resulting in the resource overloading of PMs. If a PM is overloaded, all VMs deployed on the PM will compete for resources and this can easily result in QoS degradation in the data center.

In recent years, VM placement has received extensive attention [2]–[10]. Researchers have studied various VM placement models and algorithms to optimize the mapping between VMs and PMs to obtain a higher resource utilization of PMs, better load balance among all running hosts, and lower energy consumption of the data center. In [2]–[4], VM placement is regarded as a bin packing problem. Algorithms such as first fit decreasing (FFD) [2] and power-aware best-fit decreasing (PABFD) [4] were proposed for VM placement. However, since their solutions can easily fall into local optimization, these algorithms negatively affect QoS

and energy utilization of data centers. In [5]–[10], VM placement is modeled as a combinatorial optimization problem, and intelligent heuristic algorithms were proposed used to the model. These algorithms can obtain the global optimization solution, but it involves high time complexity and large time overhead. In [11]–[13], the relationship between various types of resources and workloads in the data center is examined, and various factors of the data center that impact VM placement selected are statistically analyzed. These studies have yielded VM placement methods with good performance. However, in a cloud data center, there is usually a situation where a single type of resource of certain running physical machine is prone to overload prematurely and too quickly. This phenomenon is likely to cause waste of other types of resources, trigger unnecessary VM migration, and further degrade QoS. Therefore, it is critical to maintain the load balance of various resources at the destination hosts during the period of VM placement but challenging due to uncertain and dynamical demands.

In this paper, we propose an evidence-efficient affinity propagation scheme for VM placement (EEAP-VMP). By modeling the hosts and VMs resource such as CPU, memory, and bandwidth in the form of resource vectors, we proposed to use the similarity of resource vectors to effectively measure the compatibility between the requested resources from VMs and the remaining resources of the PMs. Considering the property of resource attributes, we leverage the Mahalanobis distance [14] to measure the similarity between resource vectors. Inspired by the idea of affinity propagation [15], we then abstract each PM and VM as sample such that all PMs and VMs in the cloud data center yield a sample space. By defining the responsibility and availability among different samples, EEAP-VMP performs the affinity propagation of the accumulated evidence of responsibility and availability between PMs and VMs in the cloud data center. Such an affinity propagation algorithm continues until the destination hosts are found or a maximum of iterations is achieved. Here, a clustering algorithm is used to identify the cluster centers in the sample space that consists of VMs and PMs. The candidate destination PMs are equivalent to cluster centers found by the algorithm. Then, EEAP-VMP used the arithmetic sum of responsibility and availability as the accumulated evidence to decide the destination hosts for the VMs to be migrated. Combining the accumulative evidence with the PM energy consumption model, we propose a criterion to select final destination hosts to place migrating VMs.

In summary, our contributions in this paper are listed as follows:

(1) By abstracting remaining resources of the PMs and the requested resources from VMs into resource vectors, we define the compatibility between the resources of VMs and PMs, which build the basis for effectively avoiding the premature overloading of a certain type of resource, thereby balancing the workload across different types of resources.

(2) We propose an affinity propagation algorithm to decide candidate destination hosts for VM placement, with the

cumulative evidence for selecting destination hosts of the migrated VMs. In addition, an energy consumption calculation model is integrated to define the criterion for selecting a destination host for VM placement.

(3) Extensive simulation experiments are conducted to evaluate EEAP-VMP. The experimental results show that EEAP-VMP significantly improves QoS, reduces the numbers of VM migrations and the amount of running PMs, and decreases energy consumption of cloud data centers.

## II. RELATED WORK

The problem of VM placement is to allocate several migrated VMs to the appropriate destination hosts, and the total number of running PMs should be as small as possible, which is considered as a NP-hard problem. The problem is often studied with the goal of keeping energy consumption of data centers keeps at a low level. Heuristic algorithms and statistics-based algorithms have been proposed to optimize the issue of VMs placement.

### A. HEURISTIC ALGORITHMS-BASED VM PLACEMENT METHODS

The heuristic algorithm-based VM placement methods usually employ the greedy selection scheme. With respect to the greedy selection destinations, the greedy selection scheme is capable of quickly finding the destination hosts for the VMs to be migrated. However, the greedy selection scheme cannot guarantee to find a global optimal VM placement solution.

Alahmadi *et al.* [2] presented the first fit decreasing (FFD) algorithm to select the first destination hosts that fits VM requirements during the search. However, it does not consider the resource utilization and possible SLA violations. Farahnakian *et al.* [3] developed a utilization prediction best fit decreasing (UPBFD) method based on resource utilization prediction. UPBFD ranks the resource utilization of the running PMs in the descending order and performs VM placement to the PMs with a high resource utilization during VM consolidation. UPBFD algorithm effectively improves the resources utilization and reduces the energy consumption of data centers. However, UPBFD only aims to optimize the energy consumption of data centers, and easily falls to local optimal. Beloglazov *et al.* [4] presented a power aware best fit decreasing (PABFD) algorithm. PABFD first arranges the unassigned VMs in terms of their CPU resource requirements in descending order, and then allocates each VM in this order to the destination hosts, which aims at minimizing the increase of energy consumption after VM placement. In addition, since the intelligence heuristic algorithms [4], [10] can efficiently search the global optimization solution, they are proposed to solve the problem of VM placement.

Alharbi *et al.* [5] models the dynamic VM placement as a constraint combination optimization problem based on the configuration information of physical resource of VMs and PMs. Its optimization objective is to minimize total energy consumption of running PMs. To obtain the optimal solution, the authors utilized Ant Colony System (ACS)

to solve the optimization problem, and compared it with that of FFD algorithm [6] and ACS-based VM placement algorithm [7]. Li *et al.* [8] considered the optimization of energy consumption with multi-resource constraints for VM placement and proposed differential evolution based method to solve it, named as Discrete-differential-evolution-based VM Placement (DDE-VMP). It is shown that DDE-VMP achieves better performance in reducing energy consumption of data centers and improving QoS than previous works. Li *et al.* [9] considered VM placement with the optimization objectives minimizing energy consumption, the amount of VM migrations, and the overloading probability of PMs, and proposed a multi-objective optimization model. The method they proposed is an intelligent heuristic algorithm which uses a pheromone matrix to retain the accumulated experience during the iterative search process for optimizing the mapping relationship between the running PMs and VMs. Although the intelligent heuristic algorithm usually obtains a better solution for the optimization model, the algorithm takes a large time cost and exist a risk of falling inside local optimization.

### B. STATISTICS-BASED VM PLACEMENT

By establishing a Bayesian network model for dynamic probability estimation of live VM migration, a Bayesian-network-model-based VM Placement (BN-VMP) method is proposed [11]. The model treats VMs and PMs in data centers as nodes of Bayesian network, and evaluates the probability of live VM migration for destination hosts based on the resource requirements and workload state of each node. Melhem *et al.* [12] analyzed the resource workload of PMs in data centers, and proposed a Markov-prediction-model-based power aware best fit decreasing (MPABFD) VM placement algorithm. According to the characteristics of various resource in data centers, MPABFD uses a first-order Markov chain model to establish a prediction model to predict future workload status of the running PMs and perform VM placement accordingly. Li *et al.* [13] developed a strategic game-based VM placement (SG-VMP) algorithm, which employs the First Fit Decreasing (FFD) algorithm to resolve the corresponding VM placement optimization model for each item of game strategies. The algorithm obtains a Nash equilibrium solution in terms of the payment function values for the different VM placement possibilities.

## III. CANDIDATE DESTINATION HOST SELECTION ALGORITHM

### A. ESTIMATION OF REMAINING RESOURCES OF PMs

In a cloud data center, the computing entities have different types of resource including CPU, memory, and bandwidth constitute. Assuming that a cloud data center has  $m$  PMs, we denote these PMs by set  $PM = \{pm_1, pm_2, \dots, pm_i, \dots, pm_m\}$ . Let  $C_{pm}^i$  be the resource vector configured for each physical machine  $pm_i$ , represented as  $C_{pm}^i = (c_{pm_m}^{cpu}, c_{pm_m}^{mem}, c_{pm_m}^{bw})^T$ . Suppose that at a certain time,

the data center has  $n$  migrating VMs, for which destination hosts are to be selected for VM placement. We express these VMs using set  $VM = \{vm_1, vm_2, \dots, vm_j, \dots, vm_n\}$ . The cloud users submit their tasks to the cloud data center. The data center allocates the tasks through a task scheduler to the VMs for processing. Due to uncertainty of user demands in the tasks, the CPU, memory, and bandwidth resources consumed by VMs when processing user tasks can vary with time. Therefore, the migrating VM, i.e.,  $vm_j \in VM$  at any time, can be represented by its current resource demand vector,  $D_{vm}^j$ , i.e.,  $D_{vm}^j = (d_{vm_j}^{cpu}, d_{vm_j}^{mem}, d_{vm_j}^{bw})^T$ . Subsequently, the remaining resources of physical machine  $pm_i$  can be expressed as shown in formula (1).

$$R_{pm}^i = C_{pm}^i - \sum_{vm_j \in pm_i} D_{vm}^j \quad (1)$$

where  $vm_j \in pm_i$  represents the VMs that have been deployed on  $pm_i$ . Furthermore,  $vm_j$  deployed on  $pm_i$  should satisfy the constraints shown in formula (2).

$$\begin{cases} \sum_{vm_j \in pm_i} d_{vm}^{cpu} \leq C_{pm_i}^{cpu} \\ \sum_{vm_j \in pm_i} d_{vm}^{mem} \leq C_{pm_i}^{mem} \\ \sum_{vm_j \in pm_i} d_{vm}^{bw} \leq C_{pm_i}^{bw} \end{cases} \quad (2)$$

wherein, the equation (2) means that the total amount of a certain type of the requested resources by VMs deployed on the host  $pm_i$  cannot be greater than the total resource capacity of the host.

### B. RESOURCE COMPATIBILITY CALCULATION

As for the VM placement, the resource vectors of VMs and PMs have different resource type in different dimensions, and they are closely related to each other. The data sample sets released by major cloud service providers indicate that the numerical differences between different resource types are significant. Therefore, considering the differences between the attributes of different resource types of VMs and PMs, we used Mahalanobis distance [14] to calculate the similarity and dissimilarity between resource vectors to reduce the negative effect of different dimensions on the compatibility between samples.

For the convenience of analysis, we abstracted a cloud data center into a sample space. The resource vectors of PMs and VMs form a sample set,  $Z = \{Z_{pm}^m, Z_{vm}^n\}$ , where  $Z_{pm}$  represents a subset of the resource vectors of PMs,  $C_{pm}^i \in Z_{pm}$ , and  $C_{pm}^i = (c_{pm_m}^{cpu}, c_{pm_m}^{mem}, c_{pm_m}^{bw})^T$ ; and  $Z_{vm}$  represents a subset of the resource vectors of VMs,  $D_{vm}^j \in Z_{vm}$ , and  $D_{vm}^j = (d_{vm_j}^{cpu}, d_{vm_j}^{mem}, d_{vm_j}^{bw})^T$ .

Furthermore, the similarity between samples in the resource vector space can be measured by formula (3).

$$f(i, j) = \sqrt{(z_i - z_j)^T \sum^{-1} (z_i - z_j)} \quad (3)$$

where  $z_i \in Z$ ,  $i \in [0, n - 1]$ , and  $j \in [n, n + m - 1]$ .  $\sum$  represents the covariance of the resource attributes of VMs and PMs in each dimension, which is calculated as shown in formula (4).

$$\sum z^r \cdot \bar{z}^r = \frac{1}{n + m - 1} (z^r - \bar{z}^r)^T \times (z^r - \bar{z}^r)^T \quad (4)$$

where  $r \in \{cpu, mem, dw\}$ ;  $z^r$  represents the  $r$  type of resource in set  $Z$ ;  $\bar{z}^r$  represents the mean value of the  $r$  type of resource in set  $Z$ .

The dissimilarity between the samples in the resource vector space is calculated as formula (5).

$$dis\_f(i, j) = 1 / \left( 1 - \frac{f(i, j) - f_{\min}}{f_{\max} - f_{\min}} \right) \quad (5)$$

where  $f_{\min}$  and  $f_{\max}$  denote the minimum and maximum values of sample similarity in the resource vector space, respectively.

**Definition 1:** Resource compatibility is used to indicate the compatibility between a migrating VM with a destination host for VM placement, i.e., the degree to which the requested resources by the migrating VM are compatible with the remaining resources of the running PM.

The resource compatibility degree is determined by formula (6),

$$c(i, j) = \frac{f \times dis\_f}{f + dis\_f} \quad (6)$$

where,  $f$  is given by equation (3), and  $dis\_f$  is calculated by formula (5).

### C. ESTIMATION OF PM ENERGY CONSUMPTION

The main purpose of VM placement is to allocate all types of resources in the data center appropriately with optimizing the deployment relation between VMs and PMs. It can efficiently reduce energy consumption and improve QoS. Therefore, the index of energy consumption of a data center is one of the most effective measures for validating a VM placement scheme. Reference [18] provided the power parameters of different models of PMs under different CPU utilizations, based on which the energy consumption of PMs can be approximated to estimate for a data center. Based on the method [18], Li *et al.* [9] developed a method to estimate the energy consumption of host based on the linear relationship between the CPU utilization and energy consumption of hosts, as shown in formula (7),

$$EC(host_j) = \int_{t_0}^{t_1} P_{run}(u_{j,cpu}(t)) dt \quad (7)$$

where  $P_{run}(u_{j,cpu}(t))$  represents the energy consumption of physical machine  $host_j$  when the CPU utilization of a host is  $u_{j,cpu}(t)$  at time  $t$ . The above mentioned method can estimate the energy consumption more accurately than the linear energy consumption model [21].

### D. COMPATIBILITY MATRIX GENERATION ALGORITHM

Using Eqs. (3) and (5) in part B of Section III, we can obtain the dissimilarity or similarity between any two samples in the resource vector sample space  $Z$ , i.e., the relationship between different PMs, between different VMs, and between PMs and VMs. Typically, the distance is a negative value. Therefore, the less the distance, the higher the similarity between VMs and PMs and their compatibility are. In a cloud data center, the compatibility between  $n$  VMs and  $m$  PMs is represented by a compatibility matrix  $C$ , which is then used by a clustering algorithm to find cluster centers in the sample space of resource vectors of VMs and PMs. Let the compatibility matrix  $C$  be a symmetric matrix in  $(n + m) \times (n + m)$  dimensions.  $c(i, i)$  is the diagonal element of the compatibility matrix, which is also known as the preference  $p$ . When the value of  $c(i, i)$  is larger, the probability that sample  $i$  becomes a cluster center is greater. Typically, each  $c(i, i)$  is a priori designated as the mean of the similarity values in the compatibility matrix  $C$ . Therefore, all samples in the sample space are equally likely to become cluster centers. In the case of VM placement, the migrating VMs must be relocated to destination hosts; therefore, the PMs should naturally become cluster centers, and none of the migrating VMs should become cluster centers. Considering this requirement in the clustering process, we define  $c(i, i)$  as formula (8).

$$\begin{cases} c(i, i) = -Max\_Value & \text{if } i < vmNum \\ c(i, i) = -Min\_Value & \text{else} \end{cases} \quad (8)$$

where  $Max\_Value$  represents the maximum value in the compatibility matrix  $C$ . If  $c(i, i)$  is the preference of a VM, then  $Max\_Value$  is designated as the negative maximum value such that the VM cannot become a cluster center.  $Min\_Value$  represents the minimum value in the compatibility matrix  $C$ . If  $c(i, i)$  is the preference of a PM, then  $Min\_Value$  is designated as a negative minimum value, rendering the PM more likely to become a cluster center.

Based on the above, we propose a compatibility matrix generation (CMG) algorithm, as described in Algorithm 1.

### E. EVIDENCE-EFFICIENT CANDIDATE DESTINATION HOST SELECTION ALGORITHM

The compatibility of resources between VMs and PMs in a dimension can provide guidance for the selection of destination hosts for the migrating VMs and maintain the load balance of the destination hosts. This can effectively avoid the premature host overloading of a single resource, thereby eliminating its negative impact on QoS. However, if VM placement is conducted based only on the aforementioned compatibility, the presented VM placement scheme is unable to guarantee global optimal solution. Inspired by the idea of affinity propagation [15], we redefine the responsibility and availability based on the actual situation of the cloud data center as follows,

**Definition 2:** Responsibility denoted by  $r(i, j)$ , is the propensity of  $vm_i$  in selecting  $host_j$  as its destination host, i.e.,

**Algorithm 1** CMG

---

**Input:**  $Z$ ,  $vmNum$ ,  $hostNum$   
 //  $Z$ , the data set of resource vectors;  $vmNum$ , the number of VMs;  $hostNum$ , the number of PMs  
**Output:**  $C$  // the compatibility Matrix

- 1: **Initial**  $C$
- 2: **for each**  $z_i$  in  $Z$  **do**
- 3:   **for each**  $z_j$  in  $Z$  **do**
- 4:      $c(i, j) = -c(z_i, z_j)$  // Calculating the similarity between different samples.
- 5:   **end for**
- 6: **end for**
- 7: **for each**  $c(i, i)$  in  $C$  **do**
- 8:   **if**  $i < vmNum$  **do**
- 9:      $c(i, i) \leftarrow -Max\_Value$  // Assigning value to the preference of VMs
- 10:   **else if**
- 11:      $c(i, i) \leftarrow -Min\_Value$  // Assigning value to the preference of PMs
- 12:   **end if**
- 13: **end for**
- 14: **return**  $C$

---

the responsibility of  $host_j$  for serving as the destination host for the migrated virtual machine  $vm_i$ .

Responsibility is determined as shown in formula (9).

$$r(i, j) \leftarrow c(i, j) - \max_{s.t. j' \neq j} \{a(i, j') + c(i, j')\} \quad (9)$$

*Definition 3:* Availability denoted by  $a(i, j)$ , represents the degree of acceptance of  $host_j$  for deploying the migrated virtual machine,  $vm_i$ , i.e., the maximum tolerance of  $host_j$  to become the destination host for  $vm_i$ .

Availability is determined by the following formula (10).

$$a(i, j) \leftarrow \min \left\{ 0, r(j, j) + \sum_{s.t. i' \notin \{i, j\}} \max\{0, r(i', j)\} \right\} \quad (10)$$

where  $r(i, j)$  and  $a(i, j)$  are both initialized to 0.  $a(i, j')$  represents the availability of the running PMs other than  $host_j$  to  $vm_i$ .  $c(i, j')$  represents the compatibility between  $vm_i$  and running PMs other than  $host_j$ .  $r(i', j)$  represents the responsibility of VMs except  $vm_i$  for being placed on  $host_j$ .

Responsibility and availability are propagated in the sample space composed of VMs and PMs. Therefore, the availability of each sample (e.g., PM or VM) for itself, i.e., the compatibility of a sample in selecting itself as a cluster center, is calculated as formula (11).

$$a(j, j) \leftarrow \sum_{s.t. i' \neq j} \max\{0, r(i', k)\} \quad (11)$$

Responsibility and availability may encounter cyclical oscillations during propagation, causing the algorithm to fail to converge. To avoid this deficiency, we introduce the learning rate  $\lambda$  into the updates of responsibility and availability.

The responsibility and availability were updated using formulas (12) and (13), respectively.

$$r_{t+1}(i, j) \leftarrow (1 - \lambda)r_{t+1}(i, j) + \lambda r_t(i, j) \quad (12)$$

$$a_{t+1}(i, j) \leftarrow (1 - \lambda)a_{t+1}(i, j) + \lambda a_t(i, j) \quad (13)$$

where  $\lambda$  is an empirical value in the experiments and set to 0.5 in this study.

Summarizing the above steps, we have an algorithm for candidate destination host selection, namely, Evidence-propagation-based Candidate Destination Host Selection (EPCDHS) algorithm, described as Algorithm 2. Algorithm 2 can be used to calculate the responsibility and availability between the migrated VMs and destination hosts. The sum of responsibility and availability can be regarded used as the accumulated evidence for selecting the candidate destination hosts.

**Algorithm 2** EPCDHS

---

- 1: **Input:**  $C$  // the compatibility matrix
- 2: **Output:**  $r, a, pm_j$  //  $r$ : responsibility;  $a$ : availability;  $pm_j$ : destination hosts
- 3: **Initial**  $r$
- 4: **Initial**  $a$
- 5: **while**  $iterTimes < setNum$  **do**
- 6:   **for each**  $r_{ij}$  **do**
- 7:     use formula (9), (12) calculate  $r_{ij}$  // Calculating the responsibility between samples
- 8:     put  $r_{ij}$  in  $r$
- 9:   **end for**
- 10:   **for each**  $a_{ij}$  **do**
- 11:     use formula (10), (13) calculate  $a_{ij}$  // Calculating the availability between samples
- 12:     **if**  $i == j$  **do**
- 13:       use formula (11) calculate  $a_{ij}$  // Calculating the availability of the sample itself
- 14:       put  $a_{ij}$  in  $a$
- 15:     **endif**
- 16:   **end for**
- 17: **endWhile**
- 18:  $pm_j \leftarrow \max(r_{ij} + a_{ij})$  // Selecting the PMs with the largest accumulative evidence of the sum of responsibility and availability as the candidate destination hosts
- 19: **return**  $r, a, pm_j$

---

The algorithm inherits the basic architecture of the well-known AP algorithm [15], so the time complexity of EPCDHS is the product of the number of iterations and the total amount of samples.

**IV. EEAP-VMP METHOD****A. DESTINATION HOST SELECTION CRITERION**

Reducing energy consumption is one of the primary challenges of virtual resource scheduling and management in a cloud data center. The energy consumption of cloud data centers is comprehensively impacted by various factors in

data centers. Therefore, it is necessary to consider energy consumption while selecting destination hosts for the VMs to be migrated. Hence, we propose a destination host selection criterion, as shown in formula (14).

$$pm_j \leftarrow \max\{(r(i, j) + a(i, j)) \cdot e^{-U_{pm_j}^{+vm_i}}\} \quad (14)$$

where  $U_{pm_j}^{+vm_i}$  indicates the growth rate of energy consumption of the destination host  $pm_j$  after  $vm_i$  is allocated on  $pm_j$ . This is calculated using formula (15).

$$U_{pm_j}^{+vm_i} = \frac{EC_{pm_j}^{+vm_i} - EC_{pm_j}}{EC_{pm_j}} \quad (15)$$

where  $EC_{pm_j}$  represents the energy consumption of  $pm_j$ ; and  $EC_{pm_j}^{+vm_i}$  represents the energy consumption of the destination host  $pm_j$  after  $vm_i$  is placed on  $pm_j$ .

In fact, Eq. (14) implies the criterion that, the running PMs, with the largest product of the cumulative evidence and energy consumption growth ratio after finishing VM placement on them, are selected as the final destination host for performing VM placement.

## B. VM PLACEMENT METHOD

Combining the CMG algorithm in Algorithm 1 and the EPCDHS algorithm in Algorithm 2, we propose the EEAP-VMP method. First, the EEAP-VMP method calculates the resource compatibility between VMs and PMs using the CMG algorithm to obtain a resource compatibility matrix. Subsequently, the responsibility and availability are calculated using the EPCDHS algorithm. The accumulated evidence of responsibility and availability is propagated in the sample space composed of VMs and PMs to provide effective evidence for selecting candidate destination hosts. Finally, the final destination host is determined for live VM migration according to the destination host selection criterion (i.e., Eq. (14)) to place the migrating VM. The EEAP-VMP algorithm is described as follows:

### Algorithm 3 EEAP-VMP

---

```

1: Input:  $VM_{list}, PM_{list}$ ;
2: Output:  $Map < vm, host >$ ;
3: while  $VM_{list} \neq \emptyset$  do
4:   if  $PM_{list} \neq \emptyset$ 
5:      $C \leftarrow CMG(VM_{list}, PM_{list})$  // Calling algorithm 1
6:      $r, a, pm_j \leftarrow EPCDHS(C)$  // Calling algorithm 2
7:   end if
8: end while
9: use formula (14) find  $pm_j$  for the migrating  $vm_i$  // Selecting destination hosts
10: put  $vm_i$  and  $pm_j$  in the data structure of map
11: return  $Map < vm, host >$ 

```

---

The time cost of the proposed EEAP-VMP algorithm mainly comes from line 6, which is identical to the time complexity of algorithm 2.

TABLE 1. Physical machine type.

Types	CPU(MIPS)	RAM(GB)
HP ProLiant ML110 G4	Intel Xeon 3040/1860	4
HP ProLiant ML110 G5	Intel Xeon 3075/2660	4

TABLE 2. Virtual machine type.

Types	CPU(MIPS)	RAM (GB)
High-CPU Medium Instance	2500	0.85
Extra Large Instance	2000	3.75
Small Instance	1000	1.7
Micro Instance	500	0.633

## V. EXPERIMENTAL RESULTS AND ANALYSIS

### A. EXPERIMENT SETUP

To validate the actual performance of the proposed algorithms, we employ the CloudSim [16] simulation platform for experiments and simulates a data center composed of 800 heterogeneous hosts. Instances of PMs are shown in Table 1.

With respect to the VMs instance type provided by Amazon Elastic Compute Cloud (EC2) [12], combined with the real cloud data centers, 4 different types of VM instances were selected in the experiment as shown in Table 2.

To validate the effectiveness and efficiency of the proposed algorithms, the experiment selected two real workload data trace: Bitbrains [17]; Alibaba Cluster Data V2018 [18]. Due to the huge number of Bitbrains data sets and Alibaba Cluster Data V2018 data sets, 10 days of running records were randomly selected from the two items of data sets for experiments.

### B. EVALUATION INDICES

Beloglazov et al. [4] presented four performance evaluation indicators for data centers. Here, we use six performance indices to evaluate performance in our experiments: SLA violation time per running PM (SLATAH), performance degradation due to migration (PDM), SLA violations (SLAV), energy consumption (EC), and energy and SLA violations (ESV) [3], [13], as well as Virtual machine migrations (VMMs).

a) SLATAH, as a measure of QoS for a running host, is formulized as

$$SLATAH = \frac{1}{n} \sum_{j=1}^n \frac{\ell_j^{violation}}{\ell_j} \quad (16)$$

where  $\ell_j^{violation}$  is the duration time of SLAV resulting from the overloading CPU resource of host  $h_j$ ,  $\ell_j$  the running duration time of the host  $h_j$ , and  $n$  the number of PMs.

b) PDM measures the extent of performance decline of VM-migration-related situation, which is calculated as

formula (17),

$$PDM = \frac{1}{m} \sum_{i=1}^m \frac{R_i^{mig}}{R_i} \quad (17)$$

where  $R_i^{mig}$  indicates the size of unsatisfied demand for CPU resources as a result of a case of VM migration of a given virtual machine  $v_i$ ,  $R_i$  the total amount of demand for CPU resources from the given virtual machine  $v_i$ , and  $m$  the number of VMs.

c) SLAV is defined to measure QoS of data centers on a single day, which is defined as

$$SLAV = SLATAH \times PDM \quad (18)$$

SLATAH, PDM, and SLAV are inversely proportional to QoS.

d) ESV is an integrated evaluation index, which is defined by formula (19). ESV denotes the comprehensive performance of energy consumption, VMs and service quality.

$$ESV = EC \times SLAV \quad (19)$$

where EC represents the energy consumption of data centers on a single day, which is determined by formula (7). A lower value of ESV implies that the presented scheme can save more energy and guarantee QoS.

Since VMs always suspend services during live VM migration, prolonged VM migrations can degrade QoS. Since reducing the VM migrations can improve QoS, if limited number of VM migrations is enough to yield desirable performance using a given VM placement method, it indicates that the scheme of VM placement is highly efficient.

### C. EFFECTIVENESS

To study the effectiveness of the proposed EEAP-VMP algorithm, we compared it with UPBFD [3], FFD [2], and PABFD [4] algorithms in terms of the six evaluation indicators proposed in part B of Section V. According to [22], VM placement is crucial in VM consolidation, and it includes several stages of host overload detection, VM selection, and VM placement [19]. In this study, we applied the static threshold (ST), interquartile range (IQR), and mean absolute deviation (MAD) methods proposed in [4] to perform PM overload detection, and the maximum correlation (MC) selection [4], random selection (RS) [4], and minimum migration time (MMT) selection [4] algorithms to select the VMs to be migrated. The PM overload detection algorithms and VM selection algorithms formed nine different combinations, namely IQR-MMT, IQR-MC, IQR-RS, MAD-MC, MAD-MMT, MAD-RS, ST-MC, ST-MMT, and ST-RS. Each of nine combinations is then combined with the four VM placement algorithms, i.e., EEAP-VMP, UPBFD [3], FFD [2], and PABFD [4] to form 36 instances of VM consolidation methods. For all of the 36 VM consolidation methods, the pre-copy mechanism [25] is employed to perform live

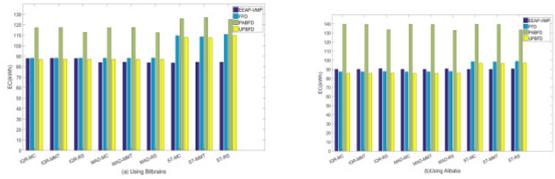


FIGURE 1. Comparison of EC.

VM migration. We tested the 36 integrated VM consolidation methods using the Bitbrains data trace and Alibaba Cluster Data V2018 dataset and compared their experimental results based on the six indicators.

EC represents the energy consumption of running PMs in a data center. Fig. 1 compares the energy consumption obtained using the 36 VM consolidation methods. Figs. 1(a) and 1(b) show the compared results on the Bitbrains data trace and Alibaba Cluster Data V2018 datasets, respectively. As shown in Fig.1(a), when IQR-MC, IQR-MMT, and IQR-RS are used to handle host overloading detection and VM selection, the energy consumptions obtained using the EEAP-VMP, FFD, and UPBFD algorithms are similar but lower than that obtained using PABFD. Compared with the other six combinations (e.g., MAD-MC, MAD-MMT, MAD-RS, ST-MC, ST-MMT, and ST-RS), the energy consumption obtained using the EEAP-VMP algorithm is the lowest. In particular, when ST is employed as the host overloading detection method, the energy consumption obtained using EEAP-VMP reduced by 22.7% to 32% compared with the other algorithms, indicating the greater advantage of EEAP-VMP according to energy consumption. This is because in a real data center, the requests by VMs for different types of resources are uncertain, resulting in dynamic changes in the resource load of the PMs. However, the ST algorithm fixes the overload thresholds of different types of resources on the PMs; therefore, the EEAP-VMP algorithm reduces the energy consumption. The FFD, UPBFD, and PABFD algorithms use a greedy selection strategy for VM placement. Consequently, a certain type of resource of running PMs is expected to prematurely reach the threshold by the ST algorithm, leading to load imbalance. Meanwhile, more PMs must be turned on to satisfy the randomly requested resource by VMs, thereby increasing the energy consumption of data centers.

Additionally, the EEAP-VMP algorithm analyzes the compatibility of various resources between PMs and the migrated VMs during VM placement, which achieves load balancing among various resources of the PMs more effectively and reduces unnecessary resource wastage. Therefore, EEAP-VMP still performs better with regard to energy consumption even if the ST algorithm is used as a host overloading detection algorithm. As shown in Fig. 1(b), EEAP-VMP does not perform well in terms of energy consumption on the Alibaba Cluster Data V2018 dataset. Only when it is combined with ST-MC, ST-MMT, or ST-RS, it outperforms other

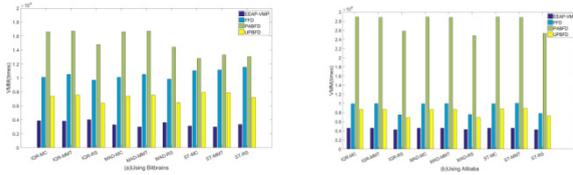


FIGURE 2. Comparison of VMs.

algorithm combinations on energy consumption. When it is combined with algorithms other than ST-MC, ST-MMT, and ST-RS, its energy consumption is slightly higher than those of the FFD and UPBFD algorithms are. This is because in the Alibaba Cluster Data V2018 dataset, the requests of VMs for memory resources are approximately three to four times the requests for CPU resources [8]. The energy consumption calculation model utilized in this study depends on the CPU utilization of PMs. Therefore, when the EEAP-VMP algorithm is used to optimize the loads of various resources of the PMs during VM placement, more PMs have to be turned on to satisfy the random requests of the VMs for memory resources, resulting in increased energy consumption. Through a combined analysis of Figs. 1(a) and 1(b), we discover that the EEAP-VMP algorithm fluctuates slightly in terms of energy consumption on the Alibaba Cluster Data V2018 dataset. This is because the EEAP-VMP algorithm finds destination hosts that are more compatible for the migrated VMs by propagating the cumulative evidence of responsibility and availability, thereby achieving load balancing on the destination hosts.

VM migration will increase the consumption of bandwidth resources in the data center. VMs suspend services during the duration of live VM migration, thereby negatively impact the QoS [23]. Therefore, a good VM placement method should minimize the number of VM migrations. Fig. 2 compares the number of VM migrations among different algorithms. Figs. 2(a) and 2(b) show the compared results on the Bitbrains data trace and Alibaba Cluster Data V2018 datasets, respectively. As shown in Figs. 2(a) and 2(b), the amount of VM migrations of the EEAP-VMP algorithm stabilizes at approximately 4,000, which is much lower than the other compared algorithms are. This implies that the proposed EEAP-VMP algorithm can well balance various types of resources. It mitigates the premature PM overloading risk of a single resource of PMs, thereby effectively avoiding insignificant VM migrations. As shown in Fig. 2, the total number of VM migrations of the UPBFD algorithm is second only to the EEAP-VMP algorithm because the UPBFD algorithm mitigates the risk of PM overload by predicting the resource utilization of PMs. By combining Figs. 2(a) and 2(b), we discover that the amount of VM migrations of the PABFD algorithm differs significantly on the two datasets. The number of VM migrations on the Bitbrains data trace ranged between 12,000 and 16,000, whereas that on the Alibaba Cluster Data V2018 dataset reached 24,000–28,000, with an increase of

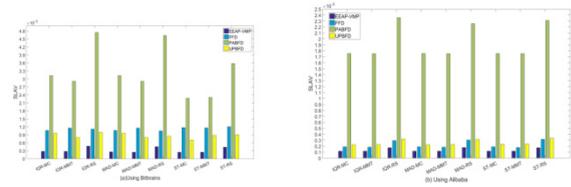


FIGURE 3. Comparison of SLAV.

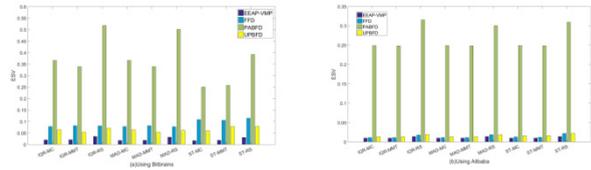


FIGURE 4. Comparison of ESV.

approximately 90% compared with the Bitbrains dataset. This is because the PABFD algorithm always preferentially places the VMs on the PMs with the least energy consumption growth. However, the PABFD algorithm disregards the physical machine requirements for load balancing among various types of resources. Consequently, a certain single physical resource of PMs may be susceptible to premature overloading, causing frequent insignificant VM migration.

Fig. 3 shows the compared results of the SLAV index on two different datasets. The results on the Bitbrains data trace are shown in Fig. 3(a), while those on the Alibaba Cluster Data V2018 dataset are shown in Fig. 3(b). Combining the analysis of Figs. 3(a) and 3(b), we find that the SLAV of EEAP-VMP on the Bitbrains dataset is 0.0003–0.0006, whereas that on the Alibaba Cluster Data V2018 dataset is 0.0001–0.0002. The EEAP-VMP algorithm performs the best in the SLAV indicator on both datasets. This is because EEAP-VMP fully matches the compatibility between the remaining resources of the PMs and the requested resources by the VMs. Through compatibility and propagation of cumulative evidence, EEAP-VMP achieves the optimal scheme of VM placement, thereby reducing the risk of PM overload. As shown in Fig. 3, when the RS algorithm is employed as the VM selection algorithm, all the compared algorithms indicate a significant increase in SLAV on the two datasets because the RS algorithm randomly selects the migrated VMs. Although this mechanism is general, it cannot guarantee that the final selection of migrating VMs is the most appropriate. Consequently, some VMs that must be re-allocated cannot be effectively migrated.

ESV considers both energy consumption and service-level agreement violation of data centers. Thus, it can comprehensively measure the effectiveness of the algorithm [24]. Fig. 4 shows the comparison results of ESV among four VM placement algorithms. The results on the Bitbrains dataset are shown in Fig. 4(a), whereas those on the Alibaba Cluster Data V2018 dataset are shown in Fig. 4(b). As shown in Figs. 4(a) and 4(b), the VM consolidation method integrating

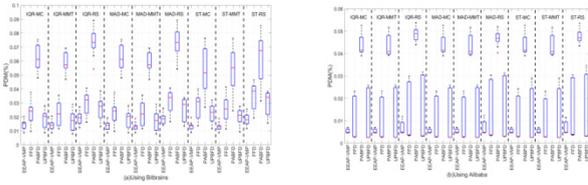


FIGURE 5. Comparison of PDM.

EEAP-VMP is superior to FFD, PABFD, and UPBFD on the two datasets. This is because ESV is the product of EC and SLAV. Although EEAP-VMP differs slightly from the FFD and UPBFD algorithms in EC, it exhibits a significant advantage in SLAV. Therefore, EEAP-VMP can still perform well in ESV. Comparing the experimental results on the two datasets, we observe that the SLAV values of the EEAP-VMP, FFD, and UPBFD algorithms on the Alibaba Cluster Data V2018 dataset are similar. However, their SLAV values on the Bitbrains dataset differs significantly. Further, combining Figs. 1 and 3, we discover that this is mainly because the SLAV and EC values of the EEAP-VMP, FFD, and UPBFD algorithms on the Alibaba Cluster Data V2018 dataset are similar.

Fig. 5 shows a comparison result of the PDM on two datasets. Figs. 5(a) and 5(b) show the index of PDM on the Bitbrains data trace and Alibaba Cluster Data V2018 datasets, respectively. Combining Fig. 5(a) and Fig. 5(b), we find that the VM consolidation method integrating EEAP-VMP yield the optimal PDM value on the two datasets, and its PDM changes stable with small fluctuations. The median and upper limits of the PDM of the EEAP-VMP algorithm on the Bitbrains dataset are both less than 0.03, whereas those on the Alibaba Cluster Data V2018 dataset are less than 0.02. This indicates the robustness of EEAP-VMP, which attempts to maintain load balancing of resources in the destination hosts. In addition, the VMMs analysis in Fig.2 indicates that the fewer number of VM migrations of EEAP-VMP reduces the QoS degradation caused by VM migration. Similarly, the PABFD algorithm performs poorly with respect to the PDM on the Bitbrains data trace and Alibaba Cluster Data V2018 datasets owing to its high number of VM migrations. Comparing Figs. 5(a) and Fig. 5(b), we observe that the PDM values of the FFD and UPBFD algorithms on the Alibaba Cluster Data V2018 dataset changes significantly. This case indicates that the FFD and UPBFD algorithms have a large amount of VM migrations during VM consolidation, which impacts the PDM. This is also supported by the VMMs results in Fig. 2.

Fig. 6 shows a comparison result of the SLATAH of EEAP-VMP and the compared algorithms. The compared results on the Bitbrains and Alibaba Cluster Data V2018 datasets are shown in Fig. 6(a) and Fig.6(b), respectively. The VM consolidation method integrating EEAP-VMP yielded an SLATAH value superior to those of the other compared algorithms. This is because the EEAP-VMP algorithm fully

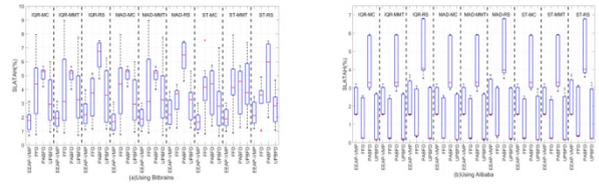


FIGURE 6. Comparison of SLATAH.

matches the requested resources by the VMs with the remaining resources of the PMs to obtain the most compatible destination hosts for the migrated VMs. This reduces the SLA violation time of the PMs while effectively mitigating the risk of premature PM overloading of a single resource. Fig. 6(b) shows the compared results on the Alibaba Cluster Data V2018 dataset. Among them, the FFD and UPBFD algorithms demonstrate better SLATAH index than that of EEAP-VMP, and the median, upper limit, and lower limit of the SLATAH indicator of EEAP-VMP are only superior to those of PABFD. This is because in the Alibaba Cluster Data V2018 dataset, the memory resource required by the PMs is approximately three to four times that required by the CPU resource [8], causing the single memory resource of the PMs to be prematurely overloaded.

By analyzing the six evaluation indicators, we find that the proposed EEAP-VMP performs better than those of the compared FFD, PABFD, and UPBFD algorithms in terms of the VMMs, SLAV, ESV, PDM, and SLATAH. This clearly shows that the consumption of various resources of the PMs can be balanced by abstracting the PM and VM resources into resource vectors and entirely matching the compatibility through the mechanism of affinity propagation of cumulative evidence of responsibility and availability. Therefore, EEAP-VMP can more effectively reduce the number of VM migrations and energy consumption, while maintaining the load balance of PMs.

#### D. EFFICIENCY

In this section, the efficiency of the proposed EEAP-VMP algorithm is analyzed and validated in terms of various resource utilization and variation in the number of running PMs. In the effectiveness analysis experiment presented in part C of Section V, the indicators of the MAD-MMT method indicate large discriminations. Thus, we select this combined method as the host overloading detection and VM selection methods. We combine it with the EEAP-VMP, FFD, PABFD, and UPBFD algorithms into four VM consolidation methods for efficiency validation. We assume that VMs consolidated are done every 5 min for a total of 288 cycles per day. The resource utilization of PMs and the number of running PMs after each cycle of VM consolidation are compared among the four compared methods. Figs. 7, 8, and 9 show the average CPU, memory, and bandwidth utilization of the running PMs, respectively. Figs. 10 and 11 show the variation in the number of PMs of the proposed EEAP-VMP algorithm

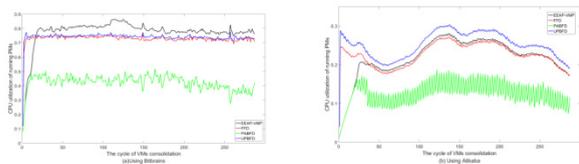


FIGURE 7. Comparison of CPU utilization of PMs.

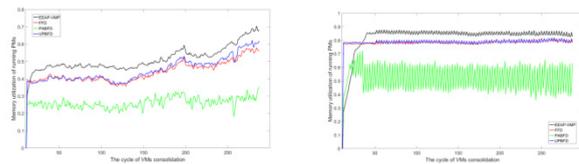


FIGURE 8. Comparison of Memory utilization of PMs.

on the Bitbrains data trace and Alibaba Cluster Data V2018 datasets, respectively.

Fig. 7 shows the variation in the CPU utilization of the PMs during ongoing VM consolidation. The experimental results on the Bitbrains dataset are presented in Fig. 7(a), whereas those on the Alibaba Cluster Data V2018 dataset are shown in Fig. 7(b). As shown in Fig. 7(a), EEAP-VMP obtains a higher CPU utilization than that of the compared algorithms during all 288 cycles of VM consolidation. As shown in Fig. 7(b), EEAP-VMP achieves a higher CPU utilization than the FFD and PABFD algorithms but lower CPU utilization than that of the UPBFD algorithm. This is because the data in the Alibaba Cluster Data V2018 dataset are not CPU resource intensive, and the VMs request more memory resources. Therefore, the EEAP-VMP algorithm balances various resources requested by the VMs during ongoing VM consolidation. It first satisfies the memory requirements to maintain the memory utilization at a high level. Consequently, the CPU utilization of PMs of EEAP-VMP is lower than the other algorithms are. As shown in Fig. 7(a) and Fig. 7(b), as for the Bitbrains data trace and the Alibaba Cluster Data V2018 datasets, the CPU utilization of the proposed EEAP-VMP algorithm increases slowly at an early stage of VM consolidation. This shows that EEAP-VMP cannot quickly find VM placement with the most suitable destination hosts; however, the resource utilization can reach its peak only after several cycles of VM consolidation. This is owing to the mechanism of affinity propagation in the EEAP-VMP algorithm. As the number of cycles of VM consolidation increase, more evidence of responsibility and availability is accumulated. Subsequently, after sufficient cycles of VM consolidations, the algorithm obtains a more compatible destination host for the migrated VM.

Fig. 8 shows the compared results of memory utilizations for the four algorithms on the Bitbrains and Alibaba Cluster Data V2018 datasets respectively. The experimental results on the Bitbrains data trace are shown in Fig. 8(a), and those on the Alibaba Cluster Data V2018 dataset are shown in Fig. 8(b). As shown in Fig. 8(a), the memory utilization of

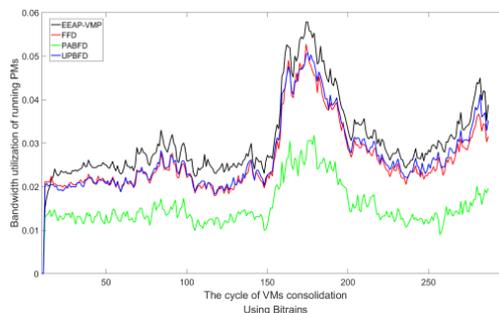


FIGURE 9. Comparison of Bandwidth utilization of PMs.

PMs obtained using EEAP-VMP is higher than that of the other compared algorithms. From the 50-th to 288-th cycle of VM consolidation, the memory utilization of EEAP-VMP is stable at 50%–70%, whereas that of the other algorithms ranged between 20% and 60%. The mechanism of affinity propagation of accumulation evidence in EEAP-VMP maintains load balancing in the destination hosts, thereby sustaining the memory utilization at a reasonable level. As shown in Fig. 8(b), the memory utilization of EEAP-VMP is stable at 80% to 90%, which is superior to those of the FFD, PABFD, and UPBFD algorithms. In addition, EEAP-VMP is more stable than the PABFD algorithm, and has smaller fluctuations compared with PABFD. This is because EEAP-VMP matches the compatibility of all types of resources and obtained the global optimal destination host for the migrated VM through the accumulation evidence. In this case, the number of VM migrations decreases, rendering a steady variation in the resource utilization. The memory utilization of the PABFD algorithm fluctuates significantly because the algorithm caused frequent on-off switching of idle PMs during VM consolidation, thereby resulting in excessive meaningless VM migrations. In combination with analysis of Fig. 8(a) and Fig. 8(b), we discover that the memory utilization exhibits a similar trend with the CPU utilization trend shown in Fig. 7. Although EEAP-VMP generally performs better than that of the other algorithms in terms of memory utilization, it indicates a smaller growth rate in the memory utilization compared with the other algorithms at the early stage of the entire VM consolidation, i.e., at the 25th to 50th cycle. This is because EEAP-VMP uses an evidence accumulation mechanism. In the case of limited cycles of VM consolidation, the accumulated evidence is limited; therefore, the utilization of memory resources at the early stage increases slowly. Along with the number of VM consolidation cycles uptrends, the accumulated evidence increases; therefore, the memory utilization of EEAP-VMP at the end of the 50-th cycle is the highest.

Fig. 9 shows the variation in the bandwidth utilization of the PMs using the four algorithms. The bandwidth load data trace in the Bitbrains dataset are used in our experiments. As shown in the figure, when EEAP-VMP is used, the bandwidth utilization is always higher than those of the other algorithms are, and the PM bandwidth utilization of

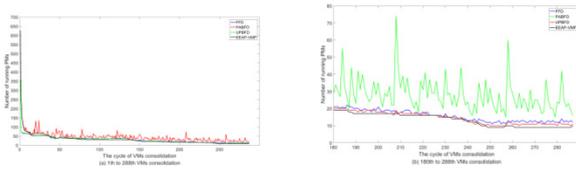


FIGURE 10. Number of running PMs on Bitbrains.

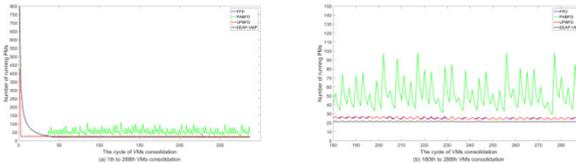


FIGURE 11. Number of running PMs on Alibaba cluster data V2018.

all the four VM placement algorithms is less than 0.06%. This indicates that the remaining bandwidth resource of the PMs is much greater than the bandwidth resources requested by the migrating VMs. Moreover, it indicates that none of the four VM placement algorithms caused insignificant VM migrations; therefore, the bandwidth consumption is low. This is verified by the VMMs results shown in Fig. 2.

Fig. 10 and Fig. 11 show the curves of the number of running PMs with the cycles during ongoing VM consolidation using Bitbrains data trace and Alibaba Cluster Data V2018 dataset respectively. Fig. 10(a) and Fig. 11(a) show the VM consolidation from the first circle to the 288th cycle. For clarity, Figure 10(b) and 11(b) are the partial sub-graphs of the corresponding Figs. 10(a) and 11(a), which display the changes of the running PMs from the 180th to 288th cycle of VM consolidation. In combination with Fig. 10(a) and Fig. 11(a), it can be found that the four compared VM consolidation methods are able to effectively degrade the total number of running PMs, thereby reducing energy consumption of data centers. However, in contrast, the number of running PMs of the proposed EEAP-VMP algorithm decreases more slowly than that of the other compared algorithms in the early stage of VM consolidation. This is because the EEAP-VMP algorithm entirely matches the compatibility between the migrated VMs and the destination hosts through affinity propagation of the accumulative evidence of responsibility and availability. Only as the number of VM consolidation increases, the accumulative evidence of responsibility and availability becomes more and more sufficient. In contrast, in the early stages, due to insufficient accumulation evidence, the number of PMs in data centers degraded relatively slowly.

Based on Figs. 10(b) and 11(b), the proposed EEAP-VMP algorithm can reduce the number of running PMs efficiently. With respect to the compared results on the Bitbrains data trace, the number of running PMs reduces to approximately 10. The experimental results on the Alibaba Cluster Data V2018 dataset show that the amount of running PMs stabilizes at approximately 20. Additionally, although the

UPBFD and FFD algorithms degrade the number of running PMs to a relatively low level of approximately 25, the number of running PMs varies often during VM consolidation. This is because UPBFD and FFD donot consider host load balancing, thereby inducing the risk of premature overloading of a single resource of PMs. In this case, more PMs must be turned on to satisfy the random requests of the VMs for a single resource. When finding hosts in the low load status, the UPBFD or FFD algorithm turns off low-loaded PMs, which causes the PMs to frequently switch between power-on and sleep modes. This case results in frequent changes of the number of the running PMs and triggers more live VM migration.

As shown above, a comparative analysis is performed for various resource utilization of the running PMs using the Bitbrains data trace and Alibaba Cluster Data V2018 datasets, as well as the amount of running PMs, CPU utilization, memory utilization, and bandwidth utilization of the four compared algorithms. The experimental results indicate that the resource utilization of EEAP-VMP are relatively high, which facilitates in maintaining the load balance of the hosts. Additionally, EEAP-VMP performs well in guaranteeing QoS, resulting from efficient VM placements.

## VI. CONCLUSION AND FUTURE WORKS

Cloud computing has grew significantly, because it is capable of providing users with on-demand accessing, flexible expansion, and low-cost services. VM placement is the key issue during VM consolidation. Improper VM placement leads to low resource utilization and high energy consumption. Efficient and effective VM placement with optimizing the deployment relation between VMs and PMs, are critical for improving resource utilization, reducing energy consumption, and guaranteeing QoS.

This paper addresses the issue of workload imbalance of various type resources on destination hosts during live VM migration. First, by abstracting the requested resources of the migrated VM and the remaining resources of the running PMs as resource vector samples, a compatibility matrix between the migrated VMs and the running PMs is derived based on the dissimilarities and similarities between the different samples; Second, we re-define the concept of responsibility and availability to allow them to affinity propagate between different samples. The accumulative evidence of responsibility and availability is used to select the cluster centroids. Here, the equivalent cluster centroids, namely the physical machines, are the candidate destination hosts to allocate the migrated VMs; Finally, with combination the presented accumulative evidence and the model of energy consumption of PMs, an energy-efficient destination host selection criterion is proposed. Based on this, an evidence-efficient affinity propagation scheme for virtual machine placement (EEAP-VMP) algorithm is proposed. Simulation experiments show that the proposed EEAP-VMP algorithm effectively balances

the workload of various types of resources of running PMs and reduces the number of VM migrations. We show that EEAP-VMP has a greater advantage in improving resource utilization, reducing energy consumption, balancing workload of PMs, and guaranteeing QoS naturally.

However, there are several limitations that need to be further studied in our future works. The EEAP-VMP algorithm regards the PM as the clustering centroids during VM consolidation, so in the compatibility matrix, the physical host preference is pre-determined as the default maximum value, and the VM preference is pre-configured as the default minimum value. The default maximum value of preference can ensure that the cluster centroid is always the PMs during the iterative updating process, which inevitably incur waste of resources without consideration of the resource competition between PMs themselves. In theory, the preference should be adaptively adjusted with respect to the real workload of PMs and the risk of overloading, so that the PMs with a lower preference is gradually eliminated during the iterative process, leading to a better VM placement solution.

## REFERENCES

- [1] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilisation of resources in cloud computing systems," *Super Compute*, vol. 2012, no. 60, pp. 268–280, 2012.
- [2] A. Alahmadi, A. Alnowiser, M. M. Zhu, D. Che, and P. Ghodous, "Enhanced first-fit decreasing algorithm for energy-aware job scheduling in cloud," in *Proc. Int. Conf. Comput. Sci. Comput. Intell.*, Mar. 2014, pp. 69–74.
- [3] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, and H. Tenhunen, "Utilization prediction aware VM consolidation approach for green cloud computing," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, Jun. 2015, pp. 381–388.
- [4] 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.
- [5] F. Alharbi, Y.-C. Tian, M. Tang, W.-Z. Zhang, C. Peng, and M. Fei, "An ant colony system for energy-efficient dynamic virtual machine placement in data centers," *Expert Syst. Appl.*, vol. 120, pp. 228–238, Apr. 2019.
- [6] X.-F. Liu, Z.-H. Zhan, K.-J. Du, and W.-N. Chen, "Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach," in *Proc. Conf. Genetic Evol. Comput.*, 2014, pp. 41–48.
- [7] Alharbi, Fares, "Profile-based ant colony optimization for energy-efficient virtual machine placement," *Proc. 24th Int. Conf. Neural Inf. Process.*, vol. 2017, pp. 863–871, 2017.
- [8] Z. Li, X. Yu, L. Yu, S. Guo, and V. Chang, "Energy-efficient and quality-aware VM consolidation method," *Future Gener. Comput. Syst.*, vol. 102, pp. 789–809, Jan. 2020.
- [9] Z. Li, C. Yan, L. Yu, and X. Yu, "Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic consolidation method," *Future Gener. Comput. Syst.*, vol. 80, pp. 139–156, Mar. 2018.
- [10] M. H. Ferdous, "Virtual machine consolidation in cloud data centers using ACO metaheuristic," in *Proc. Eur. Conf. Parallel Processing.*, 2014, pp. 306–317.
- [11] Z. Li, C. Yan, X. Yu, and N. Yu, "Bayesian network-based virtual machines consolidation method," *Future Gener. Comput. Syst.*, vol. 69, pp. 75–87, Apr. 2017.
- [12] S. B. Melhem, A. Agarwal, N. Goel, and M. Zaman, "Markov prediction model for host load detection and VM placement in live migration," *IEEE Access*, vol. 6, pp. 7190–7205, 2018.
- [13] Z. Li, X. Yu, and L. Zhao, "A strategy game system for QoS-efficient dynamic virtual machine consolidation in data centers," *IEEE Access*, vol. 7, pp. 104315–104329, 2019.
- [14] P. C. Mahalanobis, "On the generalised distance in statistics," *Proc. Nat. Inst. Sci. India*, vol. 2, no. 1, pp. 49–55, 1936.
- [15] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
- [16] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Jan. 2011.
- [17] S. Shen, V. V. 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.
- [18] Alibaba. Accessed: Sep. 18, 2019. [Online]. Available: <https://github.com/alibaba/clusterdata>
- [19] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012.
- [20] [Online]. Available: [https://www.spec.org/power\\_ssj2008/](https://www.spec.org/power_ssj2008/)
- [21] D. Kusic, "Power and performance management of virtualized computing environments via look ahead control," *Cluster Comput.*, vol. 12, no. 1, pp. 1–15, 2009.
- [22] H. Wang and H. Tianfield, "Energy-aware dynamic virtual machine consolidation for cloud datacenters," *IEEE Access*, vol. 2018, pp. 15259–15283, 2018.
- [23] S. Mustafa, B. Nazir, A. Hayat, A. U. R. Khan, and S. A. Madani, "Resource management in cloud computing: Taxonomy, prospects, and challenges," *Comput. Electr. Eng.*, vol. 47, pp. 186–203, Oct. 2015.
- [24] T. H. Duong-Ba, T. Nguyen, B. Bose, and T. T. Tran, "A dynamic virtual machine placement and migration scheme for data centers," *IEEE Trans. Services Comput.*, early access, Mar. 19, 2019, doi: 10.1109/TSC.2018.2817208.
- [25] K. Z. Ibrahim, "Optimized pre-copy live migration for memory intensive applications," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, vol. 26, 2011, pp. 1–11.



**ZHIHUA LI** received the diploma and Ph.D. degrees in computer science from Jiangnan University, China. He is currently a Faculty Member with the Department of Computer Science and Technology, Jiangnan University. His research interests include network technology, parallel/distributed computing, information security, edge computing, and mobile computing.



**SHUJIE GUO** is currently pursuing the M.S. degree with the Department of Computer Science and Technology, Jiangnan University, Jiangsu, Wuxi, China. His current research interests include cloud computing and distributed computing.



**LEI YU** received the B.S. and M.S. degrees in computer science from the Harbin Institute of Technology, China, and the Ph.D. degree in computer science from the Georgia Institute of Technology, USA. He is currently a Research Staff Member with the IBM Research, Yorktown Heights, NY, USA. His research interests include cloud computing, data privacy, machine learning security, and mobile computing.



**VICTOR CHANG** was a Senior Associate Professor, the Director of Ph.D. from June 2016 to May 2018, and the Director of MRes from September 2017 to February 2019 with the International Business School Suzhou (IBSS), Xi'an Jiaotong-Liverpool University (XJTLU), Suzhou, China, from June 2016 to August 2019. He was also a Very Active and a Contributing Key Member with the Research Institute of Big Data Analytics (RIBDA), XJTLU. He was an Honorary Associate

Professor with the University of Liverpool. Previously, he was a Senior Lecturer with Leeds Beckett University, U.K., from September 2012 to May 2016. He has been a Full Professor of data science and information systems with the School of Computing, Engineering and Digital Technologies, Teesside University, Middlesbrough, U.K., since September 2019. He currently co-leads and leads two Research Groups at Teesside University. Within four years, he completed the Ph.D. (CS, Southampton) and PGCert (Higher Education, Fellow, Greenwich) while working for several projects at the same time. Before becoming an academic, he has achieved 97% on average in 27 IT certifications. He has published three books as sole authors and the Editor of two books on *Cloud Computing* and related technologies. He gave 18 keynotes at international conferences. He is widely regarded as one of the most active and influential young scientist and expert in IoT/Data Science/Cloud/security/AI/IS, as he has experience to develop ten different services for multiple disciplines. He won the European Award on Cloud Migration in 2011, the IEEE Outstanding Service Award in 2015, best papers in 2012, 2015, and 2018, the 2016 European Special Award, the Outstanding Young Scientist in 2017, and the INSTICC Service Award since 2017. He is also a Visiting Scholar/Ph.D. Examiner with several universities, an Editor-in-Chief of the *International Journal of Organizational and Collective Intelligence* (IJOCI) and the *Open Journal of Blood Diseases* (OJBD) journals, the former Editor of FGCS, an Associate Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS (TII) and *Information Fusion*, the Founding Chair of two international workshops, and the founding Conference Chair of IoTBDS and COMPLEXIS since 2016. He is the founding Conference Chair of FEMIB since 2019.

• • •