Based on QoS and Energy Efficiency Virtual Machines Consolidation Techniques in Cloud

Xinyue Sun[1], Yaqiu Liu[1], Wei Wei[2], Weipeng Jing[1,3], Chuanyu Zhao[3]

[1] College of Information and Computer Engineering, Northeast Forestry University, China
[2] College of Computer Science and Engineering, Xi'an University of Technology, China
[3] Heilongjiang Computing Center, China
sxy20054@foxmail.com, yaqiuLiu@gmail.com, taneo@126.com, nefujwp@163.com, chuanyuzhao@163.com

Abstract

Virtual machines (VMs) consolidation is one of the primary methods for improving the energy efficiency of cloud data centers. However, aggressive VMs consolidation methods may cause the hosts overload and produce massive inefficient VM migrations, so that host re-overload and even Quality of Service (QoS) degraded. Therefore, workloads are increasingly being deployed in virtualized cloud data centers to improve energy efficiency by using Virtual machines (VMs) consolidation. In this paper, we propose VMs selection algorithm and VMs placement algorithm for VMs consolidation to meet Quality of Service (QoS) for cloud applications while maximizing the energy efficiency and resource utilization of the data center. The VMs selection algorithm develops the relative CPU capacity gains to select the VMs which are selected to migrate from overloaded hosts. The VMs placement algorithm proposes an adaptive reservation CPU capacity for each physical host and allocates the selected VM to the hosts with idle CPU capacity. We experimented with real workloads, and results show that the algorithms can significantly reduce the energy consumption and improve QoS while decreasing the number of VM migrations.

Keywords: Cloud computing, Energy efficiency, Quality of service, Virtual machine consolidation

1 Introduction

As the scale of cloud computing continues to expand, there are many large-scale data centers have been established around the world [1]. However, the data centers can provide emission carbon and generate high power consumption which leads to global climate change, and the energy cost increased. Data collected for the Greenpeace have shown that the energy consumption of data centers only in the United States has already contributed about 91 billion Kilo Watt Hour (KWH), which even exceeds the electricity consumption of most countries in one year [2]. Besides, it is posing a severe threat to the environment that a significant number of carbon dioxides is emitted in the process of the data center power supply and cooling [3-6]. As a result, the energy-efficient resource management strategy has become a crucial part of overall cloud centers.

VM consolidation method can efficiently implement energy-saving management in a data center [7]. The method periodically detects the CPU usage of the physical host in the data center and migrates VMs in the low-utilization physical host to other hosts by using VM live migration technology [8]. In a data center, the VM consolidation method can efficiently improve resource utilization and reduce power consumption [9]. However, when a VM requests increased computing resources, the physical host may not be able to satisfy the VM's request, which results in service level agreement (SLA) violations [10]. Therefore, reducing the energy consumption with satisfactory SLA is a significant challenge for dynamic VM consolidation.

Recent studies [11-21] have shown that many VM consolidation methods focus on a trade-off between energy consumption and SLA violations. However, these methods can not satisfy the requirements of applications for continuous and stable computing resources [22]. Thus, it is essential to propose a new VM consolidation method so that the data center can maintain the low energy consumption status on the premise of providing more reliable and stable computing services.

There are two reasons why data centers cannot provide stable computing resources. First, when physical hosts are over-loaded, the hosts cannot provide sufficient computing resources for VMs. Second, when the VM during migration, the computing performance of the VM is decreased. Thus, we propose the MCLQBFD method to address these issues. This paper pays more attention to improve the service quality of cloud computing services. The main contributions of this paper are as follows:

- We propose a reserved CPU model that can reserve

*Corresponding Author: Wei Wei; E-mail: taneo@126.com
DOI: 10.3966/160792642019102006015

CPU resources for physical hosts based on historical CPU utilization. This model can prevent host overloading caused by cloud workloads fluctuations.

- We propose the minimization of computational loss (MCL) algorithm to reduce the computational performance loss due to VM migration, the algorithm evaluates the total loss of computing resources during VM migration and uses a greedy strategy to migrate the appropriate VM.
- Based on the reserved CPU model, we add the QoS aware best fit decreasing (QBFD) algorithm. This algorithm integrates VMs into fewer physical hosts without violating the reserved CPU model.

The remainder of this paper is organized as follows. The related works are discussed in Section 2. Section 3 introduces the system model and evaluation. Section 4 elucidates MCLQBFD method. The experiment results are given in Section 5. Section 6 makes conclusions and looks forward to the future.

2 Background

Recent work has shown that cloud computing has been widely used in all walks of life. However, the massive computing resources requirements aggravated the burden of cloud data centers, which increases energy consumption and degrade QoS [11-13]. Beloglazov et al. [7] designed a VM consolidation method to manage a cloud data center. This approach achieved a tradeoff between energy cost and QoS in the data center. They divided the process of VM consolidation into four phases: (1) determining when a physical host is overloaded; (2) determining when a physical host is underloaded; (3) selecting some VMs from the overloaded and underloaded hosts (4) allocating the selected VM to idle hosts. Their work reduced energy consumption due to VM consolidation. However, the proposed host overload detection algorithm was limited to heuristics, which leads to the result is the suboptimal solution. Therefore, Beloglazov and Buyya [12] proposed a novel method to solve the problem of the sub-optimal overloaded host. They optimized the host overload detection problem by maximizing the average interaction time under the specified QoS target based on the Markov chain model and used the Multisize Sliding Window workloads estimation technology to handle unknown non-stationary workloads. Corradi et al. [14] proposed a cloud platform management method to optimize VM consolidation from three perspectives of energy cost, computing resources, and network. Their experimental results showed the feasibility of VM consolidation in cloud data centers.

The above studies saved the energy cost of the data center using VM live migration. However, they have little focus on performance degradation and additional energy costs caused by VM migration. Xu et al. [15] proposed a lightweight interference-aware VM real-time migration strategy (iAware) that estimates the performance of VMs by the experience of benchmark workloads testing on the Xen cluster platform. This strategy decreased the number of VM migrations based on previous work. Tao et al. [16] established a BGM-BLA to optimize VM consolidation by considering energy consumption, VM communication, and VM migration. They divided the VM migration into two parts: (1) dividing the VMs into different groups; (2) determining the best way to allocate the groups into physical hosts. The method performed well regarding computational time, and the Pareto sets obtained. Ye et al. [17] presented an analysis-based VM consolidation framework, which minimizes the number of actively physical hosts while maintaining satisfactory performance for a variety of workloads. The management framework contained two modules: (1) consolidation planning module, which can minimize the number of active hosts according to a given set of workloads; (2) migration planning module, which can minimize the number of VM migrations by a polynomial time algorithm. The framework efficiently reduced the number of physical hosts in the data center while maintaining the high performance of the workloads. Shidik et al. [18] proposed a virtual machine selection model based on fuzzy Markov normal algorithm for dynamic virtual machine consolidation to improve the energy efficiency of cloud data centers. Their proposed fuzzy logic has been used to classify the attributes of VM candidates, and the Markov Normal Algorithm is used to determine which class of VMs should be migrated from an overloaded host. Shidik et al. [19] proposed a VM selection method based on K-means clustering technique and computational model Markov conventional algorithm (K-mMA). The purpose of the VM selection is to select the appropriate VM that should migrate from the overloaded physical machine and avoid oversubscribing the host, thereby improving the energy efficiency and quality of service (QOS) of the cloud data center.

The VM placement problem of VM consolidation is strictly NP-hard, the heuristic algorithm can be used to solve it. Therefore, Farahnakian et al. [20] proposed a distributed system architecture and online optimization meta-heuristic algorithm Ant Colony System (ACS) to perform dynamic VM integration. Their work achieved the lower energy consumption of cloud data centers while maintaining the required SLA. Based on the research, Liu et al. [21] proposed an OEMACS algorithm that combines order exchange and migration (OEM) local search techniques with the ACS algorithm. The algorithm allocates VMs from a global optimization perspective, placing more VMs in fewer active hosts. Zhang et al. [22] weighed the cost of network communication and the cost of VM migration, and then completed the integration of VMs by different

group intelligence algorithms (GA, ABC, PSO, and ACO). Li et al. [23] developed a Bayesian network estimation model (BNEM) for VM live migration based on cloud data centers. They proposed a hybrid Bayesian network-based VM consolidation (BN-VMC) method, which consists of three algorithms corresponding to different phases of VM consolidation. Zhou et al. [24] proposed a new algorithm called EEOM, which considers CPU and memory factors.

All the above works considered a tradeoff between energy consumption (host operating, VM migration, network communication) and QoS (VM computing performance). However, the applications require data centers to provide higher QoS of computing power. Melhem et al. [25] proposed an overload host detection algorithm based on the Markov prediction model and a VM placement algorithm based on the physical host that has received the migrated VM. Their work drastically reduced data center SLA violations, allowing data centers to support applications better. However, this method caused the data center's energy consumption rise sharply, which is contrary to the design goal of the VM consolidation method. Therefore, we propose a new method that aims to increase the energy efficiency of data centers while meeting the computing resource requirements of applications.

3 System Model and Evaluation

3.1 System Model

In this paper, the target system is a cloud data center that provides computing resources for Cloud applications. The data center has a large number of heterogeneous physical hosts which contains multicore CPU, memory, and network I/O [26-27]. The millions of instructions per second (MIPS) are units of CPU computing performance, and the storage system is network attached storage (NAS) [28]. As shown in Figure 1, the users first submit Cloud tasks to global manager and signs a service level agreement with cloud service providers. Then, the global manager sends the users' request to local managers. Finally, the local manager creates the VMs on physical hosts according to the request.

In a data center, $P=\{\mathrm{p}_1,p_2,...,p_i,...,p_N\}$ represents a set that consists of N physical hosts and $V_{\mathrm{i}}=\{v_1,v_2,...,v_j,...,v_m\}$ represents a set that consists of m VMs deployed in p_i.

v_j represents the jth VM, v_j^{mc} represents the max CPU computing capacity, v_j^{rc} is the current requested CPU computing resources, and v_j^{u} represents the current CPU utilization. The relationships of v_j^{mc}, v_j^{rc} and v_j^{u} is defined as follows:

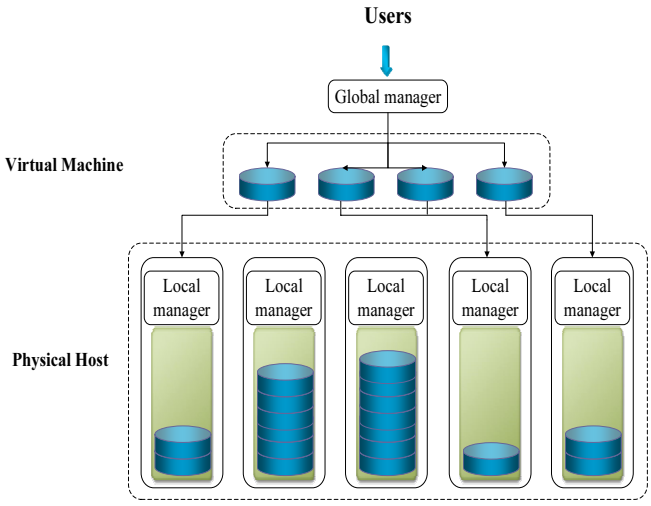

Figure 1. The mechanism of cloud data center

$$v_j^{rc}=v_j^{mc}\times v_j^{\mathrm{u}} \tag{1}$$

p_i represents the ith physical host, $\mathrm{p}_i^{\mathrm{rc}}$ is the requested CPU computing resources and can be calculated in (2), $\mathrm{p}_i^{\mathrm{mc}}$ is the max CPU computing capacity, $\mathrm{p}_i^{\mathrm{u}}$ is the current CPU utilization. The relationship of $\mathrm{p}_i^{\mathrm{rc}}$, $\mathrm{p}_i^{\mathrm{mc}}$ and $\mathrm{p}_i^{\mathrm{u}}$ are as follows.

$$\mathrm{p}_i^{\mathrm{rc}}=\sum_{\mathrm{v}_j\in V_i} v_j^{rc} \tag{2}$$

$$\mathrm{p}_i^{u}=\begin{cases}\dfrac{p_i^{rc}}{p_i^{mc}} & \text{if } p_i^{rc}<p_i^{mc}\\ 1 & \text{if } p_i^{rc}\geq p_i^{mc}\end{cases} \tag{3}$$

3.2 Power Model

The CPU, memory, disk storage, and network interfaces mostly determine the power consumption of physical hosts. However, the energy cost of CPU is the most significant part. Based on the studies [7, 29-30], the power of the host is linearly related to its CPU utilization, and the idle hosts still generate 70% of the max power cost. Therefore, Equation (4) represents the power model of a physical host.

$$E(p)=k\times p^{\max}+(1-k)\times p^{\max}\times p^{u} \tag{4}$$

Where $p^{\max}$ is the maximum power of physical host when CPU utilization is 100%; k is the ratio of the power consumption of the idle host to the fully loaded host and is set to 0.7 [31]. Thus, the total energy cost of a host is as follows.

$$EC=\int_{t_0}^{t_1} E(p^{u}(t))dt \tag{5}$$

3.3 Live Migration Cost

VM live migration technology allows VMs transfers between physical hosts without stalling. During migration, the average computing performance degradation of VM is equivalent to 10% of the CPU utilization [8]. Equation (6) expresses the migration time of VM. Thus, the live migration cost of VM can be calculated by (7).

$$T_j^m = \frac{v_j^{ram}}{p_i^{bw}} \tag{6}$$

$$\mathrm{v}_j^{du} = 0.1 \times \int_{t_0}^{t_0+T_j^m} v_j^u(t)dt \tag{7}$$

where T_j^m represents the jth VM's migration time, v_j^{ram} is the memory of the VM, p_i^{bw} represents available network bandwidth of the host, t_0 is the start time of the VM migration, v_j^{du} represents CPU utilization loss by VM migration, $v_j^u(t)$ is the VM's CPU utilization at time t.

3.4 QoS Evaluation

In this paper, QoS reflects the stability of the cloud data center to provide computing services for Cloud applications when the Cloud applications fail to obtain sufficient computing resources from the data center, which causes SLA violations and degradation of QoS. Therefore, we use the degradation of computational performance as a criterion to evaluate SLA violations by research [12]. It consists of two parts:

(1) SLATAH indicates the ratio of physical host overload time to physical host runtime and is defined in (8).

$$SLATAH = \frac{1}{N}\sum_{\mathrm{i}=1}^{N}\frac{T_{s_i}}{T_{a_i}} \tag{8}$$

Where T_{s_i} is the time of SLA violations when p_i is overload, T_{a_i} is the total operation time of p_i.

(2) PDM represents the ratio of the loss of computing resources during VM migration to the request computing resources of all VMs and is defined in (9).

$$PDM = \frac{1}{M}\sum_{\mathrm{j}=1}^{M}\frac{C_{d_j}}{C_{r_j}} \tag{9}$$

Where C_{d_j} is the loss of computing resources during the migration of v_j, C_{r_j} represents the overall requested computing resources of v_j. SLAV is regarded as a combined metric to measure the QoS of the data center and is calculated as follows:

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

4 Mclqbfd Method

The VM consolidation method divided into four parts: (1) host overload detection; (2) host underload detection; (3) VM selection; (4) VM placement. The goal of this paper is to ensure whether the data center processes cloud data efficiently. Therefore, we have improved and updated the traditional method of VM consolidation:

(1) We propose a reserve CPU model, which is applied to solve the host re-overload caused by host overload detection algorithm.

(2) To better satisfy the computing power requirements of the Cloud application, we proposed the MCL algorithm to reduce the computational performances loss in the process of VM consolidation.

(3) For VM placement, we proposed the QBFD algorithm that combines the reserve CPU model in (1) with the BFD [32] algorithm. The QBFD can prevent degradation of computing performance due to workload fluctuations.

4.1 Reserve CPU Model

Cloud applications have high requirements for data processing timeliness. In practice, the virtual machine's workload is unpredictable and fluctuating, which causes the physical host to be overloaded twice when the workload fluctuates after the virtual machine is placed. To prevent avoid secondary overload of the physical host and the Cloud data cannot be processed efficiently due to physical host overload, we reserve a part of the CPU resources for physical hosts to cope with possible load fluctuations. The reserve CPU space should satisfy three conditions: (1) The reserved space should be dynamically adaptive, and the size depends on historical CPU utilization fluctuations; (2) The reserved space should not be affected by abnormal historical data; (3) The closer the data is to the current time, the greater the impact on the reserved CPU space.

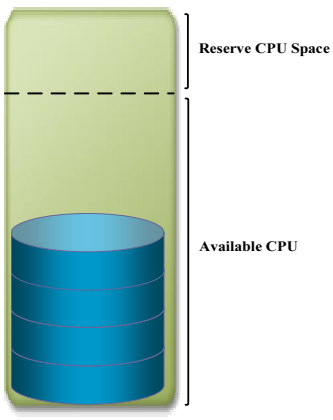

Figure 2. The CPU model of a physical host

Figure 2 shows the CPU model of a physical host. To achieve the above two conditions, we design the reserve CPU space through the data of the historical CPU utilization of the physical host, which is acquired by the data center at equal intervals.

For the first two conditions, we implement the following method. For historical CPU utilization data set $\{x_t\}(t=1,2,...)$, We use a sliding window to collect and use it and the size of window is *k*. When the total amount of historical data *k* is smaller than the window size, all the historical data is processed and the collected data set is $\{x_1, x_2, ..., x_k\}$; when the total amount *k* of the historical data is larger than the window size, the latest historical data within the window size is adopted and the collected data set is $\{x_{end-k+1}, ..., x_{end-1}, x_{end}\}$. Then let $end = k$, so that the collected data set is integrated into a new data set $\{x_1, x_2, ..., x_k\}$.

To avoid being affected by abnormal historical data, we pre-process the historical data, identify the outliers in the data through the criterion [33], and then replace the outliers with the mean.

$$\bar{x} = \frac{1}{k}\sum_{i=1}^{k} x_i \tag{11}$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{k}(x_i - \bar{x})^2} \tag{12}$$

$$x_i = \begin{cases} x_i & \text{if } \bar{x} - 3\sigma < x < \bar{x} + 3\sigma \\ \bar{x} & \text{other} \end{cases} \tag{13}$$

$$b_i = | x_i - \bar{x} | \tag{14}$$

Where x_i represents the *i*th CPU utilization in the history data, $\bar{x}$ is the mean of the data, σ is the data's standard deviation, b_i is the absolute value of x_i deviation. In a data center, the value of b_i changes as the historical data of the collected physical host constantly changes. Therefore, we regard $\{b_i\}$ as the baseline to describe the range of data center workload fluctuations, which can be updated over time.

For the last condition, we apply the sum of weighted deviations to estimate the size of the reserve CPU. The formula for data pre-processing is as follows.

$$ft = (1-\lambda)\sum_{i=1}^{k} \lambda^{i-1} b_{k-i+1} \tag{15}$$

Where ft is reserve CPU space, $\lambda (\in (0,1))$ is weight coefficients of history data. In (15), the closer the data is to the current time, the greater the effect on ft under a given parameter λ, which is decided by section 5. Further, $(1-\lambda)\sum_{i=1}^{k} \lambda^{i-1} = 1$ is always constant when *k* approaches infinity, which is one reason for designing the equation.

In this paper, we regard the local regression (LR) as host overload detection based on research [7] and assume the maximum CPU utilization of the physical is $1 - ft$.

4.2 VM Selection Algorithm

To improve the processing speed of cloud applications, we must minimize the loss of computational performance during VM migration. Therefore, we propose MCL algorithm to achieve two goals:

(1) Makes the CPU usage of the migrated physical host less than $1 - ft$, which represents the host is no overload.

(2) Reduce the computational computing performance loss caused by VM migration, which ensures that the Cloud application runs correctly.

When the VMs of the host are migrated to other hosts, the computational performance of the VMs are degraded during the migration. The loss of computing resources is defined as follows.

$$l_j = v_j^{du} \times v_j^{mc} \tag{16}$$

Where v_j is a VM that is migrated from the overloaded host, l_j represents the loss of CPU capacity during the migration of v_j.

Algorithm 1 shows the VM selection for overloaded hosts. This algorithm uses a greedy strategy to select the VM with the least performance degradation in the migration process. First, the algorithm obtains the performance loss of VM migration in the current physical host. Then, the VMs are sorted in ascending order based on performance loss. Finally, the VMs are continuously migrated in order until the physical host is no overload. If m is the number of the VMs and n is the number of overloaded hosts, the time complexity of Algorithm 1 is O (n*m).

Algorithm 1. Minimization of Computational Loss (MCL)

```
1.  Input: overloadedHostList
2.  Output: selectedVmList
3.  selectedVmList ← NULL;
4.  foreach host in overloadedHostList do
5.      vmList ← host.getVmList();
6.      hUtil ← host.getUtil();
7.      foreach vm in vmList do
8.          Calculate l for vm by using (16);
9.          vm.updataLoss(l);
10.     end for
11.     Sort vmList by increasing order of l
12.     foreach vm in vmList do
```

```
13.          if vm.getUtil() > hUtil - host.getFt()
14.              selectedVmList.add(vm);
15.              host.removeVm(vm);
16.          else
17.              break;
18.          end if
19.       end for
20.  end for
21.  return seletedVmList;
```

4.3 VM Placement Algorithm

Once selecting VMs from overloaded hosts, the next step is to allocate the VMs to idle hosts that have available CPU utilization. We propose a QBFD algorithm to reduce the possibility of host re-overload while increasing the resource utilization of the host to keep the data center running at low power. Thus, it is necessary to evaluate the CPU usage increase of the host after the VM migration.

$$\Delta p_{i,j}^{u} = \frac{v_j^u \times v_j^{rc}}{p_i^{mc}} \tag{17}$$

Where $\Delta p_{i,j}^{u}$ is increased CPU usage after p_i receives v_j. When the VM migration is complete, the remaining available CPU utilization of the host is written as follow:

$$\varphi_{i,j} = 1 - ft_i - \Delta p_{i,j}^{u} \tag{18}$$

Where $\varphi_{i,j}$ is remaining available CPU usage of p_i after p_i receives v_j, ft_i is the reserve CPU space of p_i. When placing the VMs, the QBFD algorithm minimizes the remaining available CPU resources while maintaining reserve CPU resources.

The pseudo-code is the QBFD algorithm that allocates the selected VMs. This algorithm selects the best physical host for each selected VM. The best host satisfies the condition: (1) the remainder available CPU utilization of the host is more significant than zero after the VM placement; (2) the remaining CPU utilization of the host is the smallest. If m is the number of the VMs and n is the number of overloaded hosts, the time complexity of Algorithm 2 is O (n*m).

Algorithm 2. QoS Aware Best Fit Decreasing (QBFD)

```
1.  Input: hostList, migrationList
2.  foreach vm in migrationList do
3.      rMin ← MAX
4.      allocatedHost ← NULL
5.      foreach host in hostList do
6.          r ← Calculate using equation (18)
7.          if r > 0 && r < rMin then
8.              rMin ← r
9.              allocatedHost ← host
10.         end if
11.     end for
12.     if allocatedHost ≠ NULL then
13.         allocate vm to allocatedHost
14.     end if
15. end for
```

5 Experimental Environment and Results

5.1 Experimental Setup

To evaluate the MCLQBFD algorithm, this paper simulated a data center by using CloudSim toolkit [34]. CloudSim is a cloud simulation software from the CloudBus project team that supports cloud computing infrastructure and application modeling, simulation and experimentation. The data center is consisted of 400 HP ProLiant ML110 G4 (Intel Xeon 3040, 2 cores * 1.86GHz) and 400 HP ProLiant ML110 G5 (Intel Xeon 3075, 2 cores * 2.26GHz) physical hosts. According to SPECpower, the maximum power consumption of the two types of physical hosts is 117W and 135W respectively.

In the data center, we simulated four different VMs whose parameters came from Amazon EC2. Table 1 shows the four VM types in the experiment.

Table 1. Types of Amazon EC2

VM Types	MIPS	Memory (GB)
High-CPU Medium Instance	2500	0.85
Extra Large Instance	2000	3.75
Small Instance	1000	1.70
Micro Instance	500	0.613

We used PlanetLab trace [35] from a real infrastructure. The PlanetLab trace contains the CPU utilization of physical hosts by measured every 5 minutes from more than the thousand VMs for ten days. Table 2 shows the PlanetLab trace information.

Table 2. PlanetLab trace information

Date	Number of VMs	Mean(%)	St.dev.(%)
2011/03/03	1052	12.31%	17.09%
2011/03/06	898	11.44%	16.83%
2011/03/09	1061	10.70%	15.57%
2011/03/22	1516	9.26%	12.78%
2011/03/25	1078	10.56%	14.14%
2011/04/03	1463	12.39%	16.55%
2011/04/09	1358	11.12%	15.09%
2011/04/11	1233	11.56%	15.07%
2011/04/12	1054	11.54%	15.15%
2011/04/20	1033	10.43%	15.21%

5.1 Experimental Setup

Evaluate the performance of the VM consolidation approach in terms of six different performance metrics, which are energy consumption (EC), SLA violations (SLAV), SLA time per active host (SLATAH), performance degradation due to migrations (PDM), the number of VM migrations (VMM), and ESV that is combined evaluations of energy consumption and SLAV [4]. ESV is defined as the following:

$$ESV = EC \times SLAV \tag{19}$$

5.2 Experiment and Analysis

In the section, we conducted two experiments:

In Experiment 1, we determined the weight coefficients of the reserve CPU model.

In Experiment 2, to verify the feasibility of the MCLQBFD method, we used the combination strategy of four host overloaded detection algorithms (THR, MAD, IQR, and LR) and three VM selection algorithms (MMT, MC, and RS) as benchmark methods. In the same experimental environment, we compare the MCLQBFD method with the EEOM [21] method, the RUA [36] method and the benchmark algorithm.

(1) Determining weight coefficient λ

The value of the weight coefficient λ in the reserve CPU model must first be determined. The weight coefficient determines the impact of the actual CPU utilization on a reserve CPU model. In the experiment, we used the real workload from Table 2 (date: 2011/03/03) to compare the energy consumption (EC), SLAV, and ESV of the data center under different weight coefficient λ (λ changes from 0.0 to 1.0, with a step size of 0.05). Figure 3, Figure 4 and Figure 5 respectively show energy consumption, SLAV, and ESV with different weight coefficient λ in the data center.

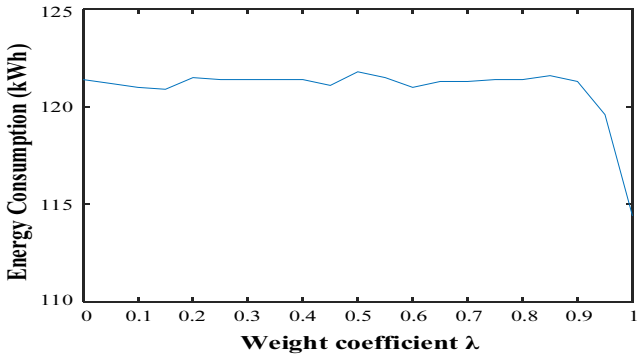

Figure 3. Energy consumption under different weight coefficient

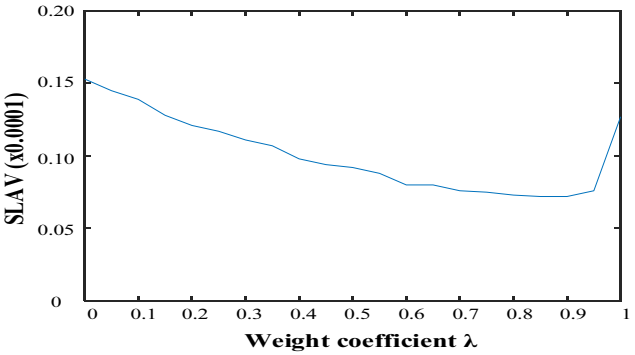

Figure 4. SLAV under different weight coefficient

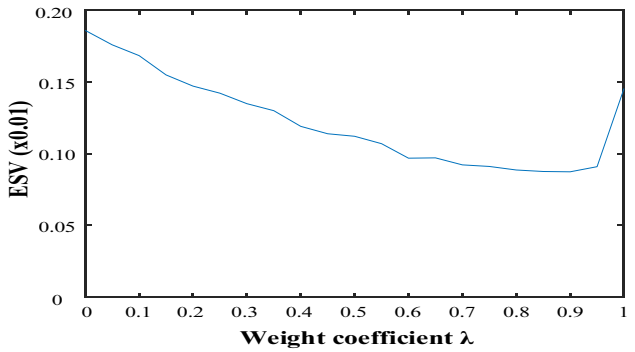

Figure 5. ESV under different weight coefficient

The curve in Figure 1 shows that the energy consumption of the data center is high during weight coefficient λ is between 0 and 0.8, and the energy cost is rapidly reduced during the process of raising the weight coefficient λ from 0.8 to 1.0. This is because when the weight coefficient λ is between 0.8 and 1.0, the reserved CPU model is gradually less affected by the latest historical data, which allows the physical host to release more available computing resources, thereby reducing the energy consumption of the data center. Figure 2 shows the SLA violations of the data center are falls when the weight coefficient λ rises from 0 to 0.9, and the SLA violations rise rapidly during weight coefficient from 0.9 to 1.0. The main reason is that during the period of 0.9 to 1.0, the reserved CPU becomes smaller, the probability of physical host overload is greatly increased and new virtual machine migrations are generated, which make the SLA violation increase rapidly. Combining the above two figures, we identify that the energy consumption decreases and the SLA violations increase when the weight coefficient λ rises from 0.9 to 1.0. The curve in Figure 5 represents the ESV, and we can identify that the data center has the lowest value of ESV when λ=0.9. This is because ESV is a comprehensive measure of energy consumption and SLA violations. As can be seen from Figure 3 and Figure 4, the data center energy consumption and SLAV are both at a low level when λ=0.9. Therefore, the weight coefficient λ is set to 0.9 in this paper.

(2) Comparing the performance between different methods

Table 3 provides the comparison results of different methods in terms of the six performance evaluations. Each value in Table 3 represents the average performance per day after the data center is running for ten days, and the numbers following the name of each method are the parameters of the corresponding method.

Table 3. Six performance evaluations of different VM consolidation methods

Method	EC (kWh)	SLATAH (%)	PDM (%)	SLAV (x0.0001)	ESV (x0.01)	VMM
MCLQBFD	117.33	3.8980	0.0240	0.0975	0.1115	10211
RUA	123.39	3.7600	0.0310	0.1201	0.1438	12690
EEOM	127.62	3.9713	0.0290	0.1152	0.1469	11713
THR-MMT-0.8	188.45	5.0380	0.0680	0.3371	0.6256	26601
THR-MC-0.8	179.33	6.9110	0.1000	0.6989	1.2410	23961
THR-RS-0.8	180.80	6.9540	0.1020	0.6991	1.2507	24217
MAD-MMT-2.5	183.43	5.0540	0.0650	0.3348	0.6061	26305
MAD-MC-2.5	173.74	7.0390	0.1010	0.7111	1.2252	23420
MAD-RS-2.5	174.96	7.0820	0.1010	0.7111	1.2338	23736
IQR-MMT-1.5	187.50	5.0230	0.0650	0.3288	0.6071	26497
IQR-MC-1.5	177.66	6.9160	0.0980	0.6805	1.1976	23394
IQR-RS-1.5	179.06	6.9730	0.0970	0.6793	1.2061	23796
LR-MMT-1.2	161.81	6.2130	0.0800	0.4974	0.7951	28175
LR-MC-1.2	148.47	7.4840	0.1020	0.7609	1.1185	23931
LR-RS-1.2	147.61	7.6220	0.1030	0.7781	1.1381	23779

Energy consumption is used to compare the power costs of different methods in the data center. As shown in Table 3, MCLQBFD method always outperforms other methods in terms of energy consumption, and the value of energy consumption is 117.33 kWh. The QBFD method maintains reserved CPU space while increasing resource utilization, shutting down many physical hosts to reduce the power consumption. The result shows that even if the MCLQBFD method reserves a part of CPU utilization to prevent host overload, it can still guarantee high energy efficiency in the data center. The benchmark method's lowest energy consumption has reached 147.61kWh that is the highest energy consumption in all methods. RUA method consumes less energy than the benchmark method and EEOM. The reason is RUA proposes the PA, which is a low-workload detection parameter that reduces the data center by continuously reallocating virtual machines. In contrast, compared the RUA method, EEOM method, best performing benchmark method, the energy efficiency of the MCLQBFD method is improved by 4.91%, 8.06%, 20.51%.

SLAV reflects the QoS provided by the data center for computing resources. SLAV consists of two parts, one is SLATAH, which indicates that losing the computing resources due to physical host overload, and the other part is PDM, which indicates that wasting computational resources due to VM migration. From Table 3, the SLATAH value of the MCLQBFD method is 3.898%, the PDM value is 0.024%, and the SLAV value is 0.0975^{-4}, which is lower than other methods. There are two main reasons for this: First, the MCLQBFD algorithm is always able to maintain the normal working load state of the physical host for a long time, which greatly reduces the frequency of overload of the physical host. Second, it reduces a large number of virtual machine migrations, thereby reducing the computational resource loss caused by virtual machine migration. From the SLAV index, the MCLQBFD method is 18.82%, 15.36%, 70.35% lower than the RUA method, EEOM method, and IQR-MMT-1.5, which indicates that our method can effectively improve the QoS of the cloud data center.

ESV represents the overall performance of the data center in both energy consumption and SLAV. As shown in Table 3, the ESV value of the MCLQBFD algorithm is much lower than the other methods, because the method outperforms other methods in term of both energy consumption and SLAV. In general, compared with the RUA method, EEOM method, and benchmark algorithm, the proposed method is reduced by 22.46%, 24.10%, and 81.60%.

VMM represents the number of VM migrations in the data center. The VMM of the MCLQBFD method is 10211, which is higher than the 12690 VM migrations of the RUA and 11713 VM migrations of the EEOM. ECLQBFD has the least frequent migration of the benchmark methods. From Table 3, it can be seen that there is little host over-load in the MCLQBFD method, which makes it unnecessary for the physical host to migrate many VMs to make the work-load of the host return to normal. Therefore, the proposed method is reduced by 19.54%, 12.82%, and 56.35% compared with RUA, EEOM, and benchmark algorithms.

The experimental results show that the MCLQBFD method outperforms other methods in five performance indexes, and only slightly inferior to the RUA method in VMM. In particular, MCLQBFD method is very prominent on SLAV. Therefore, the MCLQBFD

method can more effectively satisfy the massive computing resources required for cloud applications.

6 Conclusion and Future Work

With the expansion of cloud computing applications, the demand for data processing and application storage has grown excessively. Cloud computing services provide customers with a large number of computing resources and storage space, which effectively promotes the further development of other industries. This paper proposes a method called MCLQBFD to improve the quality of data center computing services with lower energy consumption. In this work, we proposed a CPU fault-tolerance model to prevent physical hosts overload. Then, we proposed the MCL algorithm to reduce the computational performance loss caused due to the VM migration. Finally, we designed the QBFD algorithm based on the CPU fault-tolerance model to achieve the data center low-power operation while ensuring the quality of the data center.

In the future, we plan to combine reinforcement learning with data center management and conduct experiments in a real cloud platform.

Acknowledgments

This job is supported by the National key R&D Program of China under Grant NO.2018YFB0203901.

This work is also supported by the Key Research and Development Program of Shaanxi Province (No. 2018ZDXM-GY-036).

This job is supported by the National Natural Science Foundation of China under Grant 31770768 and the Natural Science Foundation of Hei Longjiang Province of China under Grant F2017001.

This work is supported by the China Postdoctoral Science Foundation 2017M611407.

References

[1] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, J. Rodriguez, Massive Internet of Things for Industrial Applications: Addressing Wireless Iiot Connectivity Challenges and Ecosystem Fragmentation, *IEEE Industrial Electronics Magazine*, Vol. 11, No. 1, pp. 28-33, March, 2017.

[2] J. G. Koomey, Growth in Data Center Electricity Use 2005 to 2010, *A Report by Analytics Press, Completed at the Request of The New York Times*, August, 2011.

[3] S. Sharma, G. Sharma, A Review on Secure and Energy Efficient Approaches for Green Computing, *International Journal of Computer Applications*, Vol. 138, No. 11, pp. 25-32, March, 2016.

[4] W. Wei, H. M. Srivastava, Y. Zhang, L. Wang, P. Shen, J. Zhang, A Local Fractional Integral Inequality on Fractal Space Analogous to Anderson's Inequality, *Abstract and Applied Analysis*, Vol. 2014, Article ID 797561, pp. 1-7, June, 2014.

[5] W. Wei, X. L. Yang, B. Zhou, J. Feng, P. Y. Shen, Combined Energy Minimization for Image Reconstruction from Few Views, *Mathematical Problems in Engineering*, Vol. 2012, Article ID 154630, pp. 1-15, September, 2012.

[6] M. Poess, R. O. Nambiar, Energy Cost, The Key Challenge of Today'S Data Centers: A Power Consumption Analysis of TPC-C Results, *Proceedings of the VLDB Endowment*, Vol. 1, No. 2, pp. 1229-1240, August, 2008.

[7] A. Beloglazov, 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, September, 2012.

[8] W. Voorsluys, J. Broberg, S. Venugopal, R. Buyya, Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation, *IEEE International Conference on Cloud Computing*, Beijing, China, 2009, pp. 254-265.

[9] Q. Zhang, M. F. Zhani, R. Boutaba, J. L. Hellerstein, Dynamic Heterogeneity-aware Resource Provisioning in The Cloud, *IEEE Transactions on Cloud Computing*, Vol. 2, No. 1, pp. 14-28, January-March, 2014.

[10] S. K. Garg, A. N. Toosi, S. K. Gopalaiyengar, R. Buyya, Sla-based Virtual Machine Management for Heterogeneous Workloads in a Cloud Datacenter, *Journal of Network and Computer Applications*, Vol. 45, pp. 108-120, October, 2014.

[11] C. Zhu, J. J. P. C. Rodrigues, V. C. M. Leung, L. Shu, L. T. Yang, Trust-based communication for the Industrial Internet of Things, *IEEE Communications Magazine*, Vol. 56, No. 2, pp. 16-22, February, 2018.

[12] A. Beloglazov, R. Buyya, Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers Under Quality of Service Constraints, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 7, pp. 1366-1379, July, 2013.

[13] G. F. Shidik, N. S. Sulistyowati, M. B. W. Tirta, Evaluation of Cluster K-Means as VM Selection in Dynamic VM Consolidation, *2016 22nd Asia-Pacific Conference on Communications (APCC)*, Yogyakarta, Indonesia, 2016, pp. 124-128.

[14] A. Corradi, M. Fanelli, L. Foschini, VM Consolidation: A Real Case based on OpenStack Cloud, *Future Generation Computer Systems*, Vol. 32, pp. 118-127, March, 2014.

[15] F. Xu, F. Liu, L. Liu, H. Jin, B. Li, B. Li, Iaware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud, *IEEE Transactions on Computers*, Vol. 63, No. 12, pp. 3012-3025, December, 2014.

[16] F. Tao, C. Li, T. W. Liao, Y. Laili, BGM-BLA: A New Algorithm for Dynamic Migration of Virtual Machines in Cloud Computing, *IEEE Transactions on Services Computing*, Vol. 9, No. 6, pp. 910-925, November-December, 2016.

[17] K. Ye, Z. Wu, C. Wang, B. B. Zhou, W. Si, X. Jiang, A. Y. Zomaya, Profiling-based Workload Consolidation and Migration in Virtualized Data Centers, *IEEE Transactions on*

Parallel and Distributed Systems, Vol. 26, No. 3, pp. 878-890, March, 2015.

[18] G. F. Shidik, A. Azhari, K. Mustofa, Improvement of Energy Efficiency at Cloud Data Center Based on Fuzzy Markov Normal Algorithm VM Selection in Dynamic VM Consolidation, *International Review on Computers and Software (IRECOS)*, Vol. 11, No. 6, pp. 511-520, June, 2016.

[19] G. F. Shidik, A. Azhari, K. Mustofa, K-mMA VM Selection in Dynamic VM Consolidation for Improving Energy Efficiency at Cloud Data Centre, *International Journal of Communication Networks and Distributed Systems*, Vol. 21, No. 2, pp. 202-219, August, 2018.

[20] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, 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, March-April, 2015.

[21] X. F. Liu, Z. H. Zhan, J. D. Deng, Y. Li, T. Gu, J. Zhang, An Energy Efficient Ant Colony System for Virtual Machine Placement in Cloud Computing, *IEEE Transactions on Evolutionary Computation*, Vol. 22, No. 1, pp. 113-128, February, 2018.

[22] W. Zhang, S. Han, H. He, H. Chen, Network-aware Virtual Machine Migration in an Overcommitted Cloud, *Future Generation Computer Systems*, Vol. 76, pp. 428-442, November, 2017.

[23] Z. Li, C. Yan, X. Yu, N. Yu, Bayesian Network-based Virtual Machines Consolidation Method, *Future Generation Computer Systems*, Vol. 69, pp. 75-87, April, 2017.

[24] Z. Zhou, J. Yu, F. Li, F. Yang, Virtual Machine Migration Algorithm for Energy Efficiency Optimization in Cloud Computing, *Concurrency and Computation: Practice and Experience*, Vol. 30, No. 24, e4942, pp. 1-10, December, 2018.

[25] S. B. Melhem, A. Agarwal, N. Goel, M. Zaman, Markov Prediction Model for Host Load Detection and VM Placement in Live Migration, *IEEE Access*, Vol. 6, pp. 7190-7205, December, 2017.

[26] K. Wang, Y. Wang, Y. Sun, S. Guo, J. Wu, Green Industrial Internet of Things Architecture: An Energy-efficient perspective, *IEEE Communications Magazine*, Vol. 54, No. 12, pp. 48-54, December, 2016.

[27] W. Wei, Y. Qiang, J. Zhang, A Bijection between Lattice-Valued Filters and Lattice-Valued Congruences in Residuated Lattices, *Mathematical Problems in Engineering*, Vol. 2013, Article ID 908623, pp. 1-6, July, 2013.

[28] P. Zheng, Y. Qi, Y. Zhou, P. Chen, J. Zhan, M. R. Lyu, An Automatic Framework for Detecting and Characterizing Performance Degradation of Software Systems, *IEEE Transactions on Reliability*, Vol. 63, No. 4, pp. 927-943, December, 2014.

[29] P. Wang, Y. Qi, X. Liu, Power-Aware Optimization for Heterogeneous Multi-tier Clusters, *Journal of Parallel and Distributed Computing*, Vol. 74, No. 1, pp. 2005-2015, January, 2014.

[30] Q. Ke, J. Zhang, H. Song, Y. Wan, Big Data Analytics Enabled by Feature Extraction Based on Partial Independence, *Neurocomputing*, Vol. 288, pp. 3-10, May, 2018.

[31] L. A. Barroso, U. Hölzle, The Case for Energy-Proportional Computing, *Computer*, Vol. 40, No. 12, pp. 33-37, December, 2007.

[32] M. Yue, A Simple Proof of The Inequality FFD (L) ≤ 11/9 OPT (L)+ 1, ∀ L for the FFD bin-packing Algorithm, *Acta Mathematicae Applicatae Sinica*, Vol. 7, No. 4, pp. 321-331, October, 1991.

[33] M. Zhang, H. Yuan, The PauTa Criterion and Rejecting the Abnormal Value, *Journal of Zhengzhou University of Technology*, Vol. 18, No. 1, pp. 84-88, March, 1997.

[34] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, R. Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, *Software: Practice and experience*, Vol. 41, No. 1, pp. 23-50, January, 2011.

[35] K. Park, V. S. Pai, CoMon: A Mostly-scalable Monitoring System for PlanetLab, *ACM SIGOPS Operating Systems Review*, Vol. 40, No. 1, pp. 65-74, January, 2006.

[36] G. Han, W. Que, G. Jia, L. Shu, An Efficient Virtual Machine Consolidation Scheme for Multimedia Cloud Computing, *Sensors*, Vol. 16, No. 2, pp. 246, February, 2016.

Biographies

Xinyue Sun received the B.S. degree from North University of China. He is currently a master candidate of computer architecture in Northeast Forestry University. His current research interests include cloud computing, green computing.

Yaqiu Liu received the Ph.D. degree from Harbin Institute of Technology University. Currently, he is a Professor at College of Information and Computer Engineering of Northeast Forestry University. His current research interests include process control, distributed computing, cloud computing, intelligent control, soft computing and model reconstruction.

Wei Wei received the Ph.D. and M.S. degrees from Xi'an Jiaotong University in 2011and 2005, respectively. Currently he is an Associate Professor at School of Computer Science and Engineering of Xi'an University of Technology. His academic interests in the following areas: internet of things, big data, cloud computing, image processing. He is a senior member of the IEEE.

Weipeng Jing received the Ph.D. degree from Harbin Institute of Technology University. Currently, he is an Associate Professor at College of Information and Computer Engineering of Northeast Forestry University. His research interests and expertise include modeling and scheduling for distributed computing systems, system reliability estimation, fault-tolerant computing and cloud computing. He is a member of the IEEE.

Chuanyu Zhao He is a senior engineer of Heilongjiang computing center. He has presided over a number of national scientific research projects and published more than 50 papers.