

Multi-object Virtual Machine Placement Optimization Method in Cloud Computing Environment

Lijun XU^{1,2}, Chunlin LI¹ and Xiaoqing ZHANG³

¹ School of Computer Science and Technology, Wuhan University of Technology,
Wuhan, 430063, China;

² School of Computer and Information Engineering, Xinxiang University,
Xinxiang 453003, China;

³ School of Mathematics and Computer Science, Wuhan Polytechnic University,
Wuhan 430023, China;

E-mail: 1450384751@qq.com

Abstract

The necessity and significance of improving the energy efficiency of cloud implementations have increased due to the rapid growth and proliferation of cloud computing services around the world. This paper has concentrated on consolidation and migration technology in virtualized cloud datacenters as a solution to deal with this high energy consumption problem. We propose VMPA strategy as a novel energy-efficient virtual machine (VM) placement procedure in cloud datacenters. VMPA could reduce the number of active servers in cloud datacenters. Besides, we introduce the central importance of optimizing different targets in cloud datacenters at the same time including total energy consumption, SLA violation and number of VM migration. We present a novel two-threshold method along with VM migration selection algorithm VMMSA as a solution of solving this multi-criteria problem. Two-threshold method can lead to less active servers and VMMSA can generate least number of VM migration. Experimental results show VMPA-VMMSA can improve energy consumption in cloud datacenters, reduce SLA violation and number of VM migration respectively in comparison with state of the arts.

Keywords: cloud computing, datacenter, energy consumption, SLA violation, VM migration

1. Introduction

Cloud computing has recently been brought into focus in both academic and industrial communities due to the increasing pervasive applications and the economy of scale that cloud computing provides [1,2]. As a direct result of cloud computing's increasing popularity, cloud computing service providers such as Amazon, Google, IBM and Microsoft have begun to establish increasing numbers of energy hungry data centers for satisfying the growing customers resource demands [3]. Continuous increase in energy consumption of such huge data centers raises a great concern for both governments and service providers to consume energy more effectively. Apart from the overwhelming operating costs and the total cost of acquisition (TCA) caused by high energy consumption, another rising concern is the environmental impact in terms of carbon dioxide (CO₂) emissions [4]. Since hardware devices consume their near maximum power level when they are idle, not fully utilizing them leads to enormous energy wastage. Research states that servers use nearly 30% of their peak power consumption while sitting idle 70% of time [5]. So, the basic reason of energy waste in data centers' infrastructure is underutilization.

Server consolidation using virtualization is an effective approach to achieve better

energy efficiency of cloud data center. The reason is that at times of low load, VMs are consolidated on a limited subset of the available physical resources, so that the remaining (idle) computing nodes can be switched to low power consumption modes or turned off.

In related works, for optimizing VM placement, ref.[6] has investigated power management techniques in the context of large-scale virtualized systems for the first time. In addition to the hardware scaling and VMs consolidation, they have proposed a new power management method for virtualized systems. In [7], the live migration of VMs is defined as an optimization problem. The optimization objective is to minimize the energy consumption of physical hosts in clouds. But, the above works do not consider the performance and SLA violation from the view of user's applications. Ref.[8] has proposed efficient consolidation algorithms which can reduce energy consumption and at the same time the SLA violations in some cases. They have introduced an efficient SLA-aware resource allocation algorithm that considers the trade-off between energy consumption and performance. Their allocation algorithm takes into account both host utilization and correlation between the resources of a VM with the VMs present on the host. Ref.[9] has investigated the problem of power and performance-efficient resource management in virtualized data center environments. The goal of this paper is to maximize the resource provider's revenue by minimizing power consumption and SLA violation simultaneously. Ref.[10] has improved Bin-Packing algorithm considering a threshold to avoid inopportune VM migration. To this end, if the sum of VMs of a server goes below this threshold value, the VMs of this server are migrated to other servers with enough resources to turn off this server. This approach does not consider the cost of frequently swapping and VMs migration.

Obviously, the VM placement problem in cloud computing environment mainly involves only a single or two targets optimization at present [6-10]. For example, the physical host used is least, or minimizing SLA violations and guaranteeing QoS performance of applications, or reducing the total energy consumption of data center and VM migration number. However, these optimized targets may be in conflict with each other. In order to reduce the number of hosts used, the VMs need to be deployed reasonably to less hosts in bin-packing mode. But this needs multiple VMs migrations to achieve. To reduce SLA violation, VMs have to be consolidated and migrated in many times, which can turn on more hosts. Meanwhile, the energy consumption is associated with the specific resource utilization of physical hosts, so the least used hosts can not always bring the minimum energy consumption. Different from the current works, which mainly focus on the optimization of a single or two objects, our work would consider the energy consumption of physical servers in clouds, SLA violation and the number of VM migration at the same time.

2. Model of VM Placement and Power

Figure 1 gives the framework of energy-efficient virtual machines placement in cloud datacenter in our paper. Under the framework, we can design energy-efficient VMs placement strategy to allocate VMs to physical servers. The framework consists of two main modules:

1. VMs analysis module. This module determines the requested MIPS and memory of each VM arrived and VMs running on the system in each time interval based on the number of VM requests from the users calculated by the front-end server.
2. VM placement strategy module. This module deals with two parameters: 1) the resource demands calculated by the VMs analysis module; 2) the current status (turned on/off) of cloud datacenter equipment and available resources of each turned on server.

The second module tries to find the minimum number of servers and racks required to fulfill the resource demands by VMs and manage to power on the minimum number of switches and cooling devices.

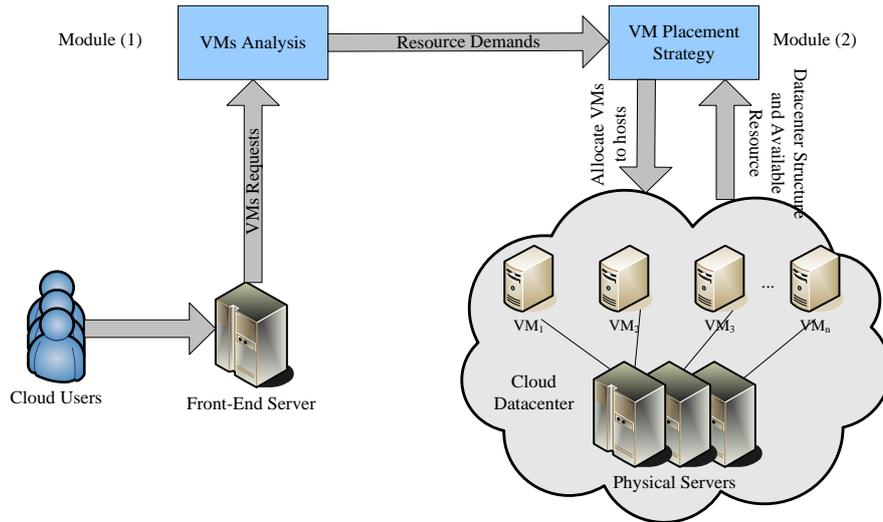


Figure 1. Model of Energy-efficient Virtual Machines Placement in Cloud Datacenter

Power consumption by computing nodes in cloud data centers is mainly determined by CPU, memory, disk storage and network interfaces. Comparing with other cloud system resources, the CPU consumes the main part of energy. Hence, we will focus on the power consumption of CPU and the power management in this paper. The CPU utilization is often proportional to the overall system load, for simplifying the model, a linear power-to-frequency relationship for a computing node in cloud data centers is applied.

The studies show that an idle server consumes approximately 70% of the power consumed by the server running at the full CPU speed on average, which justifies the technique of switching idle servers to the sleep mode to reduce the total power consumption. We apply the following the power model in this paper,

$$P(u) = k \times P_{\max} + (1-k) \times P_{\max} \times u \quad (1)$$

Where P_{\max} is the maximum power consumed when the server is fully utilized, k is the fraction of power consumed by the idle server and u is the CPU utilization. We can set different P_{\max} according to the type of physical server in our simulation experiments.

In a general way, due to the variability of the workload, the CPU utilization may change over time. Therefore, the CPU utilization u is a function of time and is represented as $u(t)$. Then, the total energy consumption of a physical node E can be defined as an integral of the power consumption function over a period of time, as the following equation.

$$E = \int_{t_0}^{t_1} P(u(t))dt \quad (2)$$

3. SLA Violation Metric

QoS requirements are commonly formalized in the form of SLAs, which can be determined in terms of such characteristics as minimum throughput or maximum response time delivered by the deployed system. As these characteristics can vary for different applications, it is necessary to define a workload independent metric that can be used to evaluate the SLA delivered to any VM deployed in an IaaS such as OTF (Overload Time Fraction) metric. In this study, we use the SLA Violation (SLAV) metric which is composed of multiplication of two metrics: the SLA violation time per active host (SLATAH) and performance degradation due to migration (PDM).

$$SLAV = SLATAH \times PDM \quad (3)$$

$$SLATAH = \frac{1}{N} \sum_{i=1}^N \frac{T_{S_i}}{T_{a_i}} \quad (4)$$

$$PDM = \frac{1}{M} \sum_{j=1}^M \frac{C_{d_j}}{C_{r_j}} \quad (5)$$

where T_{si} is the total time during which the host i has experienced the utilization of 100%; T_{ai} is the total time during which the host i has been in the active state; N is the number of PMs; C_{dj} is the estimate of the performance degradation of the VM j caused by migrations which is estimated as 10% of the average CPU utilization in MIPS during all migrations of the VM j ; C_{rj} is the total CPU capacity requested by the VM j during its lifetime; and M is the number of VMs.

4. VM Placement Strategy

VM consolidation technology in clouds has become a key meaning to reduce energy consumption. The prior works use maximum CPU load of each VM as the criteria to sort and consolidate VMs. But, due to the metric changes with the computing power of the physical processor core, this method could lead to high energy consumption in cloud datacenter. The follows we will use a numerical example to illustrate the effect of this problem.

It is assumed that a cloud resource provider has three physical servers, named S1, S2 and S3. Each server has one CPU core with performance equivalent to 1000 Million Instructions Per Second (MIPS). There are five VMs, called VM1, VM2, VM3, VM4 and VM5, which have with the maximum performance of 250MIPS, 500MIPS, 1000MIPS, 750MIPS and 500MIPS respectively. Then, it is assumed that all physical servers are initially idel and the first incoming user's requirements need 100MIPS, 200MIPS, 600MIPS, 300MIPS and 200MIPS of each VM to run. If the method based on the static maximum CPU load of VM is employed to consolidate VMs, the allocation results is showed in Figure 2.

Therefore, it uses three physical servers with actual utilization of 40%, 40% and 60% respectively.

We design an assignment based on request to consolidate VMs based on the actual MIPS each VM is requested to serve. Then the final assignment result is showed in Figure 3. Hence, this assignment only employs two physical servers with actual utilization of 70% each.

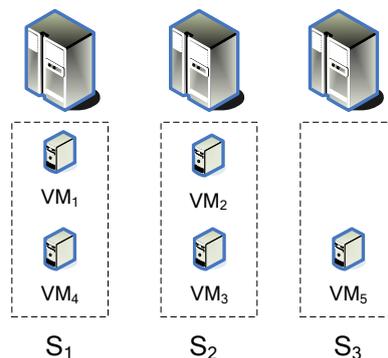


Figure 2. VM Placement based on CPU Load

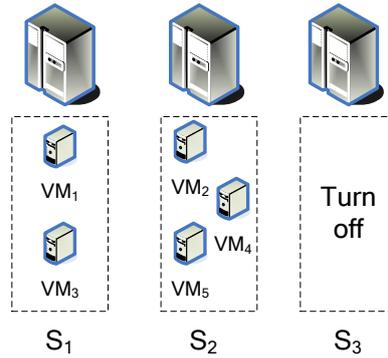


Figure 3. VM Placement based on Request

For reducing energy consumption in cloud datacenter, a VMs placement heuristic based on power-aware (VMPA) is presented to solve the placement of virtual machines. In VMPA, we first sort the VMs in decreasing order of their dynamic requested MIPS. After sorting VMs, we try to find the best server for each VM that provides the least increase of the power consumption caused by the placement. The algorithm takes full advantage of the nodes heterogeneity by selecting the most power-efficient ones first. In this step, we first find the best server among all non-underutilized and non-empty servers, if no server can be found for assigning the VM, the algorithm tries to find the best server among all underutilized servers. Finally, if it could not find among all on servers, the algorithm turns on a server from the empty-servers list to place VM on it. The pseudo-code of VMPA is showed in **Algorithm 1**.

The complexity of VMPA is $O(n \times m)$, where n is the number of physical servers that have to be deployed and m is the number of physical servers in cloud datacenter.

Algorithm 1: VMPA

1. Input: ServerList $H = \{h_1, h_2, \dots, h_n\}$, VMList $V = \{v_1, v_2, \dots, v_m\}$
2. Output: Solution of VMs placement
3. Initialize EmptyServerList E , UnderutilizedServerList U
4. SortVMsInDecreasingMIPS
5. For v_i in V
6. Initialize $P_{min} \leftarrow \text{MAX}$
7. Initialize $HV \leftarrow \text{NULL}$
8. For h_j in H
9. If h_j has no placed VMs
10. EmptyServerList $E \leftarrow h_j$
11. Else if h_j is an underutilized server
12. UnderutilizedServerList $U \leftarrow h_j$
13. Else if h_j has enough resources
14. $P(u) = k \times P_{max} + (1-k) \times P_{max} \times u$
15. If $P < P_{min}$
16. $HV \leftarrow h_j$
17. $P_{min} \leftarrow P$
18. If $HV = \text{NULL}$
19. For u_i in E
20. Return 13-17
21. If $HV = \text{NULL}$
22. For e_j in E
23. Return 13-17
24. If $HV \neq \text{NULL}$
25. $HV \leftarrow HV(v_i)$

26. Else
27. Return infeasible placement
28. Return VMs placement solution

5. Selection of VM migration

For VM migration, we consider two thresholds comprising lower and upper bounds of CPU utilization for servers so that the utilization of CPUs is kept between these thresholds. If the CPU utilization goes below this lower threshold, the server is called underutilized and all VMs have to be migrated from this host, and consequently, the host is switched to sleep mode in order to eliminate the idle power consumption. If the utilization exceeds the upper threshold, the server is called overloaded and some VMs have to be migrated from the host to reduce the utilization. This latter move leaves some headroom in the servers so that temporary spikes in the CPU usage of VMs can be accommodated without SLA violation. Two-threshold method is showed in Figure 4 and Figure 5.

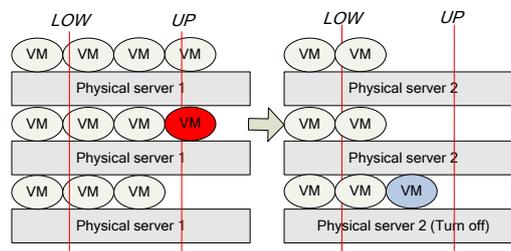


Figure 4. Upper Threshold for Overloaded Server

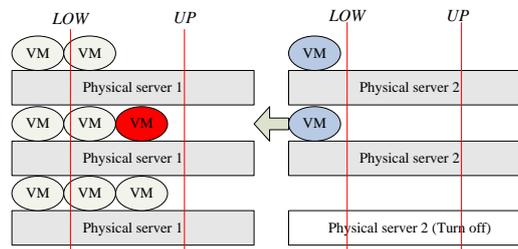


Figure 5. Lower Threshold for Underutilized Server

For choosing the best VM to migrate from an overloaded host, we present a VM migration selection algorithm called as VMMSA, which could gain minimum number of VMs migration. In VMMSA, two conditions are considered: first, the VMs which have utilization higher than the difference between the host utilization and the upper threshold are chosen; second, if a VM needs to be migrated from the host, we choose the VM that leaves the highest utilization (although below the upper threshold) on the host server. The algorithm continues until the utilization of the host becomes less than the upper utilization threshold. This policy leads to the minimum number of VMs migration in order to reduce the migration overhead.

Let V_j be a set of virtual machines currently allocated to the node j , $Q(V_j)$ is the power set of V_j (include null set and all subset of universal set). The set R of migrated virtual machines searched by VMMSA satisfies,

Minimize:

$$|R|$$

Subject to:

$$s.t. \begin{cases} R \in Q(V_j) \\ u_j - \sum_{v \in R} u_a(v) < T_u \\ Q(V_j) = \{U \mid U \subseteq V_j\} \end{cases}$$

Where, u_j is the current CPU utilization of the node j , $u_a(v)$ is the fraction of the CPU utilization allocated to the virtual machine v . The pseudo-code of VMMSA is showed in **Algorithm 2**.

Algorithm 2: VMMSA

1. Input: ServerList $H = \{h_1, h_2, \dots, h_n\}$
2. Output: MigratedVMList M
3. For h_j in H
4. Receive VMList deployed on h_j
5. VMListSortDecreasingCpuUtilization
6. Receive R_Cpu in h_j
7. Initialize $Ratio \leftarrow MAX$
8. While $R_Cpu > Upper$
9. For v_m in VMList
10. If $VM.cpu > R_Cpu - Upper$
11. $r \leftarrow VM.cpu - (R_CPU - Upper)$
12. if $r < Ratio$
13. $Ratio \leftarrow r$
14. $BestVM \leftarrow v_m$
15. Else if $Ratio == MAX$
16. $BestVM \leftarrow v_m$
17. Break
18. Update $R_cpu = R_cpu - BestVM.cpu$
19. $M \leftarrow v_m$
20. Delete BestVM from VMList
21. If $R_Cpu < Lower$
22. $M \leftarrow$ all VMs on h_j
23. Return M

6. Flow and Complexity of our Algorithm

In our algorithm design, we first only deal with the VMs on overloaded physical servers and the newly arrived VMs, and after placing them on physical servers, we relocate VMs from underutilized physical servers to turn them off. Then, we select some appropriate VMs from overloaded servers and consolidate them with newly arrived VMs as inputs for VMs placement algorithm block at the first step. Finally, we find all VMs of underutilized servers to migrate into the best other servers using VM placement algorithm, which would lead to switch idle servers to sleep mode to reduce the total power consumption of cloud datacenter. Figure 6 shows the flow of our algorithm.

About the complexity of our proposed algorithm, it is assumed that N denote the number of all physical servers, M denotes the number of VMs deployed on the physical servers, the number of newly arrived VMs is H , the number of overloaded servers is L , the number of underutilized servers is K , the complexity of VMPA is $N \times M$, the complexity of VMMSA is N . Therefore, the complexity of our algorithm is $2N + H + N \times (M + H) + N \times M + K + N \times M$, that is $O(3N \times M + N)$.

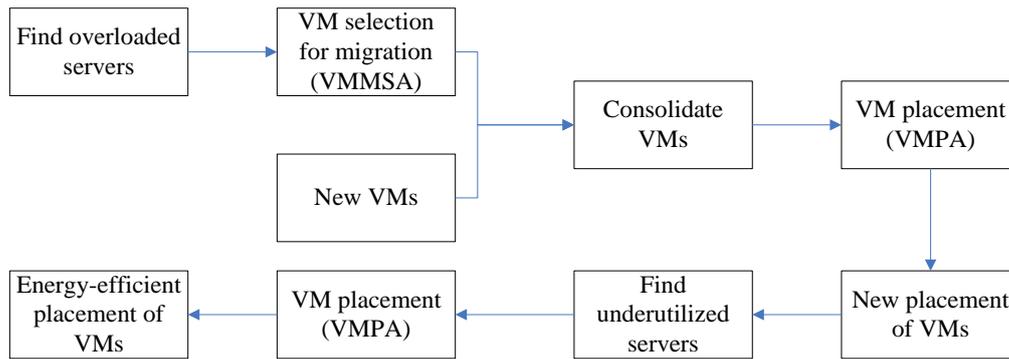


Figure 6. Flow of our Strategy

7. Simulation Experiments and Analysis

For evaluating the performance of our proposed algorithm, we will develop some simulation experiments of employing our VM placement strategy in toolkit CloudSim [11] in this section. The cloud environment includes a data center, which consists of N heterogeneous physical nodes. According to different process power of CPU, these physical nodes are divided equally into two types, represented by (1800MIPS, 2600MIPS). Two types of physical nodes both include two process elements (CPU core), 4G RAM and 1GB/s network bandwidth. Each virtual machine requires one CPU core with 2500, 2000, 1500, 500MIPS, 0.8, 1.7, 1.5, 0.6BG of RAM and 100MB of bandwidth. Each virtual machine runs a web-application or any kind of application with variable workload, which is modeled to generate the utilization of CPU according to a uniformly distributed random variable. The application runs for 150000MI that is equal to 10 min of the execution on 250 MIPS CPU with 100% utilization. Initially, all virtual machines are allocated according to the requested characteristics assuming 100% CPU utilization. Four different VM placement strategies are tested in our simulation experiments as below:

STA: Single Threshold Algorithm

DVFS: Dynamic Voltage and Frequency Scaling [12]

MPA: Maximum Power Algorithm

VMPA-VMMSA: Our algorithm proposed in this paper

Experimental results will be showed from the following four metrics:

TEC: Total Energy Consumption

SLAV: SLA Violations

VMMN: VM Migration Number

ANAS: Average Number of Active Servers

And, to evaluate the two-threshold setting policy in VM migration, we have conducted several experiments with different values of the two thresholds.

Experiment 1 we observe the performance of the above four metrics in Figure 7, Figure 8, Figure 9 and Figure 10. Figure 7 shows total energy consumption of four algorithms. MPA consumes most energy due to keep 100% CPU utilization of all physical servers regardless of deploying VMs on servers. DVFS can dynamically adjust the voltage and frequency of CPU according to the CPU utilization for decreasing the power consumption of physical nodes, which could reduce some energy consumption. STA is based on the idea of setting the upper utilization threshold for servers and placing VMs, while keeping the total utilization of servers below this threshold. However, STA could lead to some underutilized servers that placing less number of VMs. Different from STA, our algorithm VMPA-VMMSA employs two-threshold method including lower and upper threshold for consolidating VMs. When the utilization of the server is lower than the lower threshold, all VMs on the server will be migrated to other servers and the server will be turn off, which could lead to less total energy consumption in cloud datacenter. In

Figure 7, DVFS and MPA have less SLA violation since amount of requested resource equals to the amount of allocated resource in two algorithms. VMPA-VMMSA has less SLA violation than STA due to two-threshold method. Since two-threshold method would activate less physical servers, VMPA-VMMSA make most VM migration as showed in Figure 9. Based on similar reasons, our algorithm can use least number of activated physical servers, as showed in Figure 10.

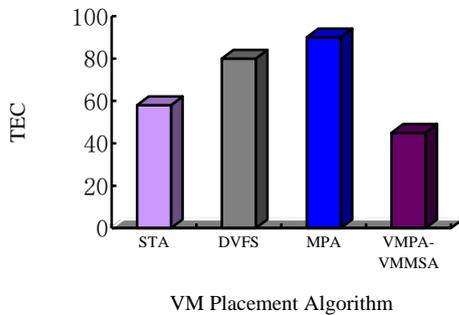


Figure 7. Total Energy Consumption

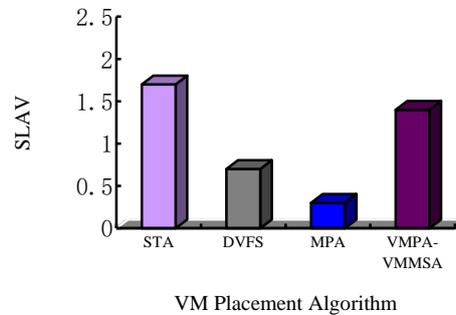


Figure 8. SLA Violation

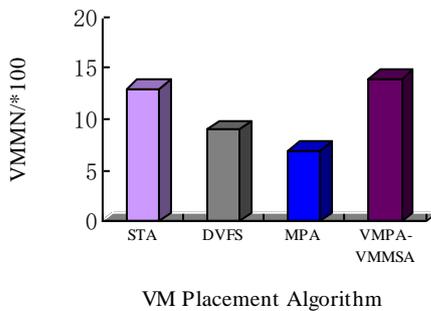


Figure 9. Number of VM Migration

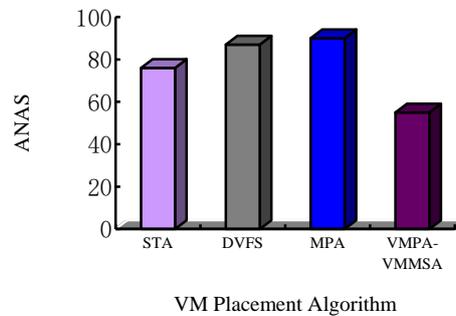


Figure 10. Average Number of Active Servers

Experiment 2 we observe the two thresholds impact on the performance of algorithm in Figure 11-Figure 12. As the upper threshold increases, the total energy consumption will reduce since less servers will be activated. And, this will lead to more VMs consolidation and more SLAV, as showed in Figure 12. The lower threshold impacts on the performance with a smaller degree, but the trend will be same as the upper threshold.

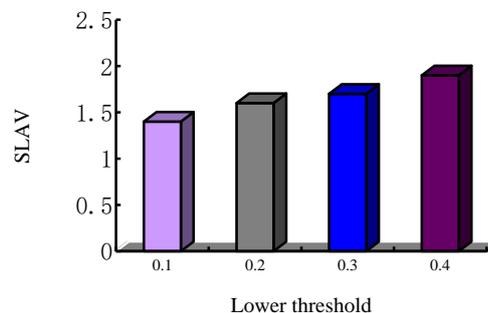
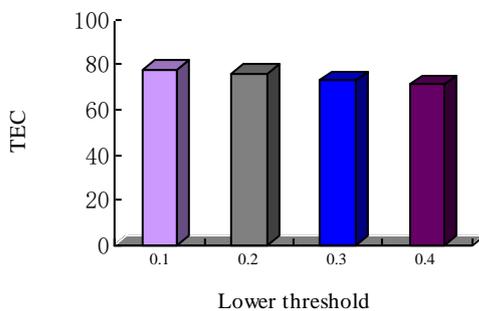


Figure 11. Increasing Lower Threshold Impacts on the Performance

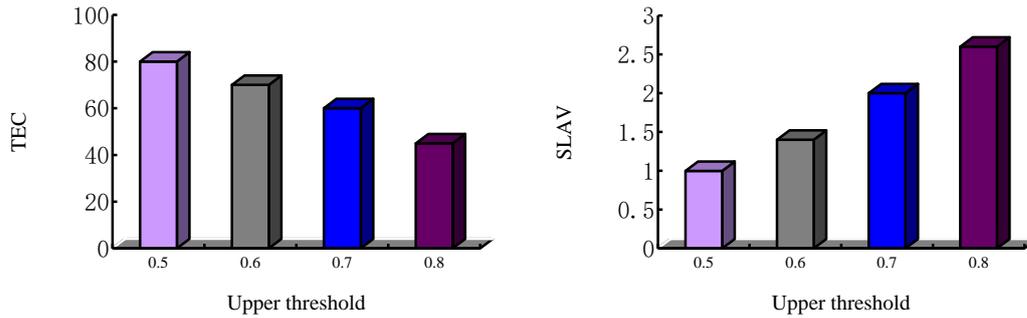


Figure 12. Increasing Upper Threshold Impacts on the Performance

Experiment 3 we compare single threshold with two-threshold method in Figure 13. Because single threshold method only sets the upper threshold to ensure servers not overload, which would lead to some underutilized servers and more energy consumption. Two-threshold can consolidate VMs to less servers via VM migration by setting lower threshold, which reduce the energy consumption. Meanwhile, two-threshold method can more reduce SLA violation than single-threshold method.

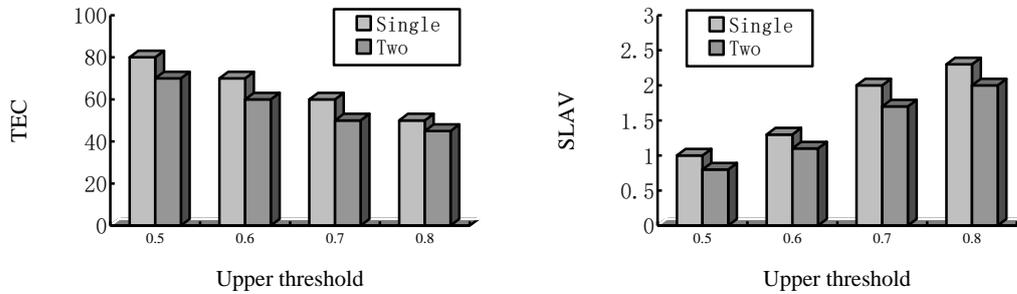
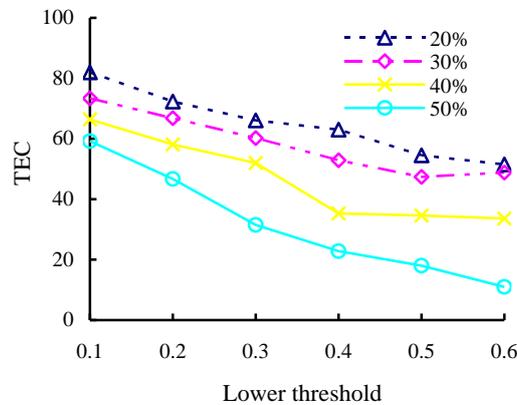


Figure 13. Single Threshold vs Two Thresholds

Experiment 4 we observe the interval between lower and upper threshold in two-threshold impacts on the performance of our algorithm. Figure 14 shows that an increase of the lower threshold leads to an increase of SLA violation and a decrease for all intervals. And, for the same lower threshold, the high the upper threshold, the less the energy consumption and the more the SLA violation, this is because the bigger interval between thresholds could consolidate more VMs to less servers via VM migration.



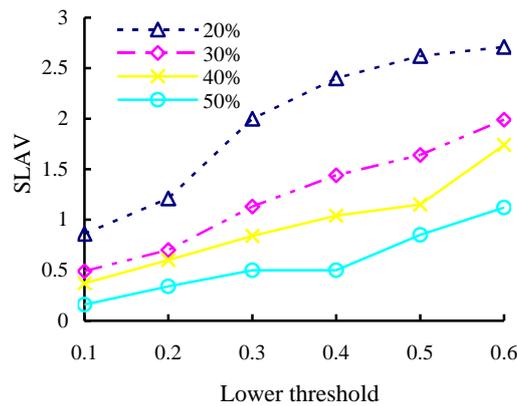


Figure 14. Interval between Lower and Upper Threshold in Two-threshold Method

8. Conclusion

Dynamic consolidation of virtual machines using live migration and switching idle nodes to the sleep mode allows cloud resource providers to optimize resource usage and reduce energy consumption. But, the obligation of providing high reliable QoS to users leads to the necessity in dealing with the energy-performance trade-off, because some aggressive consolidation can bring about performance degradation. Aiming at solving this problem, an adaptive VM placement algorithm VMPA-VMMSA is proposed in this paper. The new algorithm can carry out dynamic consolidation and migration of virtual machines and greatly reduce energy consumption, while ensuring a high level of adherence to the SLA and reducing the number of VM migration. The results of simulation experiments show that the algorithm is feasible and efficient.

Acknowledgments

The work was supported by the project of Science and technology research of henan province (152102210202), Institutions of higher learning key research projects in henan province department of education (15A520094), Science and Technology Innovation Fund of Xinxiang University (15ZA03).

References

- [1] M. Cusumano, "Cloud Computing and SaaS as New Computing Platform", *Communication of the ACM*, vol. 53, no. 4, (2010), pp. 27-29.
- [2] S. S. Manvi, G. K. Shyam, "Resource management for Infrastructure as a Service (IaaS) in cloud computing: a survey", *Journal of Network Computer Applications*, vol. 41, (2013), pp. 424-440.
- [3] Y. Gao, H. Guan and Z. Qi, "Service Level Agreement Based Energy-efficient Resource Management in Cloud Data Centers", *Computer & Electrical Engineering*, vol. 40, no. 5, (2013), pp. 1621-1633.
- [4] A. Beloglazov, R. Buyya, Y. C. Lee and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems", *Adv Comput* vol. 82, (2011), pp. 47-111.
- [5] D. Kusic, J. O. Kephart and J. E. Hanson, "Power and Performance Management of Virtualized Computing Environments via Lookahead Control", *Cluster Computing*, vol. 12, no. 1, (2009), pp. 1-15.
- [6] R. Nathuji and K. Schwan, "Virtual Power: Coordinated Power Management in Virtualized Enterprise Systems", *ACM SIGOPS Operating Syst Rev*, vol. 41, (2007), pp. 265-78.
- [7] W. Xiaorui and W. Yefu, "Coordinating Power Control and Performance Management for Virtualized Server Clusters", *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 2, (2010), pp. 245-259.
- [8] A. Horri, M. S. Mozafari and G. Dastghaibfyard, "Novel resource allocation algorithms to performance and energy efficiency in cloud computing", *Supercomput* vol. 69, (2014), pp. 1445-61.
- [9] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy and G. Jiang, "Power and Performance

- Management of Virtualized Computing Environments via Look Ahead Control, *Clust Comput*, vol. 12, (2009), pp. 1-15.
- [10] X. Wang and Z. Liu, "An energy-aware VMs placement algorithm in cloud computing environment", *Proceedings of the Second International Conference on Intelligent System Design and Engineering Application*, (2012).
- [11] C. Rodrigo, R. Rajiv and B. Anton, "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, (2011), pp. 23-50.
- [12] G. Magklis, G. Semeraro and D. H. Albonesi, "Dynamic Frequency and Voltage Scaling for A Multiple-clock-domain Microprocessor, *IEEE Micro*, vol. 23, no. 6, (2003), pp. 62-68.

Authors



Lijun Xu, he is currently a Ph.D. student in the School of Computer Science and Technology at Wuhan University of Technology and an associate professor at the Xinxiang University. He received his B.S. from Henan Normal University, Xinxiang, Henan, China, in 2001, and M.S. from the Wuhan University of Technology, Wuhan, Hubei, China, in 2007. He has worked at the Xinxiang University since 2001. His research interests include resource management in cloud computing environments and Computer network. He has published about 10 papers.