Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090331

Optimized Resource Scheduling Mechanism in Virtual Cluster System

Yunfa Li and Sheng Xiong*

School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China

Received: 1 Aug. 2014, Revised: 2 Nov. 2014, Accepted: 3 Nov. 2014
Published online: 1 May 2015

Abstract: With the growth of the number of users and the frequency of the users using virtual cluster system, the scheduling and the migration of system resources are becoming more and more frequent. In this situation, the energy consumption of system is becoming more and more serious. Therefore, it has become an important question that how to reduce energy consumption in virtual cluster system. In order to resolve this problem, we propose an optimized resource scheduling mechanism in virtual cluster system. In the optimized resource scheduling mechanism, we first propose an optimal resource scheduling model for reducing energy consumption. Then, we present a novel grouping genetic algorithm with fuzzy multi-objective evaluation and use the algorithm to resolve the optimal resource scheduling model. Based on the basis of the model and the algorithm, we design a resource scheduling management style. All these components constitute the optimized resource scheduling mechanism in virtual cluster system. In order to justify the feasibility and availability of the optimized resource scheduling mechanism, a series of experiments have been done. The results show that it is feasible to schedule system resources and reduce energy consumption in virtual cluster system.

Keywords: resource scheduling, management mechanism, grouping genetic algorithm, energy consumption, virtual cluster system

1 Introduction

Today, more systems owners are realizing their hardware which is being under-utilized, and virtualization is becoming a common solution. By using the virtualization technology, a computer system can aggregate all kinds of data resources, software resources and hardware resources and utilize those resources to provide service for different tasks. Moreover, the virtualization technology can separate hardware and software management and provide useful features including performance isolation[1] [2] , server consolidation and live migration [3] [4]. In addition, the virtualization technology can also provide secure and portable environments for these modern computing systems [5]. Therefore, the computing theorem and model that these virtual technology embodies are applied widely.

In fact, a virtual machine (VM) is a logical machine and has almost the same architecture as a real host machine. A virtual machine system can run multiple virtual machines. And each virtual machine includes an operating system. In addition, each virtual machine allows users to create,copy, save (checkpoint), read and modify, share, migrate and roll back the execution state of machine with all the cases of manipulating a file. The flexibility of system can provide significant value for users and administrator [6].

With the improvement of the application requirements of users, more and more hardware and software resources need to be integrated into a computer system. In such a case, the existing technology can't be satisfied with the requirement of the improvement of application. Thus, people begin to explore a new system structure for various applications about virtual machine.

Because the virtualization technology can carve some individual physical machines into multiple virtual containers, people begin to build some virtual cluster systems for various applications. However, the resource scheduling and the resource migration are becoming more and more frequent with the growth of the number of users and the frequency of the users using virtual cluster system. In this situation, the energy consumption of system will increase. However, the growth of energy consumption will generate a lot of heat in virtual cluster system and affect the utilization ratio and the service performance of system resources. Thus, there are two inconsistent factors between how to expand the scale of

* Corresponding author e-mail: yunfali@hdu.edu.cn

users and reduce the energy consumption of system. Therefore, it is urgent to design and implement the resource scheduling mechanism that reduces energy consumption.

In order to resolve this problem, we propose an optimized resource scheduling mechanism in virtual cluster systems. In the optimized resource scheduling mechanism, we first propose an optimal resource scheduling model for reducing energy consumption. Then, we present a novel grouping genetic algorithm with fuzzy multi-objective evaluation and use the algorithm to resolve the optimal resource scheduling model. Based on the basis of the model and the algorithm, we design a resource scheduling management style. All these components constitute the optimized resource scheduling mechanism in virtual cluster system.

The rest of this paper is organized as follows: we discuss these related works in section 2. In section 3, we propose an optimized resource scheduling mechanism in virtual cluster system. In section 4, a series of experiments are done and these results are analyzed. Finally, conclusions are drawn in section 5.

2 Related Works

With the development of virtual machine system, the scheduling management of resource and the energy consumption of system begin to be widely concerned. In order to ensure the quality of service of system resources, people presented a lot of algorithms for scheduling system resources and a lot of methods for balancing or reducing energy consumption. A lot of great processes have been made in these aspects. These great processes can be simply shown as follows.

In [7], Wang et al. focused on the relationship between input and output of their resource allocation control system, and built a simple predictive model to control resource allocation.

The borrowed virtual time (BVT) scheduling algorithm is proposed by Duda et al [8]. The essential of this algorithm is fair-share scheduler based on the concept of virtual time, dispatching the runnable virtual machine (VM) with the smallest virtual time first. Moreover, the algorithm provides low-latency support for real-time and interactive applications by allowing latency sensitive clients to warp back in virtual time to gain scheduling priority. The client effectively borrows virtual time from its future CPU allocation.

Scheduling algorithm is presented by Govindan et al [9]. In this algorithm, each domain specifies its CPU requirements. After all runnable domains receive their CPU share, SEDF will distribute this slack time fairly manner. In fact, the time granularity in the definition of the period impacts scheduler fairness.

The Credit Scheduling algorithm is described in [10]. It is Xen’s latest proportional share scheduler featuring automatic load balancing of virtual CPUs across physical CPUs on an SMP host. Before a CPU goes idle, it will consider other CPUs in order to find any runnable virtual CPU (VCPU). This approach guarantees that no CPU idles when there is runnable work in the system.

In [11], Seelam et al. presented a novel virtual I/O scheduler (VIOS) that provides absolute performance virtualization by being fair in sharing I/O system resources among operating systems and their applications, and provides performance isolation in the face of variations in the characteristics of I/O streams. In the scheduler, the VIOS controls the coarse-grain allocation of disk time to the different operating system instances and the output scheduler may determine the fine-grain interleaving of requests from the corresponding operating systems to the storage system.

In [12], Park et al. proposed an optimization model based on a linear programming in order to effectively manage the capacities and increase resource utilization of virtual machine. In the optimization model, an automated strategy for virtual machine migration is considered in self-managing virtualized server environments. By using the automated strategy, the system can effectively determine which virtual machines should be migrated.

In [13], Shi et al. proposed a model called the prioritized Chinese wall model to reduce the risk of covert flows in VM systems. By using the prioritized Chinese wall model, the virtual machine system can effectively reduce the risk of covert flows and enhance the flexibility of system management.

In [14], Kang et al. proposed a virtual machine-aware proportional share queuing scheduler, VM-PSQ, in server virtualization environment. To do that, they design virtual machine unit scheduler and adopt the concept of time slice and schedule token to support the proportional share of I/O bandwidth. By using the management method, the virtual machine system can allocate I/O bandwidth among many virtual machines equally or discriminately according to service priority or I/O requirement.

In [15], Li et al. proposed a real-time scheduling mechanism of resource for multiple virtual machine system. In the scheduling mechanism, the architecture of system and the real-time scheduling strategy of resource are two important factors. In order to minimize the total execution time, two scheduling algorithms are proposed in the real-time scheduling strategy of resource, which is named the virtual machine monitor scheduling algorithm and the processor selection algorithm, respectively.

In [16], a DMM model is proposed in order to resolve the question of dynamic memory management. In fact, the DMM model is a low-level memory management mechanism, which allows dynamic change of the mapping between the psedo-physical memory as seen from VMs and the machine memory, while the virtual machine is running. By using the DMM model, the DMM layer can incorporate high-level policies and low-level implementations and achieve the dynamic memory management.

In [17], a data management solution is presented, which allows fast Virtual Machine (VM) instantiation and efficient run-time execution to support VMs as execution environments in grid computing. By using the method, the system can provide on-demand cross-domain access to VM state for unmodified VM monitors, support user-level and write-back disk caches, per-application caching policies and middleware-driven consistency models, and leverage application-specific meta-data associated with files to expedite data transfers.

In [18], an approach is presented in order to manage the quality of service (QoS) of virtualized resources in multicore machines. In the approach, a new architecture is devised to build a high-level service that combines interdomain communication mechanisms with monitoring and control primitives for local resource management. By using the approach, fine-grain resource allocation and efficient assignment can be achieved.

In order to run multiple virtual machines in a computer system, it is necessary for people to accurately predict these idle intervals of workloads and save energy of disk drives. Deng et al. propose to divide these workloads into buckets which are equal in time length, and predict the number of the forthcoming requests in each bucket instead of the length of the idle periods [19]. By doing so, the bucket method makes the converted workload more predictable. The method also squeezes the executing time of each request to the end of its respective bucket, thus extending the idle length. By deliberately reshaping the workloads such that these crests and troughs of each workload become aligned, the peaks and the idle periods of these workloads can be aggregated. Thus, energy can be conserved.

In order to decrease the power consumption and cooling overheads, Nathuji et al. propose a set of management components and abstractions for use by software power budgeting policies [20]. The key idea is to manage power from a VM-centric point of view, where the goal is to be aware of global utility tradeoffs between different virtual machines (and their applications) when maintaining power constraints for the physical hardware on which they run. By using the set of management components and abstractions, the efficiency of data center can be improved.

In order to coordinate and tradeoff the power and performance of virtual machine, a power aware resource allocation algorithm, named PaRA, is proposed [21]. In the power aware resource allocation algorithm, the virtual machine monitor is responsible for allocating basic resources such as CPU slices, memory capacities, and disk and network I/O bandwidths. In runtime, resource allocation is automated based on the workload characterization and the real time power consumption is measured.

In order to reduce power consumption in a computer cluster, a novel power-aware scheduling algorithm is presented, which can allocate virtual machines in a DVFS-enabled cluster by dynamically scaling these supplied voltages [22]. By using the algorithm, the power consumption of a DVFS-enabled cluster can be reduced.

In order to research the energy efficiency cloud computing, a virtual machine based energy-efficient data center architecture for cloud computing is presented [23], which includes the consolidation and migration mechanism. By using the consolidation and migration mechanism, the data center can maintain energy efficiency automatically.

In order to reduce the memory energy consumption in multicore system, several heuristic scheduling algorithms by using a memory power simulator are designed and implemented [24]. By using these algorithms, the memory-aware virtual machine can save memory energy. Therefore, these algorithms are essential to reduce the memory energy consumption in power-aware memory management.

In order to reduce energy consumption in computer clusters, a new power-aware scheduling policy for heterogeneous clusters is proposed [25]. In the new power-aware scheduling policy, the power consumption information of each machine is used to find an allocation of the machine, which results in the maximum energy saving.

Though above these resource scheduling methods and energy consumption reducing mechanisms are very useful for correspondingly application, they will still confront a lot of difficulties in virtual cluster system because these scheduling mechanisms and methods of resource don't consider the factor that the energy consumption of system will increase with the growth the frequent scheduling and migration of resources. Moreover, the above resource scheduling methods and energy consumption reducing mechanisms don't analyze their interaction of the resource scheduling methods and the migration of system resources. In fact, the resources scheduling methods will affect the migration frequency of resources. And the migration frequency of resources will affect the energy consumption of virtual cluster system. Therefore, we present an optimized resource scheduling mechanism to reduce the migration frequency of resources. Thus, the energy consumption of system can be reduced in virtual cluster system.

3 Resource Scheduling Mechanism

In general, the architecture of virtual cluster system is shown as Figure 1. In the architecture, there are three levels, namely the hardware level, the single image management level and the virtual application level. In the hardware level, there are all kinds of physical machine which include CPU, storage, network card and so on. In the single image management level, there are some virtual machine monitors (VMMs) and a single image management module. Each virtual machine monitor is managed and scheduled by the single image management module. The main function of each virtual machine monitor includes: managing its local physical machine safely, providing isolation between the virtual machine monitors and executing the commands that the single image management module sends to it. The main function of the single image management module includes: enabling each virtual machine to share the corresponding physical machines safely, providing isolation between the different virtual machines, providing some different strategies for virtual machines to access hardware resources, controlling all virtual machine monitors and managing the central datum. In the virtual application level, there are some virtual machines which are constructed by people and can provide services for corresponding application.

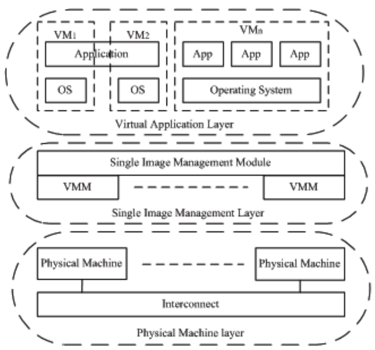

Fig. 1: The Architecture of Virtual Cluster System

In the architecture, the service process of virtual cluster system can be described as follows.

First, each virtual machine submits tasks to the single image management level in term of the requirement of user in virtual cluster system. Then, the single image management level begins to control corresponding virtual machine monitor and schedule corresponding physical resources for these tasks when it receives the task information. At last, these physical resources provide corresponding service for tasks in term of certain control mechanisms and management strategies.

Based on the above analyses, we propose optimized resource scheduling mechanism in virtual cluster system. In the optimized resource scheduling mechanism, we first propose an optimal resource scheduling model for reducing energy consumption. Then, we present a novel grouping genetic algorithm with fuzzy multi-objective evaluation and use the algorithm to resolve the optimal resource scheduling model. At last, we design a resource scheduling management style.

3.1 Optimal resource scheduling model

In order to describe the optimal resource scheduling model, we use m to denote an integer, R_i to denote the total number of the i^{th} physical resource, R to denote the total number of all physical resources in virtual cluster system. Thus, we can get the following equation

$$R = R_1 + R_2 + R_3, \ldots, +R_m \tag{1}$$

Similarly, if we use n to denote an integer, vm_i to denote the i^{th} virtual machine, and VM to denote the set of virtual machines in the virtual cluster system, we can get $VM = \{vm_1, vm_2, vm_3, \ldots, vm_n\}$.

As we all know, the total number of physical resources that all virtual machines can schedule in any time is less than the total number of system resources in virtual cluster system. Therefore, if we use ζ_i to denote the percent that the i^{th} virtual machine can schedule the system resources, $VM_i(R)$ to denote the number that the i^{th} virtual machine can schedule the system resources, we can get:

$$VM_i(R) = \zeta_i * R \tag{2}$$

For each physical resource, its energy consumption can be divided into two parts[21]. One is the static energy consumption and the other is the dynamic energy consumption. The static energy consumption refers to is the energy consumption that the system resource doesn't provide service. The dynamic energy consumption refers to is the energy consumption that the system resource processes tasks. Therefore, in the virtual cluster system, the energy consumption can also be divided into two parts for each virtual machine, namely the static energy consumption and the dynamic energy consumption. If we use $P(vm_i)$, $P_{dynamic}(vm_i)$, $P_{static}(vm_i)$ to denote the energy consumption, the dynamic energy consumption and the static energy consumption of the i^{th} virtual machine, respectively, we can get the following equation [26]:

$$P(vm_i) = P_{dynamic}(vm_i) + P_{static}(vm_i) \tag{3}$$

In fact, the dynamic energy consumption procedure of each virtual machine involves two procedures, namely the processing procedure of tasks and the migration of resources. If we use $P_{processing}(vm_i)$, $P_{migrating}(vm_i)$ to denote of the energy consumption of processing and the energy consumption of migration of the i^{th} virtual machine, respectively, we can get the following equation:

$$P_{dynamic}(vm_i) = P_{processing}(vm_i) + P_{migrating}(vm_i) \tag{4}$$

In addition, the dynamic energy consumption [27][28] $P_{dynamic}(vm_i)$ is computed as follows:

$$P_{dynamic}(vm_i) = A * C_i * \vartheta(i)^2 * s(i) \tag{5}$$

Where, A is the percent of active gates; C_i is the total capacitance load; $\vartheta(i)$ is the supply voltage of the i^{th} virtual machine; $s(i)$ is the processing frequency of the i^{th} virtual machine, which includes the processing frequency of tasks and the migration frequency of tasks.

If we use $s_{processing}(i)$ to denote the processing frequency of tasks, $s_{migrating}(i)$ to denote the migration frequency of resources, based on the equation (5), we have:

$$P_{dynamic}(vm_i) = A * C_i * \vartheta(i)^2 * (s_{processing}(i) + s_{migrating}(i)) \quad (6)$$

Because the set of virtual machines in the virtual cluster system is $VM = \{vm_1, vm_2, vm_3, \ldots, vm_n\}$, we need to find an optimized schedule, which minimizes the energy consumption of the virtual cluster system.

$$P_{min} = min \sum_{k=1}^{n} P(vm_k) \quad (7)$$

Where, $P(vm_k)$ is the energy consumption of the k^{th} virtual machine in the virtual cluster system.

In order to minimize the processing frequency of each virtual machine, a theoretical model about resource scheduling for reducing energy consumption in the virtual cluster system can be presented by using the optimal theory. The theoretical model can be described as follows:

Goal: $min \sum_{k=1}^{n} P(vm_k)$ and $min[max \sum_{i=1}^{m} (1 - \zeta_i)R]$

Constraints:

$$0 \leq \zeta_i \leq 1 \quad (8)$$

$$0 \leq \sum_{i=1}^{n} \zeta_i \leq 1 \quad (9)$$

Where, Equation (8) denotes that the i^{th} virtual machine can schedule the physical resources and the amount of physical resources that it can schedule is no more than the total number of all physical resources of virtual cluster system. Equation (9) denotes that the total of physical resources that all virtual machines can schedule is no more than the total number of all physical resources of virtual cluster system.

In fact, the above theoretical model is an objective function. It is also a multi-objective optimization question. The aim that the objective function is presented here includes two factors. One is to keep the minimum of physical resources that all virtual machine can schedule and the other is to keep the minimum that virtual machine need to be migrated. Because the total of virtual machine and physical machine are limited, it is a NP-hard problem that how to achieve the optimal result.

3.2 Grouping Genetic Algorithm With Fuzzy Multi-objective Evaluation

At present, people usually use the genetic algorithm to resolve the above similar optimal problem. But the genetic algorithm performs poorly on grouping problems such as bin-packing and introduced the grouping genetic algorithm which is a genetic algorithm heavily modified to suit the structure of grouping problems. In addition, special genetic operators for crossover and mutation are used to suit the structure of chromosomes. To further improve the performance, a novel grouping genetic algorithm with fuzzy multi-objective evaluation is proposed. The novel grouping algorithm with fuzzy multi-object evaluation is shown as follows.

Step 1: Make sure the virtual machine set $VM = \{vm_1, vm_2, vm_3, \ldots, vm_n\}$ and the total number of each physical resource $R_i, (i = 1, \ldots, m)$.
Step 2: compute $R = \sum_{i=1}^{m} R_i$
Step 3: λ_0 = Crossover rate, η_0 = Mutation rate.
Step 4: Randomly choose a group initial value $\zeta = \zeta_0, \zeta_1, \zeta_2, \ldots, \zeta_n$.
Step 5: For i=1 To m
{Offspring [1]=0
For j=1 To | VM | *λ_0
{(x, y)=select(ζ,Offspring[j])
Offspring[j]=RankingCrossover(x,y)}}.
Step 6: For j=1 To | VM | * η_0
{y=select(ζ,Offspring[j])
Offspring[j+| VM | *η_0] = Mutation(y)}.
Step 7: ζ=select(y,Offspring[j]), and compute $min \sum_{k=1}^{n} P(vm_k)$ and $min[max \sum_{i=1}^{m} (1 - \zeta_i)R]$
Step 8: End

Where, the algorithm of function select() is described as follows.

Step 1: Randomly choose an initial value ρ_0, an infinitely small number ε, and $k = 0$
Step 2: If $k \leq n$ Then $\{\alpha \leftarrow 1 - k/n$;
$\rho_{k+1} \leftarrow \rho_k * \alpha$;
$\Delta\zeta_k \leftarrow \zeta_k - \rho_k$;
If $0 \leq \Delta\zeta_k \leq 1$ Then $\{\zeta_k^* \leftarrow \Delta\zeta_k\}$
Else $\{\zeta_k^* \leftarrow \zeta_k\}$
$\zeta^* \leftarrow \{\zeta_1^*, \ldots, \zeta_k^*, \zeta_{k+1}, \ldots, \zeta_n\}$
$\Delta f(\zeta) \leftarrow f(\zeta) - f(\zeta^*)$
If $\Delta f(\zeta) > 0$ Then $\{\zeta \leftarrow \zeta^*$
If $| \rho_{k+1} - \rho_k | < \varepsilon$ Then { go to Step 3}
Else $\{k \leftarrow k+1$, go to Step 1 }}
Else $\{\zeta \leftarrow \zeta^*$, go to Step 3 }}}
Step 3: Output $\zeta = \{\zeta_1, \zeta_2, \zeta_3, \ldots, \zeta_n\}$
Step 4: End

3.3 Resource Scheduling Management Style

Based on the basis of the model and the algorithm, we design a resource scheduling management style for reducing energy consumption in resource scheduling management module. In the resource scheduling management style, we will build five function modules, namely a monitor, a calculator, an allocator, a scheduler and a manager. The structure of the resource scheduling management style is shown as Fig.2

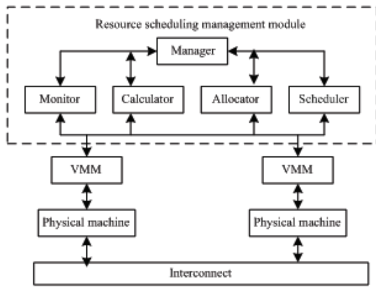

Fig. 2: The Resource Scheduling Management Style

In the resource scheduling management style, the main function of the monitor is to receive the monitor command from the manager, monitor the running state of each VMM and feedback the information that each VMM monitors. The main function of the calculator is to receive the calculate command and the state information about different physical machine from the manager and calculate the number of resource scheduling for corresponding virtual machine by using our proposed resource scheduling management model and grouping genetic algorithm with fuzzy multi-objective evaluation. The main function of the allocator is to receive the allocate command about tasks from the manager, transfer the allocate command to corresponding virtual machine monitors and allocate tasks to corresponding virtual machine monitor in term of the requirement of user. The main function of the scheduler is to receive the schedule command about system resources from the manager, transfer the schedule command to corresponding virtual machine monitors. The main function of the manager is to respectively transfer different command to the monitor, the calculator, the allocator and the scheduler, receive the monitoring information that the monitor transmits to it, count the allotting state of tasks and the scheduling state of physical resources.

The resource scheduling management style for reducing energy consumption can be described as fellows:

Step 1: The manager receives the task information, which is sent by each virtual machine and appends these tasks to a task list T_{List}.
Step 2: The manager transfers the ”monitor” command to the monitor.
Step 3: After receiving the ”monitor” command, the monitor begins to monitor the running state of each virtual machine monitor and feedback the information that each VMM monitors.
Step 4: After receiving the feedback information of each VMM, the manager transfers the ”calculate” command, the task list T_{List} and the feedback information to the calculator.
Step 5: After receiving the ”calculate” command, the calculator begin to calculate the percent that each virtual machine can schedule corresponding physical resources by using our proposed resource scheduling management model and grouping genetic algorithm with fuzzy multi-objective evaluation.
Step 6: After calculating the percent that each virtual machine can schedule, the calculator begins to transfer the result to the manager and the allocator.
Step 7: After receiving the result of the calculator, the manager begins to transfer the allocate command to the allocator.
Step 8: After receiving the ”allocate” command from the manager and the result of the calculator, the allocator begins to allocate the tasks to corresponding VMM and the scheduler.
Step 9: After allocating the tasks, the allocator begins to the result to the manager and the scheduler.
Step 10: After receiving the result of the allocator, the manager begins to transfer the schedule command to the scheduler.
Step 11: After receiving the ”schedule” command from the manager and the result of the allocator, the scheduler begins to schedule physical resources for corresponding virtual machines.
Step 12: After scheduling the physical resources, the scheduler begins to jude if all tasks of T_{List} have been exected. If all tasks of T_{List} has been exected, then {go to Step 13}, else {go to Step 2}.
Step 13: End.

4 Experiments and Results Analysis

In order to validate the efficiency of our proposed resource scheduling mechanism in virtual cluster systems, we do a series of experiments. In these experiments, we first use the open source of Xen, the corresponding virtualization technology and a cluster to construct a virtual cluster system. Then, we use different scheduling methods to schedule system resources and monitor the state of resource scheduling. At last, we will analyze these results of experiments. The whole processes can be described as follows.

4.1 A Series of Experiments

In our experiments, there are three resource scheduling mechanisms and methods, namely our proposed scheduling mechanism, the credit scheduling algorithm and the load balancing mechanism. The system can respectively use these algorithms and mechanisms to schedule the physical machine and provide service for tasks.

In our experiments, there are six physical machines $(M_1, M_2, M_3, M_4, M_5$ and $M_6)$. The supply voltage that each physical machine uses is 220 volt. Moreover, the hardware and the software configurations of each physical machine are the same. In addition, we use the virtual cluster system to built eight virtual machines. The eight virtual machines are named vm_1, vm_2, vm_3, vm_4, vm_5, vm_6, vm_7 and vm_8, respectively.

In our experiments, the task that each virtual machine submits is the same, which is to resolve a linear equation. The linear equation is described as follows.

$$\begin{pmatrix} 12 & -3 & 3 & 4 \\ -18 & 3 & -1 & -1 \\ 1 & 1 & 1 & 1 \\ 3 & 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 15 \\ -15 \\ 6 \\ 2 \end{pmatrix} \quad (10)$$

Moreover, in our experiments, the Gauss elimination method is used for resolving the linear equation. And the rate that each virtual machine submits tasks to the system is 35(Time/S), 26(Time/S), 31(Time/S), 28(Time/S), 23(Time/S), 21(Time/S), 27(Time/S), 24(Time/S), separately. In addition, the percent that each virtual machine can schedule the physical resources of system in the initiate state is shown as $15\%, 12\%, 11\%, 13\%, 13\%, 15\%, 12\%, 9\%$, separately.

By using the statistical method, we can get the even rate that each physical machine provide service for tasks in the three resource scheduling algorithms and mechanisms. The results are shown as Table 1.

Table 1: The even rate that each physical machine provides services for tasks in the three resource scheduling algorithms and mechanisms (Unit: Time/Second)

	Our scheduling mechanism	Credit scheduling	Load balancing
M_1	45.78	41.52	38.42
M_2	36.34	34.67	38.63
M_3	40.89	37.65	34.66
M_4	36.69	32.81	35.57
M_5	37.56	36.93	36.83
M_6	33.96	29.86	23.06

Similarly, we can also get the mean square deviation of rate that each physical machine provides service in these three resource scheduling algorithms and mechanisms. The results are shown as Table 2.

In addition, by using the energy consumption monitor function of our resource scheduling module, we can get the average consumption of energy that each physical machine respectively uses these three different scheduling algorithms in time interval [0, 120s]. The results are shown as Fig. 3

Table 2: The mean square deviation of rate that each physical machine provides services in these three resource scheduling algorithms and mechanisms

	Our scheduling mechanism	Credit scheduling	Load balancing
M_1	0.78	1.64	2.99
M_2	0.69	1.7	0.86
M_3	0.61	0.93	1.65
M_4	0.14	0.17	0.18
M_5	0.13	0.12	0.19
M_6	1.28	2.13	3.96

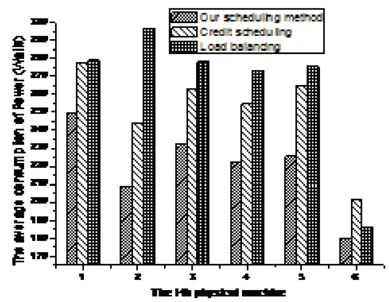

Fig. 3: The Average Consumption of Energy That Each Physical Machine Respectively Uses These Three Different Scheduling Algorithms in Time Interval [0, 120s]

4.2 Results Analysis

In this section, we will analyze these results of our experiments. In order to describe the analyzing process conveniently, we use T_{Our}, T_{Credit} and T_{Load} to denote the even rate that the virtual cluster system provides services for tasks in the three resource scheduling algorithms and mechanisms, respectively. Based on the results of Table 1, we can get:

Because $T_{Our} > T_{Credit} > T_{Load}$, the even rate that the virtual cluster system provides services for tasks in our proposed resource scheduling mechanism is faster than that in the other two scheduling algorithms. This indicates: the rate that the virtual cluster system provides services for tasks in our proposed resource scheduling mechanism is faster than that in the other two scheduling algorithms.

If we use D_{Our}, D_{Credit} and D_{Load} to denote the mean square deviation of the rate that the virtual cluster system provides service in the three resource scheduling algorithms and mechanisms, respectively. Then, we can calculate the result of D_{Our}, D_{Credit} and D_{Load} in term of the results of Table 2, respectively. The processes can be described as follows.

$D_{Our} = (0.78+0.69+0.61+0.14+0.13+1.28)/6 = 0.605$

$D_{Credit} = (1.64+1.7+0.93+0.17+0.12+2.13)/6 = 1.115$

$D_{Load} = (2.99+0.86+1.65+0.18+0.19+3.96)/6 = 1.638$

Because $D_{Load} > D_{Credit} > D_{Our}$, the mean square deviation of rate that the virtual cluster system provides service in our proposed resource scheduling mechanism is smaller than that in the other two scheduling algorithms. This indicates: the stability that the virtual cluster system provides services for tasks in our proposed resource scheduling mechanism is better than that in the other two scheduling algorithms.

If we use P_{Our}, P_{Credit} and P_{Load} to denote the average energy consumption of the virtual cluster system in the three resource scheduling algorithms and mechanisms, respectively. Then, we can calculate the result of P_{Our}, P_{Credit} and P_{Load} in term of the results of Figure 3, respectively. The results can be described as follows:

$P_{Our} = 1318.07$ (watts)

$P_{Credit} = 1506.35$ (watts)

$P_{Load} = 1588.41$ (watts)

Because $P_{Load} > P_{Credit} > P_{Our}$, the average energy consumption that the virtual cluster system provides service in our proposed resource scheduling mechanism is smaller than that in the other two scheduling algorithms. This indicates: Our proposed resource scheduling mechanism can reduce energy consumption. Relative to the other two scheduling algorithms, the energy consumption can reduce 12.5% and 17% by using our proposed resource scheduling mechanism, respectively.

5 Conclusions

Based on the above analyses, we can find: (1) our proposed scheduling method is better than the other two scheduling algorithms in virtual cluster system. (2) Our proposed scheduling method can reduce energy consumption in virtual cluster system. (3) Our proposed scheduling method is efficient. The main reasons that these situation can generate include two factors. The first factor is that our proposed scheduling mechanism can minimize the frequency that each virtual machine processes tasks. The second factor is our proposed scheduling mechanism can minimize the frequency that the system resources are migrated. Based on the two reasons, the energy consumption of virtual cluster system can be reduced by using our proposed resource scheduling mechanism.

Acknowledgement

This paper is supported by National Basic 973 Research Program of China under grant No.2007CB310900, Zhejiang Povincial Natural Science Foundation of China under Grant No. Y1090297, Y6090312.

References

[1] R. J. Creasy, IBM Journal of Research and Development **25**, 483-490 (1981).
[2] H. B. Zou, H. Jin, Z. F. Han, X. H. Shi, H. H. Chen. IEEE International Conference on e-Business Engineering. 541-544 (2005).
[3] M. Kallahalla, M. Uysal, R. Swaminathan, D. Lowell, M. Wray, T. Christian, N. Edwards, C. Dalton, and F. Gittler, IEEE Press **37**, 46-54 (2004).
[4] H. B. Zou, H. Jin, Z. F. Han, J. Tie, X. H. Shi. Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence. 50C53 (2005).
[5] J. Sugerman, G. Venkitachalam, and B. H. Lim, USENIX Annual Technical Conference, 1-4 (2001).
[6] X. H. Sun, C. Du, H. B. Zou, Y.Chen, Shukla P.IEEE International Conference on Cluster Computing and Workshops.1-7 (2009).
[7] Z. K. Wang, X. Z., S. Singhal, Proceedings of the 16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, 133-144 (2006).
[8] B. Lin, P. A. Dinda, First International Workshop on Virtualization Technology in Distributed Computing, 6-6 (2006).
[9] K. Korotaev, IEEE International Conference on Cluster Computing, 1-1 (2005).
[10] J. F. An, X. Y. Fan, S. B. Zhang, The Second International Conference on Embedded Software and Systems, 514-521 (2005).
[11] S. R. Seelam, P. J. Teller, Proceedings of the 3rd International Conference on Virtual Execution Environments, 105-115 (2007).
[12] J. G Park, J. M. Kim, H. Choi, and Y. C. Woo, Proceedings of the 11th International Conference on Advanced Communication Technology **3**, 2077-2083 (2009).
[13] X. H. Shi, H. Y. Tan, S. Wu; and H. Jin, Japan-China Joint Workshop on Frontier of Computer Science and Technology, 155-162 (2008).
[14] D. J. Kang, C. Y. Kim, K. H. Kim, and S. I. Jung, International Conference on Computer Science and Information Technology, 647-653 (2008).
[15] Y. F. Li, X. H. Xu, J. Wan, and W. Q. Li, The 5rd ChinaGrid Annual Conference, 137-143 (2010).
[16] X. L. Wang, and Y. W. Luo, Chinese Science Bulletin **55**, 3071 (2010).
[17] M. Zhao, J. Zhang, and R. J. Figueiredo, Cluster Computing **9**, 45-56 (2006).
[18] F. R. Haro, F. Freitag, and L. Navarro, Annals of Telecommunications **64**, 5-6 (2009).
[19] Y. H. Deng, and B. D. Pung, Computing **91**, 3-21 (2011).
[20] R. Nathuji, K. Schwan, A.Somani, and Y. Joshi, Cluster Computing **12**, 189-203 (2009).
[21] C. F. Jiang, J. L. Zhang; J. Wan, X. H. Xu, Y. Y. Yin, R. T. Yu, and C. P. Lv, IEEE/ACM Int'l Conference on Green Computing and Communications, 313-318 (2010).
[22] V. L. G, L. Z. Wang, A. J. Younge, and X. He, IEEE International Conference on Cluster Computing and Workshops, 1-10 (2009).
[23] K. J. Ye, D. W. Huang, X. H. Jiang, H. J. Chen, and S. Wu, IEEE/ACM Int'l Conference on Green Computing and Communications, 171-178 (2010).

[24] J. W. Jang, M. Jeon; H. S. Kim, H. Jo, J. S. Kim, S. Maeng, IEEE Transactions on Computers **60**, 552-564 (2011).
[25] A. D. Hadil, A. A. Issam, D. G. Down, 2010 International Green Computing Conference, 325-332 (2010).
[26] K. H. Kim, R. Buyya, and J. Kim, the Seventh IEEE International Symposium on Cluster Computing and the Grid, 541-548 (2007).
[27] R. Ge, X. Feng, and K. Cameron, Proceedings of the 2005 ACM/IEEE Conference on Supercomputing.IEEE Computer Society Washington, (2005).
[28] J. Li, and J. F. Martinez, In High-Performance Computer Architecture, 77-87 (2006).

Yunfa Li, born in 1969, is a Ph.D. and associate professor in school of Computer Science and Technology at Hangzhou Dianzi University. His research interests include cloud computing, cluster computing, grid computing, big data and system security.

Sheng Xiong, born in 1990, is a postgraduate in school of Computer Science and Technology at Hangzhou Dianzi University. His research interests include cloud computing, big data and system security.