# A Virtual Machine Dynamic Migration Scheduling Model Based on MBFD Algorithm

Xin Lu and Zhuanzhuan Zhang

***Abstract*—This paper presents a modified best fit decreasing algorithm (MBFD algorithm) for virtual machines dynamic migration scheduling model. The model uses the bin packing problem classical approximation algorithm idea, designs a new algorithm to deal with virtual machines dynamic migrating scheduling. It finds out load hot-spot hosts in the cloud platform by executing the selection algorithm program. Then the resource loads of virtual machines in hot-spot hosts are sorted in descending order. As for non-hot-spot hosts, their resource loads are sorted in ascending order. By traversing the non-hot-spot hosts queue, it can find the most appropriate host to act as the migration target host.**

***Index Terms*—Cloud computing, virtual machine dynamic migrating, resource scheduling, load balance.**

## I. Introduction

Configuration of software and hardware resources in the traditional data center is usually needed enough high to meet business processing's peak demand, but this will lead to the waste of resources in off-peak periods. To solve this problem, modern data centers are turning to cloud computing model treatment [1]. One of the cloud data center features is that it should have a scalable dynamic resources allocation [2]. Namely according to the user demand load changing of data centers, the data center can automatically expand or contract the allocation of resources in different sever hosts. So that it can effectively avoid the waste of resources and solve the problem of too heavy or too light load in a server host. According to host's current load, cloud data center makes virtual machines migrate to the proper target host for achieving dynamic resource allocation scalable [3], [4]. This is helpful for effective dynamic scheduling and maximum utilizing of computing resources.

There are various methods to solve the problem of virtual machine migration scheduling. Traditional static resource scheduling algorithms include: Round-Robin Scheduling [5], Weighted Round-Robin Scheduling [6], Destination Hashing and Source Hashing Scheduling [7], [8]. These static virtual machine migrating strategies don't mind the system load changes and also do not take dynamically scheduling resource on the basis of user requests into consideration. Because of this, unnecessary migration overhead exists. Mohammad H. Al Shayeji and M. D. Samrajesh [9] put forward an energy aware virtual machine migration algorithm. By referring to the load state of a physical host, this algorithm can effectively migrate virtual machines among hosts. In addition, this algorithm can make the energy consumption minimized by closing the idle hosts. Document [10] provides a double-threshold dynamic virtual machine migration scheduling strategy. This strategy only finds a solution method for making virtual machine migration, but it doesn't put forward how to choose the appropriate migrating target host. In document [11], the existing virtual machine migration scheduling algorithms are analyzed and compared. However, it does not propose a specific solution that how the dynamic virtual machine migration scheduling strategy can be implemented.

The Bin Packing Problem [12] can be described as shown below.

Given: $n$ objects need to be placed in bins of capacity $L$ each. Object $i$ requires $l_i$ units of bin capacity.

Object: determine the minimum number of bins needed to accommodate all n objects.

Bin packing problem is NP complete when formulated as a decision problem. As an optimization problem bin packing is NP-hard. Approximation Algorithm for Bin Packing includes First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD) and Best Fit Decreasing (BFD).

In this paper, the virtual machine dynamic migration scheduling will be seen as a NP-Hard bin packing problem. We will study how to improve the approximation algorithm for bin packing in virtual machine migration scheduling [13]. Realize a virtual machine dynamic migrating model in cloud data center, to meet the demand of load balance and the service level agreement.

## II. The Problem Description

Cloud data center virtual machine migration scheduling refers to platform system getting the load data of each host by monitoring its running state in data center. And according to a certain scheduling strategy, the virtual machine dynamic migration executes. On the cloud computing virtualization infrastructure platform environment, this virtual machine migration scheduling needs to ensure all physical hosts as far as possible to achieve load balance in the data center. The virtual machine dynamic migration scheduling mechanism based on cloud data center can be described in Fig. 1.

Cloud data center is mainly responsible for creating, migrating and cancelling virtual machines in hosts.

Manuscript received March 12, 2014; revised June 24, 2014. This work was supported by a grant from the Joint Research Project of Changhong Electric Co., Ltd and University of Electronic Science and Technology of China in 2013.

Xin Lu is with the School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China (e-mail: luxinmail@uestc.edu.cn).

Zhuanzhuan Zhang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China (e-mail: zhuanzhuanzhang@126.com).

DOI: 10.7763/IJCTE.2015.V7.971

Moreover it can deploy a virtual machine to a proper host, and choose this host to process the service requests from user applications. Load monitoring program is executed for collecting load data of CPU, memory, storage, bandwidth and other resources of each virtual machine and physical host. And the cloud data center captures the platform's resources usage information by interacting with the load monitoring program. When a host's load is over heavy or over light, cloud data center will execute the VM migrating scheduling by the platform's load balance strategy.

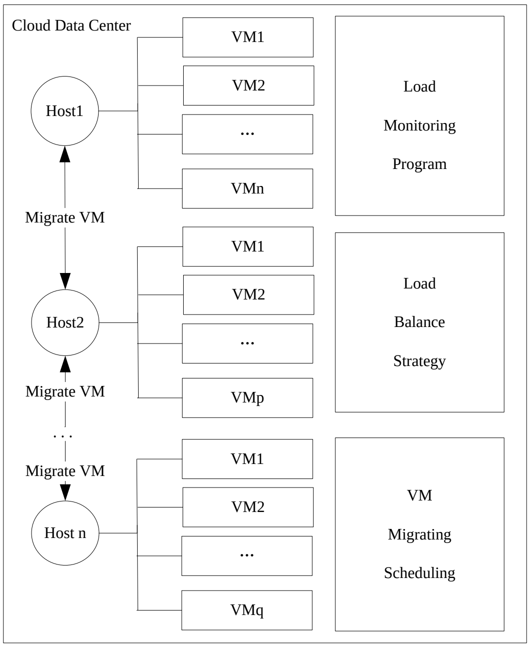



Fig. 1. The virtual machine dynamic migration scheduling mechanism.

In order to fulfill rapidly processing for user request service, the cloud data center needs dynamically migrating virtual machines among hosts to meet the system processing performance requirements. The key problems for the virtual machine dynamic migration scheduling in cloud data center include: 1) determine when to migrate virtual machine, and what is the migration scheduling strategy; 2) when to run the migration operation, how to reduce overhead of the platform; 3) how to judge whether the platform can achieve the load balance.

## III. Design of Scheduling Model

In order to solve these problems of virtual machine dynamic migration scheduling in cloud data center, this paper presents the virtual machine dynamic migration scheduling model as be shown in Fig. 2.

When a user request service reaches to the cloud data center, firstly it will be processed by the Scheduling Main Controller. The main controller will evoke the relevant host's task running. Each host has a load monitoring program. It real-time monitors the load state of host and its virtual machines. Load monitoring programs provide the host load data to the hot-spot hosts selecting program of this platform. Through analyzing these data, hot-spot hosts in this platform can be found out. By this way, all hot-spot hosts can be listed into a queue. The virtual machine dynamic migration scheduling strategy program will be responsible to make reasonable virtual machines migration scheduling. These selected virtual machines will be migrated from the hot-spot-hosts queue to appropriate target hosts, so that all hosts to achieve load balance in the data center platform. So this model needs to solve hot-spot host selection algorithm program and virtual machine dynamic migration scheduling strategy.

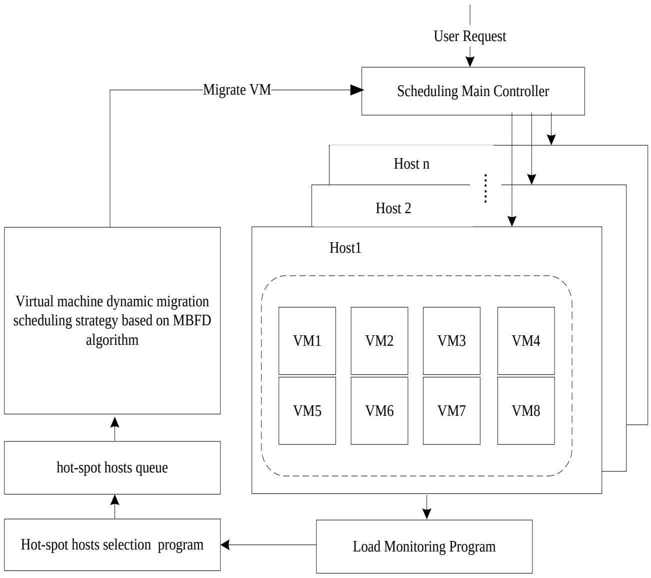



Fig. 2. Virtual machine dynamic migration scheduling model based on MBFD algorithm.

### A. The Selection Algorithm of Hot-Spot Hosts

The basic method of cloud computing dynamic resources scheduling is migrating virtual machine among hosts in the data center, so that every host can run at the optimal load states. If the load of a host is too heavy, it will violate the SLA service level agreement, thus affecting the QoS of executing tasks. On the other hand, if the load of a host is too light, recourses of the host will be in a low utilization state. The power consumption of these hosts will also be wasted. Therefore, the resource utilization thresholds of a host need to be set up in the hot-spot hosts' selection algorithm. Max usage amount of a host is set as the maximum threshold. Min usage amount of a host is set as the minimum threshold. The load monitoring program is executed to monitor resources utilization of every host. Once it finds that the load condition of a host exceeds the maximum threshold or the minimum threshold, this host will be marked as hot-spot host. Specific criteria are shown in Equation (1).

$$\begin{cases} U_{\text{cpu}}(p_i) > \text{Max}_{\text{cpu}} \quad \text{or} \quad U_{\text{cpu}}(p_i) < \text{Min}_{\text{cpu}} \\ U_{\text{ram}}(p_i) > \text{Max}_{\text{ram}} \quad \text{or} \quad U_{\text{ram}}(p_i) < \text{Min}_{\text{ram}} \\ U_{\text{band}}(p_i) > \text{Max}_{\text{band}} \quad \text{or} \quad U_{\text{band}}(p_i) < \text{Min}_{\text{band}} \end{cases} \tag{1}$$

In the Equation (1), $U_{\text{cpu}}(P_i)$, $U_{\text{ram}}(P_i)$ and $U_{\text{band}}(P_i)$ denote the load state of CPU, memory and network bandwidth in the physical host $P_i$. And $\text{Max}_{\text{cpu}}$, $\text{Max}_{\text{ram}}$ and $\text{Max}_{\text{band}}$ respectively represent a predetermined maximum load threshold of CPU, memory and network bandwidth. $\text{Min}_{\text{cpu}}$, $\text{Min}_{\text{ram}}$ and $\text{Min}_{\text{band}}$ respectively represent a predetermined minimum load threshold of CPU, memory and network bandwidth. As long as one of these conditions is established, the host will become a hot-spot one.

Only the host is confirmed at least k times in the recent n times monitoring. It will eventually be identified as a hot-spot host. The value selection of *n* and *k* depends on using a positive strategy or a conservative one. If the value of n is given, the smaller value k is used to recognize a hot-spot host in a short time, which is the positive strategy. On the contrary, it means that a longer period of time will be needed to recognize a hot-spot host, which is the conservative strategy. This above strategy application should be decided by the actual situation.

### *B. The Scheduling Strategy Based on MBFD*

Using the hot-spot host selection algorithm introduced in the previous subsection, we can easily identify hot-spot hosts in the cloud data center. And all hot spot hosts will form a queue. Once the hot-spot hosts are found, using what kind of scheduling strategy for dynamic virtual machine migration to achieve load balancing among hosts, becomes the key problem of dynamic virtual machine migration scheduling model.

This paper will utilize the ideas of approximation algorithms in bin packing problem, and presents a modified best fit decreasing algorithm (MBFD algorithm) based on the cloud computing application environment. So that it can be used to resolve the problem of virtual machine dynamic migration scheduling strategy.

When using virtual machine dynamic migration scheduling strategy based on MBFD algorithm, the hot-spot host is seen as an item. The target host is seen as a packing bin. Firstly, the scheduling strategy needs to sort hot-spot hosts in descending order by the size of items. In the hot-spot hosts queue, monitor the load state of the CPU, memory, network bandwidth of all the virtual machines in every hot-spot host. Weigh these three types of resources in some way to measure the load state of the entire virtual machine. Then, the scheduling strategy program can sort all the virtual machines of a certain hot-spot host in descending order by its resource load state. Traverse the non-hot-spot host queue to find the most appropriate one as a migrating packing bin. This means after loading the virtual machine into the target host, the difference between the current load state of this host and the Max threshold must be a minimum.

### *C. The Scheduling Algorithm*

In the previous subsection, the virtual machine dynamic migration scheduling strategy based on the MBFD is expounded. This section will introduce the virtual machine dynamic migration scheduling algorithm based on MBFD.

According to host load state data collected by the monitoring program, and combining with the hot-spot host selection algorithm proposed in Section A, hot-spot host queue can be drawn as shown in Equation (2).

$$\begin{cases} \text{Heat}_p = \{p_1, p_2, \ldots\ldots, p_n\}, \quad n \in N^* \\ U_{p_i} \geq \text{Max} \;\text{ or }\; U_{p_i} \leq \text{Min}, \quad \forall i \in [1, n] \end{cases} \tag{2}$$

In the Equation (2), the element $p_i$ of hot-spot host queue, $\text{Heat}_p$ represents the physical host which meets certain conditions. Namely the resource load state of this host exceeds a predetermined Maximum threshold value max, or it is below the minimum threshold value Min.

After determining the hot-spot host queue, next the virtual machines in the hot-spot host need to be sorted in descending order according to their resource load state. In order to do this, the virtual machine resource load status $u_i$ should be calculated, specifically as be shown in Equation (3).

$$u_p = w_1 u_{\text{cpu}} + w_2 u_{\text{ram}} + w_3 u_{\text{band}} \;\&\; \sum_{i=1}^{3} w_i = 1 \tag{3}$$

In the Equation (3), $u_{\text{cpu}}$, $u_{\text{ram}}$, $u_{\text{band}}$ respectively denote the load state of CPU, memory and network bandwidth of a virtual machine. These three types of data are collected by the load monitoring program. Here, a weighted sum operation is conducted. And $w_1$, $w_2$ and $w_3$ represent different weights.

Then the algorithm can sort virtual machines in the hot-spot hosts by descending order according to virtual machines' resource load state, as be shown in Equation (4).

$$\begin{cases} L_p = \{v_1, v_2, \ldots\ldots, v_m\}, \; p \in \text{Heat}_p, m \in N^* \\ u_i \leq u_j, \quad \forall i, j \in [1, m] \end{cases} \tag{4}$$

In the Equation (4), $L_p$ represents a virtual machine queue. These virtual machines are in the same physical host, and sort in descending order according by the state of their resource load $u_i$. This has been determined under what kind of situation the migration of virtual machines takes action. It means that choosing which virtual machine in which hot-spot host to do the migration operation.

It is necessary to sort all the non-hot-spot hosts in an ascending order according to their resource load state. Specifically be shown in Equation (5).

$$\begin{cases} \text{Target}_p = \{p_1, p_2, \ldots\ldots, p_k\}, \quad k \in N^* \\ U_{p_i} \geq U_{p_j}, \quad \forall i, j \in [1, k] \\ \text{Min} < U_{p_i} < \text{Max} \end{cases} \tag{5}$$

$\text{Target}_p$ indicates the sorted target host queue. The so-called target host is the one which the resource load state is between maximum threshold Max and minimum threshold Min.

The next step is to consider how to select the most appropriate target during the migration. Traverse $\text{Target}_p$ target host queue to find the most suitable one. Use $V\text{Target}_p$ to represent, specifically be shown in Equation (6).

$$\begin{cases} V\text{Target}_p = p_i \\ \text{Min}\{\text{Max-}(U_{p_i} + u_j)\} \\ u_j \in L_p, p_i \in \text{Target}_p \end{cases} \tag{6}$$

The most suitable target host means after loading the virtual machine in the hot-spot host, the difference between the current load state of this host and the Max threshold must be a minimum. Use the second condition in the Equation (6) to represent.

In order to measure the load balance of the entire data center after the virtual machine dynamic migration scheduling is done, load balancing degree Bal will be calculated below, specifically as be shown in Equation (7).

$$\text{Bal}^2 = \frac{1}{n}\left[\left(U_1\text{-}\overline{U}\right)^2 + \left(U_2\text{-}\overline{U}\right)^2 + \ldots + \left(U_n\text{-}\overline{U}\right)^2\right] \quad (7)$$

In the Equation (7), $\overline{U}$ represents the average value of load resource state of all the physical hosts in the data center, $U_i$ represents the resource load state of the *i*-th physical host. Calculate the load balancing degree using the Equation (7), the smaller the value is the better the host load balance is.

The virtual machine dynamic migration scheduling algorithm can be shown in Fig. 3.

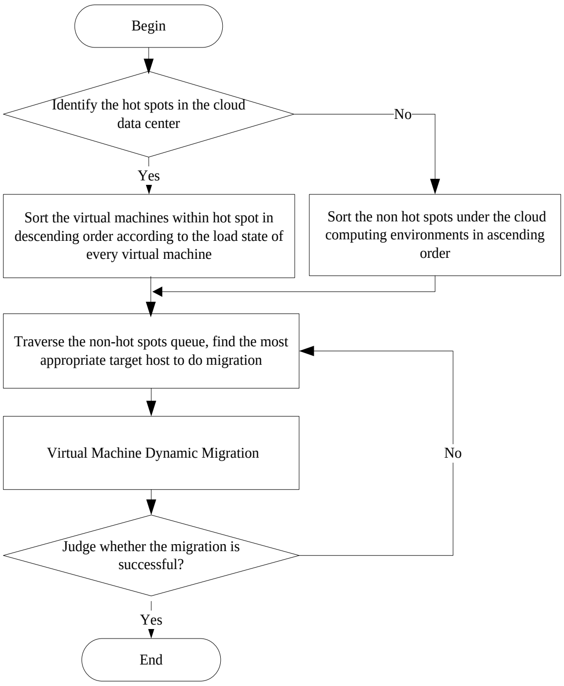



Fig. 3. The flow of scheduling algorithm.

The virtual machine dynamic migration scheduling process is composed by the following steps: 1) Identify hot-spot hosts: Load monitor program detects resource load state of every physical host regularly, so that it can determine the virtual machine in which host needs to be migrated. 2) Select the target host: Sort the non-hot-spot hosts under the cloud computing environments in ascending order, choose the one which meets the conditions of migration as the target. 3) Virtual Machine Dynamic Migrate: Pack CPU, memory, network connection information of a virtual machine in the hot-spot host which you want to migrate, and transport it to the target. 4) Judge whether the migration is successful: Hot-spot host sends a message to the target to tell it that migration is finished. If the submission is completed, the target sends a confirmation message to the hot-spot host, which means that now it has a fully consistent virtual machine image of a hot-spot host. Namely the migration is successful.

## IV. Simulate Experiment

In this paper, we use cloud computing simulation software CloudSim [14], [15] to simulate the virtual machine dynamic migration scheduling model based on MBFD. In this simulation, cloud data center contains five physical hosts with the same configuration, which provides 50 identical virtual machines. And simulate 50 tasks under the cloud environment. At the same time, the maximum threshold of the host load is defined as 80% of the system resources load, and the minimum threshold is defined as 20% of system resources load. This experiment dynamically migrates virtual machines based on the current resource load state of hosts. We have finished several experiments comparing the virtual machine dynamic migration scheduling model based on MBFD with NPA (Non Power Aware) [16], [17], DVFS (Dynamic Voltage and Frequency Scaling) [18] and ST (Simple Threshold), which reflecting the advantages of energy-saving goals, reducing energy consumption and the number of migration. In this experiment, take 10s time as a cycle to monitor the load status of each host.

This simulation is based on the dynamic virtual machine migration scheduling strategy based on MBFD algorithm and ST (0.8), ST (0.6) scheduling strategy. After finishing this experiment, the load balancing degree of these models' hosts are shown in Fig. 4.

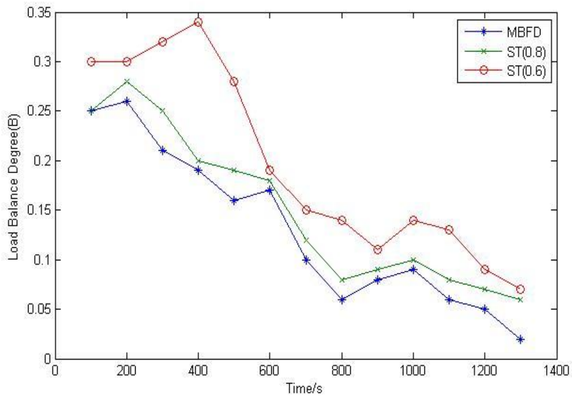



Fig. 4. Comparison of load balance degree.

It can be seen from the figure above that load balance degree of ST (Single Threshold) virtual machine dynamic scheduling is significantly greater than the load balance degree of scheduling model based on MBFD algorithm. So a conclusion can be drawn that the virtual machine dynamic migration scheduling model based on MBFD algorithm can achieve better load balancing degree.

System overhead parameters of dynamic virtual machine dynamic migration process in this experiment are energy consumption, number of migrations and SLA violation percentage. The specific experiment results are shown in Table I.

TABLE I: System Overhead Experiment Results

| Policy | Energy(E/kWh) | Migration | SLA Violation |
|---|---|---|---|
| NPA | 4.48 | — | — |
| DVFS | 2.32 | — | — |
| ST(0.6) | 1.72 | 338 | 11.32% |
| ST(0.7) | 1.63 | 332 | 10.45% |
| ST(0.8) | 1.59 | 335 | 10.81% |
| MBFD | 1.43 | 309 | 10.17% |

The simulation results in Table I are shown that the proposed scheduling strategy based on MBFD in this paper compared with NPA and DVFS, the traditional static resources scheduling strategy is more energy overhead. Strategy model based on MBFD achieves the best energy savings: by 68% and 38% less energy consumption relative to NPA, DVFS. At the same time, MBFD has an advantage over ST, the dynamic resources scheduling strategy in energy consumption. And it is about 13% lower.

The number of migration when using the virtual machine dynamic migration scheduling strategy based on MBFD migration is reduced by about 30 times than ST. So it can reduce the system overhead caused by migration. Although change in migration times appears to be small, but the reduction of migration cost in the actual situation is considerable. In addition, the percentage of SLA violations of scheduling strategy based on MBFD is also shown a downward trend compared with ST (Simple Threshold), which can provide greater performance to meet the requirement of a service level agreement.

## V. Conclusions

In this paper, the load monitor program is used to monitor resources load state of physical hosts and virtual machines. Once the load state of a host exceeds the maximum threshold, or is less than the minimum threshold, this host will be added into the hot-spot host queue. The virtual machine dynamic migration scheduling model based on MBFD algorithm will sorts the virtual machines within the hot-spot host queue in descending order according to their load state. As for non-hot-spot hosts, this model also sorts them in ascending order by their load state. This will help model to fulfill virtual machines dynamic migrating from hot-spot hosts into the proper non-hot-spot hosts. All above needs to be realized by the scheduling model based on MBFD. This model makes the cloud data center to achieve better load balance and decrease the overhead of virtual machines migrating.

## References


[1] UCB/EECS-2009-28, "Above the Clouds: A Berkeley view of cloud computing," UC Berkeley Technical Report, UC Berkeley, 2009.

[2] M. Mishra and A. Das, "Dynamic resource management using virtual machine migrations," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 34-40, September 2012.

[3] D. G. do Lago, E. R. M. Madeira, and L. F. Bittencourt, "Power-aware virtual machine scheduling on clouds using active cooling control and DVFS," in *Proc. 2011 ACM/IFIP/USENIX 12th International Middleware Conference*, 2011.

[4] H. Liu, H. Jin, X. Liao, C. Yu, and C. Xu, "Live virtual machine migration via asynchronous replication and state synchronization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 1986-1999, December 2011.

[5] R. V. Rasmussen and M. A. Trick, " Round Robin scheduling — A survey," *European Journal of Operational Research*, vol. 188, no. 3, pp. 617-636, August 2008.

[6] Wikipedia. (September 2013). Weighted round robin. [Online] Available: http://en.wikipedia.org/wiki/Weighted_round_robin

[7] B. X. Chen, X. F. Fu, X. Y. Zhang, L. Su, and D. Wu, "Design and implementation of intranet security audit system based on load balancing," in *Proc. 2007 IEEE International Conference on Granular Computing*, 2007, pp. 588-591.

[8] K. S. J. Hielscher and R. German, "A low-cost infrastructure for high precision high volume performance measurements of web clusters," *Computer Performance Evaluation/Tools*, vol. 2794, pp. 11-28, 2003.

[9] M. H. Al Shayeji and M. D. Samrajesh, "An energy-aware virtual machine migration algorithm," in *Proc. 2012 International Conference on Advances in Computing and Communications*, 2012, pp. 242-246.

[10] Y. Q. Fang, D. H. Tang, and J. W. Ge, "Research on scheduling strategy based on dynamic migration of virtual machines in cloud environment," *Microelectronics & Computer*, vol. 29, no. 4, pp. 45-48, 2012.

[11] A. Beloglazov and R. Buyya, "Energy efficient allocation of virtual machines in cloud data centers," in *Proc. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 577-578, 2010.

[12] A. Das, C. Mathieu, and S. Mozes, "The train delivery problem- vehicle routing meets bin packing," in *Proc. 8th Workshop on Approximation and Online Algorithms*, 2011, pp. 94-105.

[13] S. I. Gass and C. M. Harris, *Encyclopedia of Operations Research and Management Science*, US: Springer, 2011, pp. 33-35.

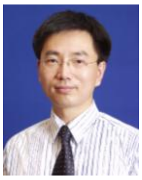
[14] R. N. Calheiros, R. Ranjan, and A. Beloglazov, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23-50, 2011.

[15] The Cloud Computing and Distributed Systems Laboratory, University of Melbourne. CloudSim: A framework for modeling and simulation of cloud computing infrastructures and services. (June 2013). Available: http://www.cloudbus.org/cloudsim/

[16] P. Bohrer, E. N. Elnozahy *et al*., "The case for power management in web servers," *Power Aware Computing*, US: Kluwer Academic Publishers, pp. 261–289, 2002.

[17] A. Wierman, L. Andrew, and A. Tang, "Power-aware speed scaling in processor sharing systems: Optimality and robustness," *Performance Evaluation*, vol. 69, no. 12, pp. 601–622, 2012.

[18] K. H. Kim, A. Beloglazov, and R. Buyya, "Power-aware provisioning of cloud resources for real-time services," in *Proc. the 7th International Workshop on Middleware for Grids, Clouds and e-Science*, 2009, pp. 1:1–1:6.


**Xin Lu** was born in Chengdu, Sichuan Province. He has been an associate professor at School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include cloud computing, software engineering and information system.



**Zhuanzhuan Zhang** is currently a master student with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. Her main research interests are in cloud computing, software engineering and information system.