Utilization-based VM Consolidation Scheme for Power Efficiency in Cloud Data Centers

Qi Chen, Jianxin Chen, Baoyu Zheng, Jingwu Cui
Key Lab of Broadband Wireless Communication and
Sensor Network Technology of Ministry of Education
Nanjing University of Posts&Telecommunications,
Nanjing, China 210003
Email: chenqi199401@gmail.com,
{chenjx, zby, cuijw}@njupt.edu.cn

Yi Qian
University of Nebraska-Lincon
Omaha, Nebraska 68182-0572, USA
Telephone: (402) 554-4990
Email: yqian@ieee.org

***Abstract*—Cloud computing offers utility-oriented services to users, which is supported by large-scale data center. Although virtualized data centers provide high performance computing service, they also consume enormous amount of power. To solve the problem, dynamic consolidation of Virtual Machines (VMs) is considered as an efficient way to reduce power consumption and guarantee Quality of Service (QoS). Live migration is applied into the dynamic consolidation, which allows VMs to be migrated to other hosts and aims to minimize the number of hosts in data centers. However, the migration overhead is essential to be taken into account and massive migrations will lead to performance degradation and extra power consumption. In this paper, we propose a utilization-based migration algorithm (UMA) to migrate VMs to stable hosts, which efficiently reduces migration time and power consumption. Experiment results show that our UMA can reduce about 77.5%-82.4% migrations and save up to 39.3%-42.2% power consumption compared with the MinPower policy.**

I. Introduction

Cloud computing provides infrastructure, platform and software as a service, which provisions resources to users on demand. Meanwhile, those services are supported by large-scale data centers, which not only consume enormous amounts of power but also lead to substantial carbon dioxide emission contributing to the greenhouse effect. What's more, current studies have found that in data centers the utilization of hosts ranges over 10% to 50% in most of time [1].

In order to improve the power efficiency in data center, developing an effective management strategy has already become an urgent issue. Nathuji et. al. [2] implemented Virtual Power Management (VPM) to control and coordinate various power management policies and systematically managed the data center with the local and global level. However, it did not take migration cost into account. There are also some studies [3] [4]which used the historical data to predict the system state and decide whether or not to reallocate resource. Beloglazov et. al. [5] applied MinPower-based greedy algorithm to migrate VMs among hosts which yields the minimal power consumption. However such solution causes a large amount of migrations and results in low resource utilization. Fig. 1 depicts the resource utilization of MinPower. It can be noted that 58% hosts operate with low utilization, i.e. less than 50%, and 17% hosts operate with utilization over 50%-70%. Although in [6] they improved the overload detection and maximized the mean time of migrations from overloaded hosts so that it can improve the quality of VM consolidation, the low utilization problem still exists.

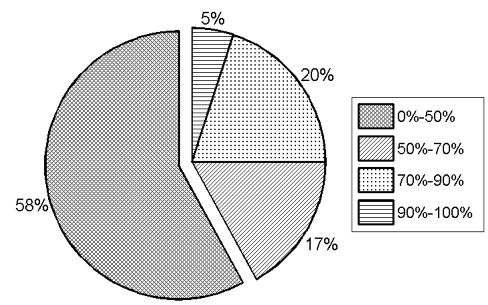


Fig. 1. Resource Utilization of MinPower Solution

In this paper, we propose to reduce the power consumption by improving the resource utilization of hosts and reducing the migrations by using a utilization-based migration algorithm. The main contributions of this paper are the following:

- We propose a utilization-based VM migration framework for cloud computing.
- We define the performance function and design a utilization-based migration scheme to optimize the VM placement in terms of performance function, which consolidates VMs to improve the power efficiency so as to guarantee the QoS.
- We evaluate the scheme by simulations and the results show that 10% hosts have low utilizations compared to 58% of MinPower policy.

The remainder of this paper is organized as follows. In Section II we introduce the system model for VM consolidation. Section III describes UMA algorithm in detail. In Section IV we perform extensive simulations and show that our scheme outperforms MinPower policy. Finally the conclusion is drawn in Section V.

II. Problem Formulation

Before we formulate the problem, we give a discussion on the background and define some metrics for cloud data center.

A. Power Consumption

In cloud data center, several main factors affect power consumption in hosts, e.g. Central Processing Unit (CPU), memory, and disk storage. Among them, CPU plays a key role. Therein we focus on the CPU resource. Previous study [7] has shown that the CPU utilization has a linear relationship with the power consumption. The power consumption by a single host can be defined as

$$P(u) = P_{\min} + (P_{\max} - P_{\min}) \cdot u, \tag{1}$$

where $P_{\max}$ is the power consumption at 100% utilization, $P_{\min}$ is the power consumption at 0% utilization and u is the current CPU utilization. As we have discussed above [1], the power consumption is about 70% when the utilization is 0. Thus the hosts with low utilization do not mean low power consumption.

B. Service Level Agreement (SLA) Violation

To guarantee the specific Quality of Service (QoS) during the operations, consumers negotiate with cloud provider and request for specific SLA. Provider will pay penalties to consumers if SLA violation happens. Therefore SLA violation is another important metric that needs to be taken into account. In [5], a metric of average SLA violation percentage was defined as

$$SLA = \frac{\sum_{j=1}^{n} \int (U_{rj}(t) - U_{aj}(t))dt}{\sum_{j=1}^{n} \int U_{rj}(t)dt}, \tag{2}$$

where U_{rj} and U_{aj} are the requested and allocated resources (e.g. Million Instructions Per Second (MIPS)) by the j^{th} VM. Herein we use this definition to evaluate the SLA violation degree during resource allocation.

C. Performance Function

There are several factors affecting the cost of data center, such as power, SLA violation, migration, communciation and cooling system. The live migration cost is related to the entire amount of memory of VM [14]. And the communication resources and cooling system are amortized over the hosts. Therein these two factors are also assumed to be relatively independent of the host's workload [13]. Then we define the cost of a host related to utilization u as

$$C_u = C_{P_u} + C_{SLA_u}, \tag{3}$$

where C_{P_u} is the power cost, C_{SLA_u} is the cost for SLA violation. Let Θ represents the electricity charge, from Eq.1 we have $C_{P_u} = \Theta * P(u)$. According to the simulation results of SLA violation in [12], we define the SLA violation cost as a quadratic function related to the host utilization u, i.e.

$$C_{SLA_u} = \begin{cases} 0, u \le u_0 \\ \alpha * (u - u_0)^2, u_0 < u \le 1 \end{cases} \tag{4}$$

Here α is a constant for SLA cost. When the utilization of host is below u_0, SLA cost is zero.
Normalize C_u, the normalized function is expressed as

$$f(u) = \begin{cases} \Theta \cdot (\Delta P + P_{\min}/u), 0 \le u \le u_0 \\ \Upsilon + \frac{\Theta P_{\min} + \alpha {u_0}^2}{u} + \alpha u, u_0 < u \le 1 \end{cases}$$
$$\Delta P = P_{\max} - P_{\min}, \Upsilon = \Theta \cdot \Delta P - 2\alpha u_0. \tag{5}$$

Here $f(u)$ is a function related to the tradeoff between the performance and cost, where Υ is a constant. From it we can find that $f(u)$ reaches the minimum when $u^2 = \Theta P_{\min}/\alpha + {u_0}^2$, i.e. $u_{opt} = \sqrt{\Theta P_{\min}/\alpha + {u_0}^2}$. The smaller the function value is, the more profit the cloud provider will get. From this point, cloud provider prefers to letting hosts work with $f(u)$ approximating to the minimum to increase their profits. Previous research result [12] shows that the best tradeoff between the power consumption and the SLA violation penalty can be obtained with the utilization of 70%.

D. Problem Formulation

According to the above discussion, to achieve high performance while achieving high efficient power consumption, either underload or overload hosts are not preferred to cloud data center. Combined to the requirement of SLA, how to control VM consolidation among hosts with less migrations will be a new challenge for cloud data center. This problem can be describe as

$$\begin{aligned} &min \quad \sum_{i=1}^{n} x_i \cdot f(u)_i, \\ &s.t. \quad x_i = 0 \quad or \quad 1. \end{aligned} \tag{6}$$

Here $x_i = 0$ or 1 denotes that the i^{th} host is turned off or on. $f(u)_i$ denotes the performance-related function value of host i. Thus the VM migration problem becomes an optimization problem. In the next section, we will formulate and solve the problem in details.

III. Utilization-based Migration Algorithm

A. Migration Probability

In [8], the authors proposed a probabilistic method to select VMs from overload hosts, which effectively avoids aggressive VM migrations. A migration probability function was defined as follows

$$f_m(u) = \left(1 + \frac{u-1}{1-T_h}\right)^{\lambda}, \tag{7}$$

where u is the CPU utilization and $f_m(u)$ is the migration probability. Here T_h is a threshold for high load and λ is a constant. We will use this definition to optimize and improve the consolidation algorithm.

B. Available Hosts

Available hosts are the set of hosts that VMs could be migrated to, these consisting of underload hosts and idle hosts. The decision equation is defined as follows

$$\chi * (R_u + n * R_H * T_h) \ge R_r, \tag{8}$$

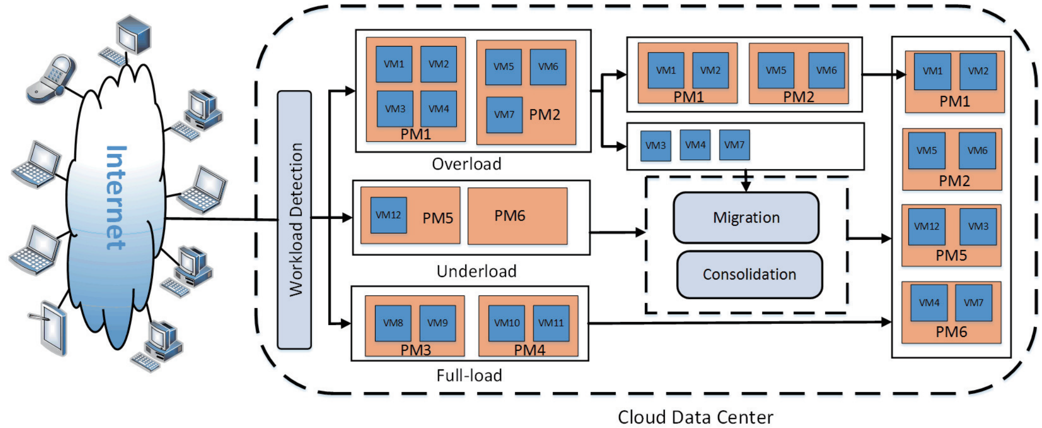


Fig. 2. A Utilization-based VM consolidation Framework

$$R_u = \sum_{i=1}^{m} (T_h - u) * R_{UH_i}, \tag{9}$$

where R_u is the total available CPU resource in MIPS from underload hosts, R_H is the CPU resource capacity for one host, R_r is the requested CPU resource from VMs to migrate. n represents the number of idle hosts needed to be switched on. The parameter χ decides whether to turn on more idle hosts or not, and we set it as 0.7 in our system. In Eq.9, m is the number of underload hosts in the data center and R_{UH_i} is the available resource for the i^{th} underload host.

C. VM consolidation Framework

Fig.2 depicts a framework of our proposed solution to the above problem. In this system, a workload detection module classifies the hosts into three categories: overload, full-load and underload. Then VMs on those hosts with overload are computed migration probability and choose the candidates to the queue waiting for migrations. The VMs on underload hosts will be consolidated to improve the utilization of hosts. All hosts with full-load will not be changed. With this solution, hosts in the cloud data center might achieve the optimal performance while yield high power efficiency. Below we will discuss the details of VM placement.

D. Best Fit with Decreasing (BFD)

The VM placement problem, which is mapping VMs to hosts, is considered as a bin packing problem. With the objective to optimize VM placement, we let the utilization of overload and underload hosts work toward the optimal performance-related utilization. Aiming to this target, we use BFD [9] to find the local optimum solution that those VMs are consolidated to the underload hosts whose utilization will approach to the optimal performance after deploying. It has been proved that BFD uses no more than $\frac{11}{9}OPT+1$ hosts, where OPT is the optimal solution.

Algorithm 1 BFD Algorithm

according to Eq.8 add $idlehosts$ into $availableHost$
SortDecreasingUtiliztion($VMsToMigrate$)
for all $vm \in VMsToMigrate$ **do**
    $min \leftarrow MAX$, $allocated \leftarrow null$
    $meanMips \leftarrow vm.getRequestMips()$
    for all $host \in availableHost$ **do**
        $Value \leftarrow f(host,meanMips)$; //Compute the Performance-related utilization
        if $Value < minValue$ **then**
            $min \leftarrow Value$, $allocatedHost \leftarrow host$
        end if
    end for
    put ($allocatedHost$,vm) into $MigrationMap$
end for
return $MigrationMap$

Algorithm 1 depicts the procedure of BFD algorithm for VM migration. Firstly, all VMs waiting for migration (VMsToMigrate) are sorted with decreasing order in a queue according to their requested MIPS. For each VM, it is allocated to the hosts with the optimal performance according to the definition of performance function in Eq.5. Then all VMs are consolidated sequentially. BFD provides a local optimization solution, but it does not perform well especially when some idle hosts are switched on as available hosts. In these scenarios, VMs prefer to migrating to those hosts with utilization exceeding U_{opt} than idle hosts, which results in that the idle hosts highly under utilization. Fig.3(a) depicts the utilization of hosts after BFD migration scheme. Here without loss of generality, there are four available hosts with utilizations of 50%, 60%, 65% and 0% respectively in a homogeneous environment. Five VMs are waiting for migration with MIPS requests for 20%, 20%, 15%, 15% and 10% CPU utilization. It can be noted that the load among all used hosts are preferred to overload, e.g. hosts 2 and 3 have overload about 80%. To solve this problem, we propose a utilization-based migration algorithm in the next section.

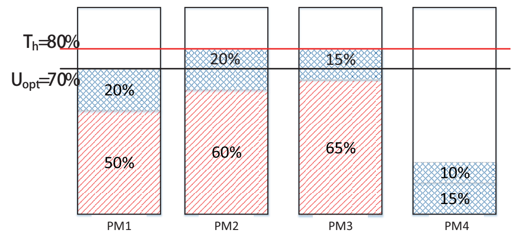
Algorithm 2 Utilization-based Migration Optimization

$s \leftarrow MigrationMap$, $sBest \leftarrow s$, $tabuList \leftarrow null$
while (not stoppingCondition()) **do**
    put several migrations not containing in $TabuList$ into $migrationList$
    $migration \leftarrow$ getBestMigration($migrationList$)
    $s \leftarrow$ getMigrationMap($s, migration$)
    if FunctionValue(s) $<$ FunctionValue($sBest$) **then**
        $sBest \leftarrow s$
    end if
    $tabuList \leftarrow tabuList + migration$
    if Size($tabuList$) $>$ maxTabuListSize **then**
        removeFirsElement($tabuList$)
    end if
end while
return s

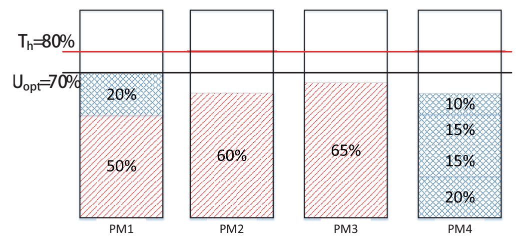
E. Utilization-based Migration Algorithm

For the optimization problem in Eq.6, we use Tabu Search to find the optimal solution when some idle hosts are turned on for migration destination. Firstly we use BFD for a pre-migration for all VMs. Thus according to Algorithm 1, a table of VMs to hosts mapping is created. After that, we optimize this VMs to hosts maps table using Tabu Search algorithm. Tabu Search uses a local search procedure to iteratively move from one potential solution according to Eq.5 to an improved solution in the neighborhood. In Tabu Search algorithm, if a potential migration has happened previously within a short-term period, this migration is put into the tabu list so that the algorithm does not consider this possibility repeatedly. Then after a number of iterations, the optimal solution is obtained. With this tabu list, the algorithm can achieve the global optimal solution to the problem in Eq.6 . The detailed implementation is illustrated in Algorithm.2.

Fig.3(b) depicts the utilization after VM consolidation. It is noted that utilizations of all hosts are 70%, 60%, 65%, 60% respectively, which is more approximating to the full-load, much better than that of BFD algorithm.



(a) MigrationMap After Executing BFD



(b) MigrationMap Optimized By UMA

Fig. 3. Utilization after UMA

TABLE I. Workload Data

VMs	500	750	1000	1250	1500	1750	2000
Avg. Util.(%)	12.7	12.2	12.1	11.3	11.3	11.2	11.1

IV. Performance Analysis

A. Experimental Environment

To verify the efficiency of our proposed algorithm, CloudSim toolkit [11] is chosen for our simulations, in which we adopt the real workload data provided as part of the CoMon project [10]. Here a data center is built consisting of two type of hosts: HP ProLiant ML110 G4 with 1860 MIPS each core, and HP ProLiant ML110 G5 with 2660 MIPS each core. At the same time, there are four types of VMs which is correspond to Amazon EC2 instance types: Medium, Large, Small and Micro. During the experiment, we change the number of VMs from 500 to 2000 to analyze our algorithom. The relative average utilization for different number of VMs is listed in Table.I.

B. Power Consumption

We compared our UMA algorithm with MinPower [5] using different overhead thresholds (e.g. 0.8 and 0.9) for eight days real workload shown in Fig.4. Specifically on 22 April, UMA-0.8, UMA-0.9, MinPower-0.8 and MinPower-0.9 consume 115.88, 111.19, 201.80 and 188.84 Kwh respectively. UMA-0.8 and UMA-0.9 improve power efficiency about 42.2% and 41.1% compared with MinPower-0.8 and MinPower-0.9. From the data over those eight days, UMA can reduce 39.3%-42.2% power consumption, which verifies the power efficiency of our VM consolidation algorithms.

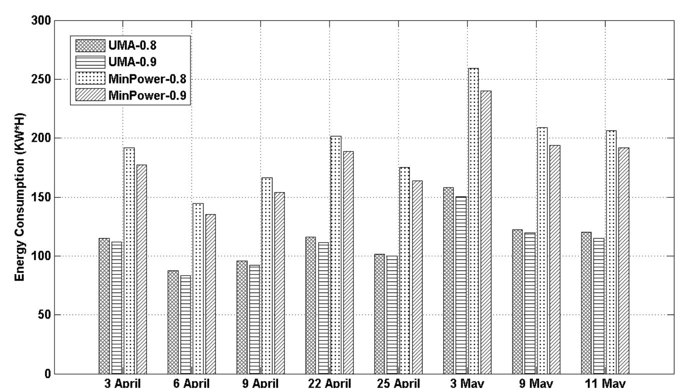


Fig. 4. Power Efficiency

C. Resource Utilization

To better consolidate VMs and save power energy, the utilization of hosts should be close to OPT-utilization according to Eq.5, which is able to tradeoff between the power consumption and SLA violation. As discussed above in Fig.1, MinPower policy results in that most hosts are underload. With the same workload, Fig.5 depicts the CPU utilization of hosts under UMA algorithm. It can be found that the hosts with low utilization is very few, about 10%, which is rather smaller than that of MinPower policy. Besides that most of hosts work with utilization between 70% and 90%, which shows that UMA can effectively use the CPU resource.

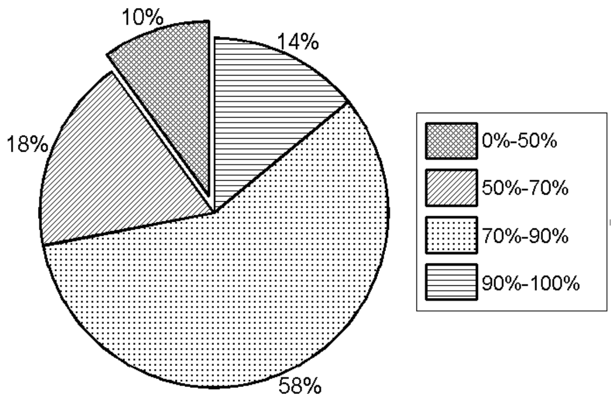


Fig. 5. Resource Utilization In UMA

D. Active Hosts

Generally, VMs should be consolidated into the minimal hosts. Fig.6 depicts the hosts used compared with the optimal solution and MinPower. With the increasing of VMs, the performance ratio of UMA to OPT converges to 1.137, which is better than BFD. With the same workload, the MinPower policy seriously increases the amount of active hosts. The result shows that UMA can reduce 39.2%-45.7% active hosts compared with MinPower policy.

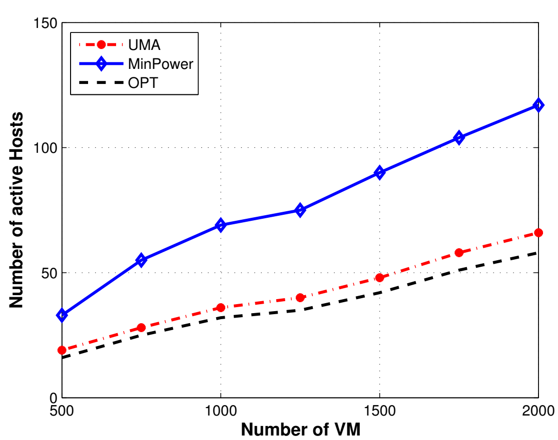
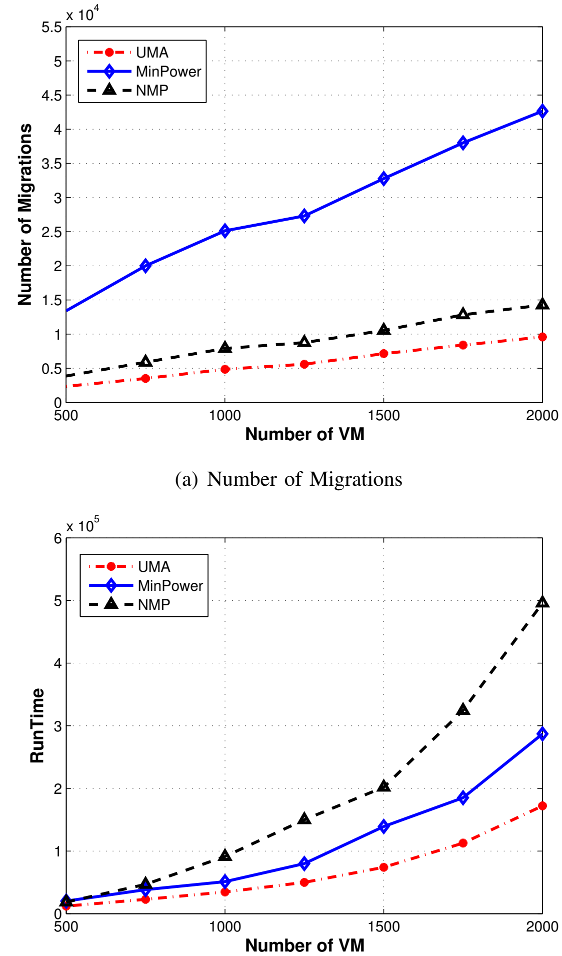


Fig. 6. Number of Active Hosts

E. Number Of Migrations

Fig.7(a) depicts the VM migrations among hosts. Compared with MinPower and the algorithm without migration probability (NMP), UMA could avoid arbitrary migrations



(a) Number of Migrations

(b) RunTime

Fig. 7. Simluation Result on Migration and Runtime

and reduce migrations about 77.5%-82.4%. In addition, The complexity of BFD [5] is $O(n \cdot m)$, where n is the amount of available hosts and m is the number of VMs that need to be migrated. Our UMA method minimizes both the number of migrations and available hosts to shorten the runtime. Fig.7(b) showed that our proposed algorithm obtains 32.4%-46.9% reduction in runtime.

F. SLA Violation

TABLE II. SLA Violation Experiment Result

Date	UMA-0.8	UMA-0.9	Min-0.8	Min-0.9
03/03/2011	9.85%	10.23%	10.14%	10.56%
06/03/2011	10.13%	10.41%	10.13%	10.53%
09/03/2011	10.15%	10.67%	10.25%	10.78%
22/03/2011	9.93%	10.29%	10.25%	10.60%
25/03/2011	9.97%	10.27%	10.11%	10.35%
03/04/2011	10.17%	10.35%	10.27%	10.42%
09/04/2011	10.06%	10.11%	10.09%	10.24%
11/04/2011	9.85%	10.24%	10.08%	10.38%

Table.II shows the average SLA violation for the real workload. The results show that UMA slightly alleviates the SLA violation. Compared to MinPower, UMA works a little better than MinPower algorithm when the overload threshold is equal to 0.8, and very similar when the overload threshold is equal to 0.9 in terms of SLA violations.

V. Conclusions and Future Work

In this paper we investigated the power-efficient VM consolidation schemes, which aims to cluster VMs into as less hosts as possible. However, such consolidation is not easy due to the varying workload and the inherent complexity of the bin packing problem. We define the performance function and design a utilization-based migration scheme to optimize the VM placement in terms of performance function, which consolidates VMs to improve the power efficiency so as to guarantee the QoS. We evaluated our proposed algorithm with a large-scale experiment using real workload from CoMon. The experiment results show that our proposed scheme could efficiently reduce about 77.5%-82.4% migrations and save up to 39.3%-42.2% power consumption compared with the MinPower policy. What's more, in this paper we just applied static utilization threshold to decide which hosts are overload. However, the static utilization threshold does not change with varying workload, and historical workload contains a lot of information worth mining. Thus, we will pay more attention on overload detection algorithm in future work.

Acknowledgements

This work is partly supported by the Wireless Broad Communication and Wireless Sensor network, Embedded system Department of Education Key Lab Open Program (Grant No. NYKL201306, TK214001), Education Natural Science Foundation of Jiangsu Province (13KJB510026), and Nanjing University of Posts and Telecommunications Foundation (Grant No. NY211032).

References

[1] G. Dasgupta, A. Sharma, A. Verma, et al. "Workload management for power efficiency in virtualized data centers". *Communications of the ACM*, vol. 54, no.7, pp.131-141, 2011.

[2] R. Nathuji, K. Schwan. "VirtualPower: Coordinated power management in virtualized server systems." *In 21st Symposium on Operating Systems Priniples(SOSP)*, vol.41, no.6, pp.265-278, 2007.

[3] Z. Xiao, W. Song, Q. Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment." *In IEEE transactions on parallel and distributed systems*. vol. 24, no. 6, pp.1107-1117, 2013.

[4] L. Zhou, Z. Yang, H. Wang, and M. Guizani, Impact of Execution Time on Adaptive Wireless Video Scheduling, IEEE Journal on Selected Areas in Communications, vol. 32, no. 4, pp. 760-772, 2014.

[5] A. Beloglazov , R. Buyya, "Optimal Online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers", *concurrency and computation: practice and experience (CCPE)*. vol.24, no.13, 2011.

[6] A. Beloglazov, and R. Buyya, "Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers Under Quality of Service Constraints." *In IEEE transactions on parallel and distributed systems*, vol.24, no.7, pp.1366-1379, 2013.

[7] D. Kusic, J. Kephart, J. Hanson, et al. "Power and performance management of virtualized computing environments via lookahead control". *Cluster Computing 2009* vol.12, no.1, pp.1-15, 2009.

[8] C. Mastroianni, M. Meo, G. Papuzzo, "Probabilistic Consolidation of Virtual Machines in Self-Organizing Cloud Data Centers." *In IEEE transction on cloud computing*, vol. 1, no. 2, 2013.

[9] M, Yue, "A simple proof of the inequality FFD(L)$\leq$11/9OPT(L)+1, for all L for the FFD bin-packing algorithm." *Acta Mathematicae Applicatae Sinica (English Series)*, vol.7, no.4, pp.321-331, 1991.

[10] K. Park , VS Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab." *ACM SIGOPS Operating SystemsReview*, vol. 40, no. 1, pp. 65-74, 2006.

[11] R. Calheiros, R. Ranjan, A. Beloglazov, et al. "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.

[12] B. Rajkumar, B. Anton, and A. Jemal, "Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges", *in PDPTA 2010 United States of America* pp. 6-17, 2010.

[13] Goudarzi H, Ghasemazar M, Pedram M. "Sla-based optimization of power and migration cost in cloud computing", *Cluster, Cloud and Grid Computing (CCGrid), 2012*, pp.172-179, 2012.

[14] Liu, Haikun, et al. "Performance and energy modeling for live migration of virtual machines." *Cluster computing 16.2*, pp.249-264, 2013.