Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **6**, November **2018**

Resource Scheduling Method Based on Bayes for Cloud Computing

Juan Wang[1], Xiuyan Sun[1], Wenmin Song[1] and Linlin Tang[2]

[1]Department of Information EngineeringLaiwu Vocational and Technical College
[2]Harbin Institute of Technology, Shenzhen
Email: hittang@126.com

Received May, 2018; Revised September, 2018

Abstract. *As well-known for us all, Cloud Computing (CC) has become more and more popular for being spurred by business applications. In the view of energy shortage, more and more companies focus on green cloud. At the same time, they also should ensure lower Service Level Agreement (SLA) Violation for users. For this, a new resource scheduling policy which combines Bayes Visual Machines (VM) selection method and the improved VM placement algorithm are proposed here. Experimental results prove that our improved algorithm excels to popular bin-packing algorithms. It can tremendously decrease SLA violation.*
Keywords: Cloud Computing; Resource Scheduling; Bayes theory; Energy Efficiency; CloudSim

1. **Introduction.** With the rapid development of Internet, Cloud Computing [1] has been paid more and more attention as a new trend. It is experiencing a rapid development both in academy and industry [2], tremendously promoted by the business rather than academic. Since it is a kind of business model, many companies invest more resources to this field[3]. For greenhouse effect and energy consumption problem becoming prominent, many companies want to optimize their cloud system to reduce energy consumption [4]. Generally speaking, there are four indicators to weigh a resource algorithm at resource scheduling level, which are Energy consumption, number of VM migrations, overall Service Level Agreement (SLA) violation and mean execution time. Many resource algorithms perform well only at one or two indicators [5]. Few of them do well at all four. For example, in Sahar Sohrabis work [6], she introduced the Bayes theory into VM selection. But the algorithm was said that could not cut down SLA violation effectively and the future work is to improve this. In our work, a new comprehensive resource scheduling algorithm which uses the modified virtual machine placement algorithm (UPCBFD) based on sorted hosts in place of the original BFD algorithm proposed in Beloglazovs paper [7] is proposed here. After experiments, our proposed algorithm can be proved to decrease SLA violation and energy consumption.

The rest is organized as follows. In Section 2, some related work is proposed. In Section 3, the comprehensive resource scheduling algorithm which is utilization and power capacity aware best fit decreasing(UPCBFD) including theoretical analysis. In Section 4, two sets of experiments are implied. Based on the results, we analyze experimental data to verify the correctness of this algorithm. Conclusion and discuss of possible research directions for future research is given in Section 5.

2. **Related work.** There are many kinds of ideas proposed about cloud resource scheduling problem in cloud computing field. Different researchers pay attention to different problems. Some of them focus on green cloud which aims at reducing energy consumption[8], others focus on efficiency of resource scheduling algorithm[9]. All of these algorithms can be split into two small algorithms which are virtual machine selection algorithm and virtual machine allocation algorithm.

2.1. **Virtual Machine Allocation Algorithms.** In order to detect overloaded hosts to initiate migration of VMs from these host, a heuristic idea of setting an upper and lower utilization threshold was firstly proposed by Beloglazov and Buyya[10]. However due to unpredictable workload, a fixed value of utilization threshold is not appropriate. Therefore, authors proposed four overload detection techniques based on auto adjustment utilization threshold in their later work[7], which are Median Absolute Deviation (MAD), Interquartile Range (IQR), Local Regression (LR) and Robust Local Regression (LRR). These algorithms are all implemented on CloudSim[11]. Some researchers explore to use swarm intelligence optimization algorithms. In Edouard Outins paper[5], a virtual machine algorithm based on genetic algorithm is used. It cannot be an alternative for above algorithms proposed by Beloglazov[7]. T.P. Shabeera proposed a virtual machine allocation algorithm based on ACO metaheuristic[12]. In Zhen Xiaos work[13], a Fast Up Slow Down algortihm(FUSD) is proposed. The algorithm behaves well at above four indicators. But it is too complex to combine with other complex virtual machine selection algorithms such as Bayes algorithm.

2.2. **Virtual Machine Selection Algorithms.** After finding out an overloaded host, the next step is to select the particular virtual machines to migrate from overload hosts to the other. Beloglazov and Buyya[7] have proposed some simply virtual machine selection policies such as Minimum Migration Time (MMT), Random Choice Policy (RC), Minimum Utilization policy(MU) and Maximum Correlation policy (MC). These policies are implemented on CloudSim. In Sahar Sohrabis paper[14], a Maximum Utilization policy(MaxUtil) relative to above Minimum Utilization policy(MU) is proposed. In another paper of Sahar Sohrabi[6], Bayes theory to virtual machine selection problem is introduced. Just as the experiment results attached at the tail of her paper, the algorithm does better than the four simply policies proposed above[7]. This work aims at combining virtual machine placement algorithm to cut down SLA violation.

2.3. **Virtual Machine Placement Algorithms.** The virtual machine placement problem can be modeled as bin-packing problem with restricted conditions–bin sizes and prices. The physical nodes can be modeled as the bin which virtual machines are allocated to. So virtual machines could be represented as the items, bin size can be seen as available CPU capacities and price can be seen as the power consumption by the nodes. Some popular solutions for bin packing problem have been proposed for a long time, including First Fit (FF), First Fit Decreasing (FFD), Best Fit Decreasing (BFD) and Worst Fit Decreasing (WFD). Among many solutions of virtual machine placement problem, there was a modification of popular Best Fit Decreasing(BFD) algorithm, proposed by Beloglazov and Buyya [7], which was named PABFD (power aware best fit decreasing). This algorithm firstly sorts the VMs according to their CPU utilization in decreasing order and then for each VM, it checks all the hosts and finds the suitable host where the increase of power consumption is minimum. At last, it allocates the VM to that host. Thiago Kenji Okada[15] proposed a Global Power Aware Best Fit Decreasing algorithm(GPABFD) which based on above PABFD. This algorithm is a modification of the PABFD algorithm. Instead of considering the power consumption increase after the allocation of a new VM, this algorithm calculates the power consumption of the whole data center after simulating an allocation of VM in the current host. However, GPABFD is not as efficient and fast as the FFD. Lei Shi [16] proposed some packing algorithms for virtual machine placement problem. We implement two algorithms that performed quite well based on their paper, which called the CPUUtilization and the PowerCapacity algorithms. Our algorithm improves their algorithm and the performance is superior to theirs. Mohammadhossein Malekloo proposed a Multi-objective ACO Virtual Machine Placement algorithm(MACO) [17]. Although the MACO algorithm has many advantages such as low energy consumption, low resource wastage, etc. Similarly, Mohammadhossein Malekloo, Fahimeh Farahnakian also used ant colony system to consolidate virtual machines [18]. As we all know, ant colony optimization algorithm has defects such as slow convergence, easy to converge to a local solution and time-consuming. As a result, the algorithms which use ACO all inherit the disadvantages above.

3. **Proposed Methodology.**

3.1. **The Introduction of CloudSim Environment.** CloudSim is a holistic framework for modeling, simulating and experimenting with cloud computing environment. It [11] was built in CLOUDS (Cloud Computing and Distributed System) Laboratory by the University of Melbourne, Australia. Currently, it supports modeling and simulation of cloud computing environments consisting of both single and inter-networked clouds (federation of clouds), and it also supports energy-efficient management of datacenter resources. CloudSim [19] can be used to model datacenters, host, VMs, service brokers, and scheduling and allocation policies of a large scaled cloud platform. Cloudsim supports VM provisioning at two levels – the host level and the VM level. The workloads for evaluation are the ones for CoMon project, a monitoring infrastructure for PlanetLab [20] that are included in the CloudSim.

3.2. **Motivation.** Our algorithms idea is spurred by Sahar Sohrabis paper [6]. In the Conclusions and future work part of her paper, she pointed out that there is highest SLA violation in her proposed algorithm BMH. We implement the BMH algorithm stated in the paper. And we attempt to do some improvement at other places to cut down SLA violation such as virtual machine allocation section or virtual machine placement section. There is Fast Up Slow Down algorithm proposed by Zhen Xiao [13] doing well. We implemented this algorithm and found that it was very complex while the result was not the best. Hence, we want to find out some algorithm which are simple and effective. We focus on virtual machine placement section. After improving Lei Shis algorithm [16], we propose a new algorithm which behaves better than Lei Shis algorithm and the FUSD algorithm.

3.3. **The Bin Packing Problem.** Definition: Given a list of objects and their weights, and a collection of bins of fixed size, find the smallest number of bins so that all the objects are assigned to a bin.

Bin packing is an NP-hard combinatorial optimization problem which based on the partition problem. Finding out the optimal solution is known to be exponentially difficult. One kind of solution methodology is to formulate the problem as an integer programming problem [16] which is showed at formula 1. However, heuristic methods have been developed to handle larger problems. Many popular algorithms were proposed. The difference between them is the different order of items or bins before placing operation.

$$min \sum_{i=1}^{I} H_i + \alpha \sum_{j=1}^{J} \sum_{k=1}^{|v_j|} m(v_j^k) \quad (1)$$

In formula 1, H represents host, i denote the max amount available host. The first item is the number of hosts satisfying the VM requests. H_i is equal to 1, if there is at least one VM running on H_i, otherwise 0. The second item represents the number of migrations introduced by the migrations of the VMs allocated during the lifecycle of the placement. The parameter $\alpha > 0$ is used to tune the impact of each item in the above objective function.

$$m(v_j^k) = \sum_{i=1, i \neq h(v_j^k)}^{I} x_i(v_j^k) \quad (2)$$

In formula 2, for a VM v_j^k , if it is migrated then $m(v_j^k)$ is 1, otherwise 0. We aim at minimizing the total number of migrations. Therefore, it is also integrated into the objective, mainly to act as a penalty for placement plan updates which cause widespread migration of VM instances between hosts. The resource capacity constraint is specified at second item of formula 1, which ensures that physical machine resources are not over-allocated. A bin packing algorithm needs to place a new VM at each time, subject to the new VMs requirements. Before introducing our algorithm, we prefer to illustrate some popular bin packing algorithms such as FFD.

The most common sorting strategy in FFD is to sort the items according to their resource requirements. This work considers two dimensions of resources (CPU and memory) and VMs is sorted based on its Rq(v) computed by using formula 3. CPU_i denotes the number of CPUs required by VM_i and Mem_i denotes the amount of memory required by VM_i.

$$Rq(i) = \sqrt{(CPU_i)^2 + Mem_i)^2)} \quad (3)$$

Similarly, the total capability $Cp(i)$ of a host j can be computed by the following formula 4.

$$Cp(j) = \sqrt{(CPU_j)^2 + Mem_j)^2)} \quad (4)$$

To measure the utilization of a host, the magnitude of every VM on a given PM will be summed to determine the utilization of the host. Let us have a host j, a set of VMs v_j, $i \in v_j$, which is the set of VMs placed on host j. The measurement of host utilization $Ut(j)$ is given in formula 5.

$$Ut(j) = \sqrt{(\sum_{i \in v_j} CPU_i)^2 + (\sum_{i \in v_j} Men_i)^2} \quad (5)$$

With respect to the common sorting strategies of FFD algorithm and based on Lei Shis algorithm [16], our proposed algorithm firstly sorts the available hosts according to their CPU utilization in the formula 5 in the reverse order.

On the other hand, we want to introduce Bayes methodology which applied in Sahar Sohrabis paper [6]. In this work, posterior probability of elements among a matrix is given. The element which has maximum posterior probability is selected. Then, the appropriate virtual machine is selected to migration based on this. The posterior probability can be computed in the following formula (6).

$$p(x_1|x_2) = \frac{P(x_2|x_1)P(x_1)}{P(x_2)} \tag{6}$$

A mass of experiments to verify her algorithm superiority. The other indicators all performed better than the contrast algorithms. But SLA violation is the highest among all experiments algorithms. Actually, the SLA violation metrics can be computed by using the following formula mentioned in Belogazovs paper [7].

$$SLATAH = \frac{1}{N}\sum_{i=0}^{N}\frac{T_{s_i}}{T_{a_i}} \tag{7}$$

$$PDM = \frac{1}{M}\sum_{j=0}^{M}\frac{C_{d_j}}{C_{r_j}} \tag{8}$$

$$SLAV = SLATAH * PDM \tag{9}$$

4. Experimental Results and Analysis.

4.1. The First Experiment. Two algorithms are used to make comparison: one is the PercentageUtil(PU) and the other is the AbsoluteCapacity(AC) proposed in Lei Shis paper [16]. As a reference, we also select the vm placement algorithm build-in in CloudSim which is BFD. To ensure the effectiveness of the experiment, the virtual machine selection algorithm we selected is MaximumCorrelation(MC). The number of Host is 800. The utilization threshold is 0.7. The data is a mean value of all result. Among all four indicators, the SLA Violation and Energy Consumption are the most important and significant. We choose these two indicators to form the following figures.

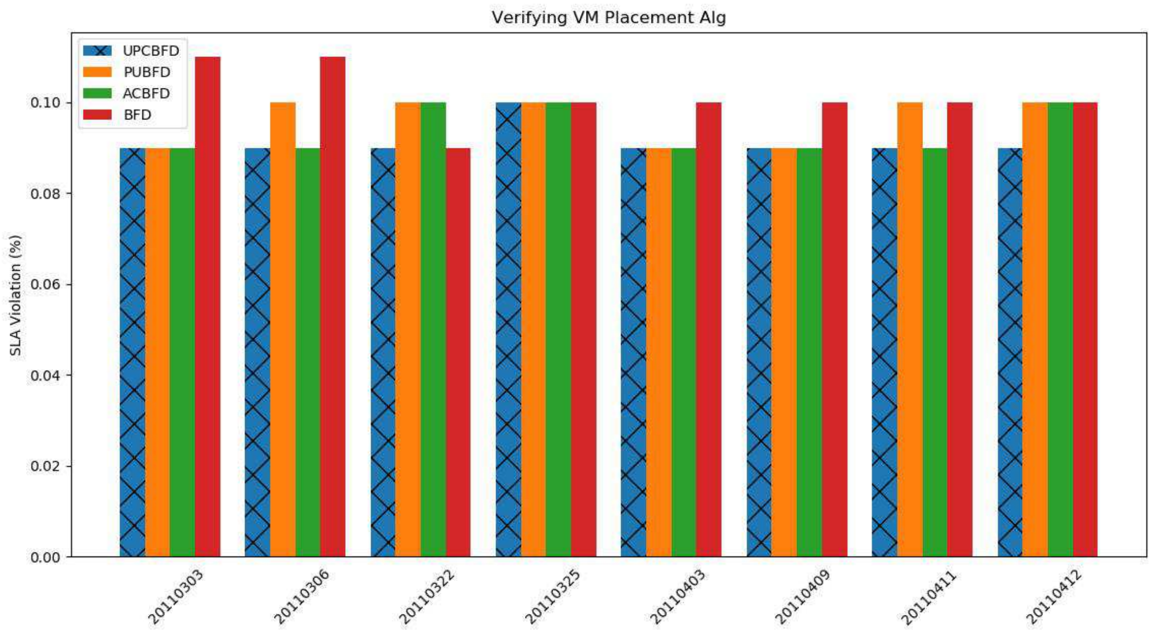

FIGURE 1. SLA Violation (%)

From the result, we can see that our algorithm performs better at SLA Violation and Energy Consumption.

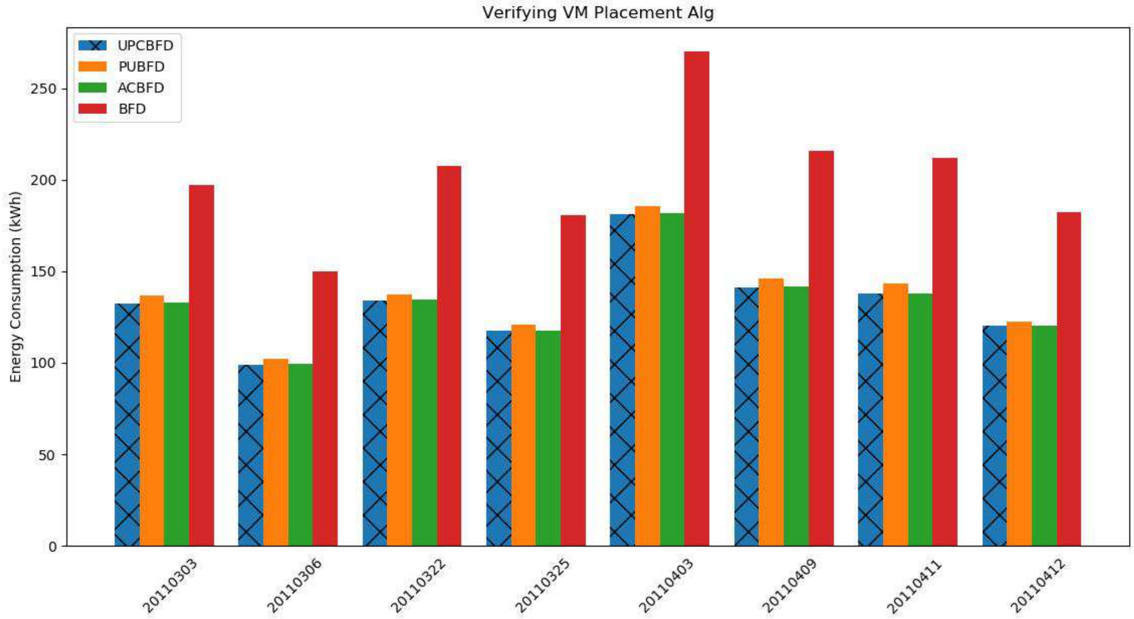

FIGURE 2. Energy Consumption (kWh)

4.2. **The Second Experiment.** The BMH algorithm is selected as a compared algorithm. The thr algorithm and MaximumCorrelation(MC) algorithm build-in in CloudSim as a replacement of BMH. The number of Host is 800 and the utilization threshold is 0.7. The data is a mean value of all experiment result. Among all four factors we know that the SLA Violation and Energy Consumption are the most important. The following Fig.3 and Fig.4 show the performance.

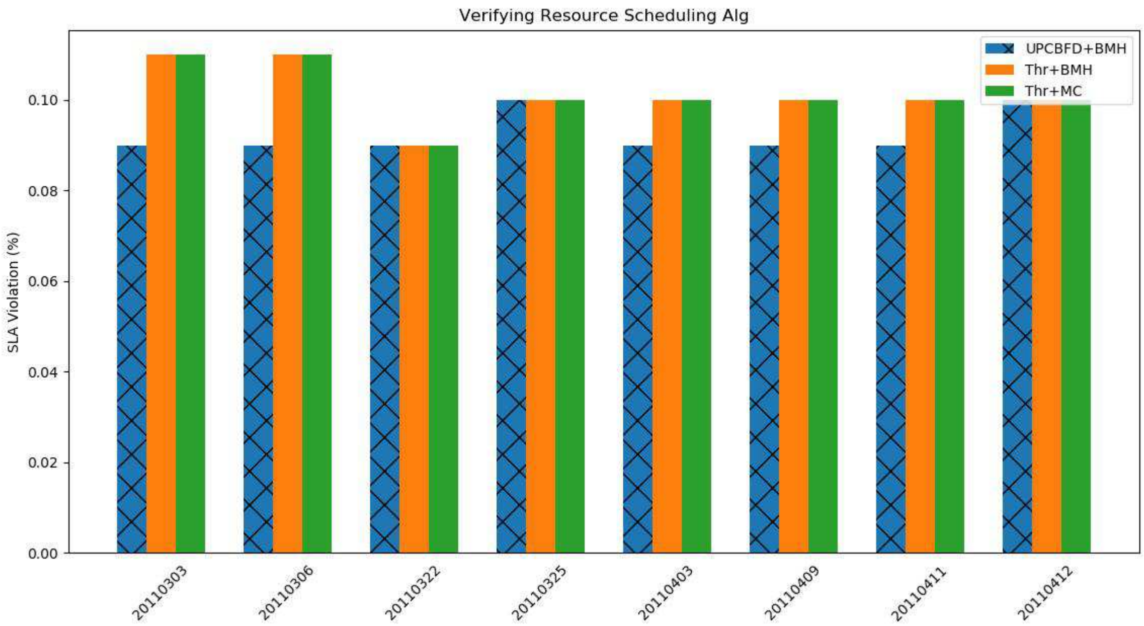

FIGURE 3. SLA Violation (%)

From the experiment result, we can see that our UPCBFD combined with BMH is best than the other two algorithms.

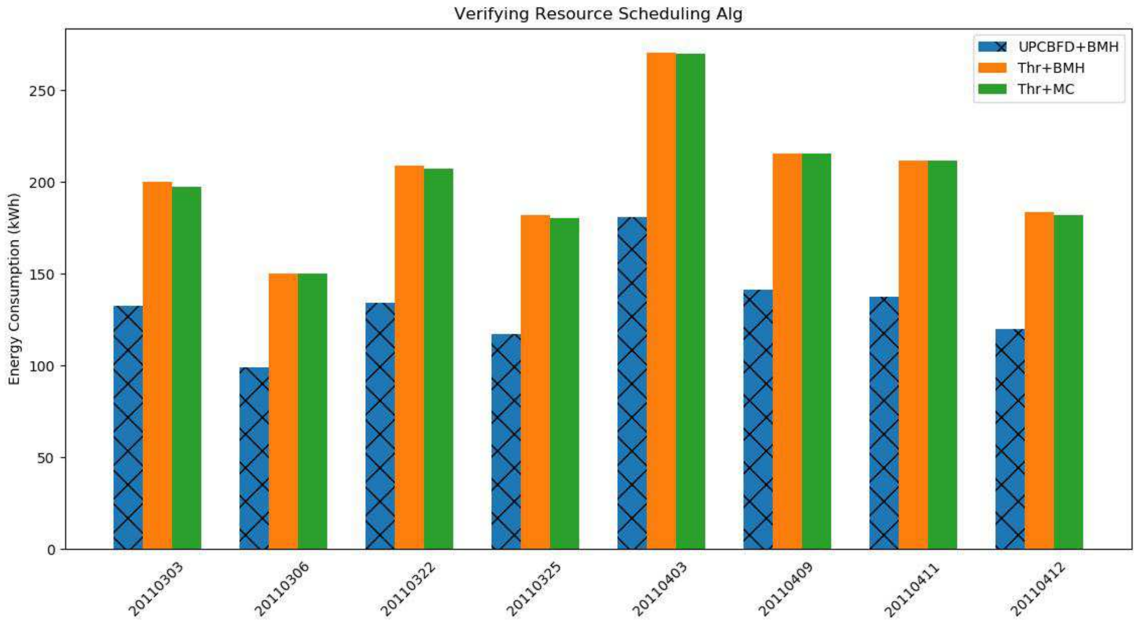

FIGURE 4. Energy Consumption (kWh)

5. **Conclusion.** This paper proposed UPCBFD virtual machine algorithm combined with BMH algorithm can substantially reduce energy consumption and cut down SLA Violation. However, we think that there is some space of optimization furthermore. As a follow-up of Sahar Sohrabis paper, we fundamentally have resolved the defect in her paper, namely the highest SLA Violation. For the future work, we will need to find out other better algorithms to reduce SLA Violation. Last but not least, physical machine failures should be considered for the dynamic VM placement.

Acknowledgement. The work was supported in part by Shenzhen Technical Project under Grant number: JCYJ20170307151750788 and in part by Shenzhen Technical Project under Grant number: KQJSCX20170327161755 and KQJSCX20170726104033.

REFERENCES

[1] Wikipedia Cloud computing[EB/OL]. [2008-12-20].

[2] K. Li, G. Xu, G. Zhao , et al. Cloud task scheduling based on load balancing ant colony optimization*Chinagrid Conference (ChinaGrid), 2011 Sixth Annual. IEEE,* 2011, pp. 3-9.

[3] Amazon EC2 Instance Types. [Online].

[4] R. Brown et al., Report to congress on server and data center energy efficiency: Public law 109-431, Lawrence Berkeley National Laboratory, 2008.

[5] E. Outin , J. E. Dartois , O. Barais , et al. Seeking for the optimal energy modelisation accuracy to allow efficient datacenter optimizations, *Pro. on Cluster, Cloud and Grid Computing (CCGrid), 2016 16th IEEE/ACM International Symposium on. IEEE*, 2016, pp. 535-539.

[6] [6] S. Sohrabi , A. Tang, Moser I, et al. Adaptive virtual machine migration mechanism for energy efficiency, *Pro. on Proceedings of the 5th International Workshop on Green and Sustainable Software. ACM,* 2016, pp. 8-14.

[7] A. Beloglazov , Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurrency Computat. Pract Exper 24, pp. 1397C1420. doi:10.1002/cpe.1867

[8] S. S.Masoumzadeh, H. Hlavacs An intelligent and adaptive threshold-based schema for energy and performance efficient dynamic VM consolidation, *Pro. onRevised Selected Papers of the COST IC0804 European Conference on Energy Efficiency in Large Scale Distributed Systems-Volume 8046. Springer-Verlag New York*, Inc., 2013, pp. 85-97.

[9] C. Xu , L. Song , Z. Han, et al. Efficiency resource allocation for device-to-device underlay communication systems: A reverse iterative combinatorial auction based approach, *Journal of IEEE Journal on Selected Areas in Communications*, 2013, vol. 31, no. 9, , pp. 348-358.

[10] A. Beloglazov , Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, *Journal of Future generation computer systems*, 2012, vol. 28, no. 5, pp. 755-768.

[11] R. N. Calheiros, R. Ranjan , De Rose C A F, et al. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services, *Journal of arXiv preprint arXiv:* 0903.2525, 2009.

[12] T. P. Shabeera, S. D. M. Kumar , Salam S M, et al. Optimizing VM allocation and data placement for data-intensive applications in cloud using ACO metaheuristic algorithm, *Journal of Engineering Science and Technology, an International Journal*, 2017, vol. 20, no. 2, pp. 616-628.

[13] Xiao Z, Song W, Chen Q. Dynamic resource allocation using virtual machines for cloud computing environment, *Journal of IEEE transactions on parallel and distributed systems*, 2013, vol. 24, no. 6, pp. 1107-1117.

[14] S. Sohrabi , I..Moser The effects of hotspot detection and virtual machine migration policies on energy consumption and service levels in the cloud, *Journal of Procedia Computer Science*, 2015, vol. 51, pp. 2794-2798.

[15] T. K. Okada, A. D. L. F. Vigliotti , Batista D M, et al. Consolidation of VMs to improve energy efficiency in cloud computing environments, *Pro. on Computer Networks and Distributed Systems (SBRC), 2015 XXXIII Brazilian Symposium on. IEEE*, 2015, pp. 150-158.

[16] L. Shi , J. Furlong , R. Wang, Empirical evaluation of vector bin packing algorithms for energy efficient data centers, *Pro. on Computers and Communications (ISCC), 2013 IEEE Symposium on. IEEE*, 2013, pp. 000009-000015.

[17] M. Malekloo , N. Kara, Multi-objective ACO virtual machine placement in cloud computing environments, *Pro. on Globecom Workshops (GC Wkshps)*, 2014. IEEE, 2014, pp. 112-116.

[18] F. Farahnakian , A. Ashraf ,T. Pahikkala , et al. Using ant colony system to consolidate VMs for green cloud computing, *Journal of IEEE Transactions on Services Computing*, 2015, 8, no. 2, pp. 187-198.

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

[20] K. S. Park and V. S. Pai, CoMon: a mostly-scalable monitoring system for planetlab. *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65-74, 2006.