



# A Clustered Virtual Machine Allocation Strategy Based on an $N$ -Threshold Sleep-Mode in a Cloud Environment

Xiuchen Qie<sup>1</sup>, Shunfu Jin<sup>1(✉)</sup>, and Wuyi Yue<sup>2</sup>

<sup>1</sup> School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, People's Republic of China  
qiexiuchen@126.com, jsf@ysu.edu.cn

<sup>2</sup> Department of Intelligence and Informatics, Konan University, Kobe 658-8501, Japan  
yue@konan-u.ac.jp

**Abstract.** In an effort to improve the energy efficiency of cloud data centers, in this paper, we propose a clustered Virtual Machine (VM) allocation strategy based on an  $N$ -threshold sleep-mode in which all the VMs in a cloud data center are clustered into two modules. The VMs in Module I are always awake, whereas the VMs in Module II will go to sleep under a light traffic load. When the number of waiting requests reaches or exceeds the threshold  $N$ , sleeping VMs will resume processing requests independently after their corresponding sleep timers expire. Accordingly, we establish an  $N$ -policy partially asynchronous multiple vacations queueing model, and derive the energy saving rate of the system. Numerical results are provided to show the efficiency of the proposed strategy in reducing energy consumption.

**Keywords:** Cloud data center · Clustered VM allocation  
 $N$ -threshold · Sleep-mode · Energy saving rate

## 1 Introduction

According to the current “Cisco Global Cloud Index”, more than four fifths of the workload in data centers will be handled in cloud data centers by 2019 [1]. As a result, energy efficiency is becoming increasingly important in a cloud environment [2].

The use of a sleep mode improves energy efficiency in cloud data centers [3]. In [4], Duan *et al.* proposed a dynamic idle interval prediction scheme that could estimate the future idle interval length of a CPU and thereby choose the most cost-effective sleep state to minimize the energy consumption during runtime. In [5], Chou *et al.* proposed a fine-grain power management scheme for data center workloads. This scheme dynamically postponed the processing of some requests, created longer idle periods and promoted the use of a deeper sleep mode. In

[6], Luo *et al.* proposed a dynamic adaptive scheduling algorithm based on flow preemption and power-aware routing. This algorithm saved energy by decreasing the ratio of low utilization devices and putting more devices into sleep mode. In the literature mentioned above, we note that a sleep mode was only applied to a Physical Machine (PM) rather than a Virtual Machine (VM).

In this paper, taking advantage of virtualization technology in cloud computing, we propose a clustered VM allocation strategy based on an  $N$ -threshold sleep-mode, and build an  $N$ -policy partially asynchronous multiple vacations queueing model. Then, we evaluate the system performance in terms of the energy saving rate of the system, both mathematically and numerically.

The rest of this paper is organized as follows. In Sect. 2, we propose a clustered VM allocation strategy based on an  $N$ -threshold sleep-mode in a cloud environment and build a system model accordingly. In Sect. 3, we analyze the system model using the method of matrix geometric solution. In Sect. 4, we derive the energy saving rate of the system. With numerical experiments, we investigate the system performance with the proposed strategy in Sect. 5. Finally, Sect. 6 concludes the whole paper.

## 2 VM Allocation Strategy and System Model

In this section, to improve the energy efficiency in a cloud environment, we first propose a clustered VM allocation strategy based on an  $N$ -threshold sleep-mode. Then, we develop an  $N$ -policy partially asynchronous multiple vacations queueing model to analyze the system with the proposed strategy.

### 2.1 VM Allocation Strategy

We note that additional energy will be consumed when the VM frequently switches from the sleep state to the awake state, and the system performance will be degraded when all the VMs are put in an imposed sleep-mode. To get around this problem, a clustered VM allocation strategy based on an  $N$ -threshold sleep-mode is proposed.

All the VMs in a cloud data center are clustered into two modules, namely, Module I and Module II. The VMs in Module I stay awake and operate on a higher speed. The VMs in Module II will go to sleep independently when there are no requests in the system buffer. At the end of a sleep period, if the requests gathered in the system buffer reaches or exceeds a certain value, namely threshold  $N$ , the corresponding VM in Module II will wake up independently and operate on a lower speed. Otherwise, the VM in Module II will restart a sleep timer and begin another sleep period.

In the proposed strategy, all the VMs are dominated by a control server, in which several sleep timers, a request counter, and a VM scheduler are deployed. Each sleep timer determines the time length of a sleep period. The request counter records the number of the requests waiting in the system buffer. Based

on the sleep timers and the request counter, the VM scheduler adjusts the system state.

In Fig. 1, we demonstrate the workflow of VMs with the clustered VM allocation strategy based on an  $N$ -threshold sleep-mode.

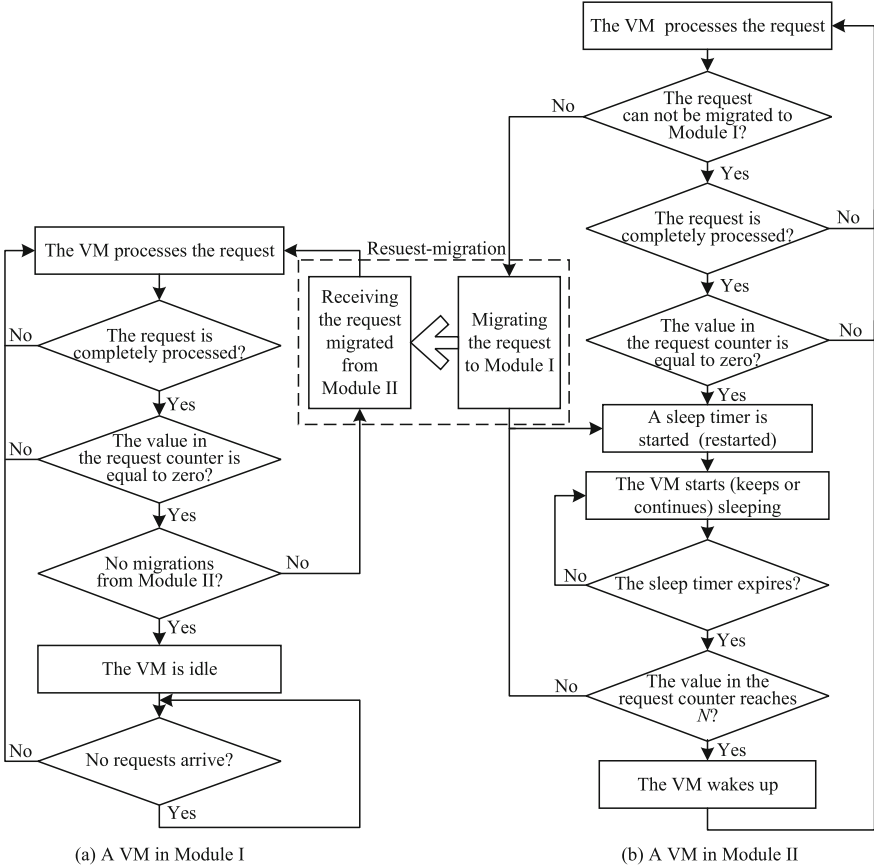


Fig. 1. The workflow diagram of a VM with the proposed strategy.

## 2.2 System Model

Regarding a request as a customer, a VM as an independent server, and a sleep period as a vacation, we model the proposed strategy as an  $N$ -policy partially asynchronous multiple vacations queueing model.

In this system model, the numbers of the VMs in Module I and Module II are denoted as  $c$  and  $d$ , respectively. The arrival intervals of requests are assumed to follow an exponential distribution with parameter  $\lambda$  ( $\lambda > 0$ ). The service times of requests processed in Module I and in Module II are assumed



$B_k$  ( $k = 1, 2, \dots, c$ ),  $\mathbf{B}_k$  ( $k = c + 1, c + 2, \dots, c + d$ ) and  $\mathbf{B}$  ( $k = c + d + 1, c + d + 2, \dots$ ) are the one-step state transition rate sub-matrices for the system level  $k$  decreasing by one.

$$B_k = k\mu_1, \quad k = 1, 2, \dots, c,$$

$$\mathbf{B}_k = \begin{pmatrix} c\mu_1 & & & & \\ & c\mu_1 + \mu_2 & & & \\ & & \ddots & & \\ & & & c\mu_1 + (x - 1)\mu_2 & \\ & & & & c\mu_1 + x\mu_2 \end{pmatrix}, \quad k = c + x, \quad x = 1, 2, \dots, d,$$

$$\mathbf{B} = \text{diag}(c\mu_1, c\mu_1 + \mu_2, \dots, c\mu_1 + d\mu_2), \quad k = c + x, \quad x = d + 1, d + 2, \dots$$

$A_k$  ( $k = 1, 2, \dots, c$ ),  $\mathbf{A}_k$  ( $k = c + 1, c + 2, \dots, c + d + N - 1$ ) and  $\mathbf{A}$  ( $k = c + d + N, c + d + N + 1, \dots$ ) are the one step state transition rate sub-matrices for the system level  $k$  remaining fixed. For convenience of presentation, we introduce  $h_y$  ( $h_y = \lambda + c\mu_1 + y\mu_2, 0 \leq y \leq d$ ) to simplify the sub-matrices  $\mathbf{A}_k$  and  $\mathbf{A}$ .

$$A_k = -(\lambda + k\mu_1), \quad k = 0, 1, \dots, c,$$

$$\mathbf{A}_k = \text{diag}(-h_0, -h_1, \dots, -h_x), \quad k = c + x, \quad x = 1, 2, \dots, \min\{N, d\} - 1.$$

For the case of  $N > d$ ,

$$\mathbf{A}_k = \text{diag}(-h_0, -h_1, \dots, -h_d), \quad k = c + x, \quad x = d, d + 1, \dots, N - 1.$$

For the case of  $N \leq d$ ,

$$\mathbf{A}_k = \begin{pmatrix} -h_0 - d\theta & & d\theta & & & \\ & \ddots & & \ddots & & \\ & & -h_{x-N} - (d - (x - N))\theta & (d - (x - N))\theta & & \\ & & & -h_{x-N} & 0 & \\ & & & & \ddots & \\ & & & & & -h_{x-1} & 0 \\ & & & & & & -h_x \end{pmatrix},$$

$$k = c + x, \quad x = N, N + 1, \dots, d - 1.$$



Applying the Gauss-Seidel method [7], we can obtain  $\pi_i$  ( $i = 0, 1, \dots, c + d + N$ ). Based on the matrix geometric solution form  $\pi_i = \pi_{c+d+N} \mathbf{R}^{i-(c+d+N)}$ ,  $i \geq c + d + N$ , we can obtain  $\pi_i$  ( $i = c + d + N + 1, c + d + N + 2, \dots$ ).

## 4 Performance Measures

We define the energy saving rate of the system as the energy conservation per unit time. Energy saving rate of the system is a measure to compare the total energy consumption in our proposed strategy and that in the conventional strategy. Based on the steady-state probability distribution of the system model given in Sect. 3, the energy saving rate  $E$  of the system with our proposed strategy is given as follows:

$$E = E_1 - (E_2 + E_3) \quad (2)$$

where  $E_1$  is the energy saving rate during the sleep period,  $E_2$  and  $E_3$  are the additional energy consumption rates caused by a request-migration and by a listening at the boundary of the sleep period.

$$E_1 = (\omega - \omega_s) \sum_{i=0}^{\infty} \sum_{j=0}^d (d-j) \pi_{ij},$$

$$E_2 = \omega_m \sum_{i=c+1}^{c+d} \sum_{j=1}^d c \mu_1 \pi_{ij},$$

$$E_3 = \omega_l \sum_{i=0}^{\infty} \sum_{j=0}^d \theta (d-j) \pi_{ij}$$

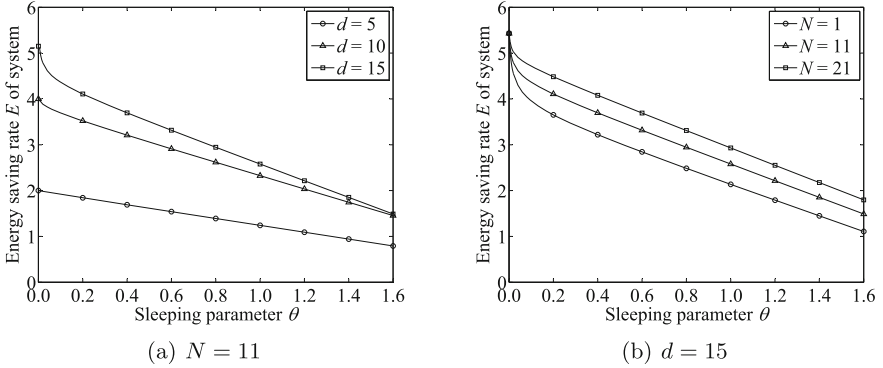
where  $\omega$  ( $\omega > 0$ ) is the energy consumption per unit time for a busy VM in Module II.  $\omega_s$  ( $\omega_s > 0$ ) is the energy consumption per unit time for a sleeping VM in Module II.  $\omega_m$  ( $\omega_m > 0$ ) is the energy consumption for each request-migration.  $\omega_l$  ( $\omega_l > 0$ ) is the energy consumption for each listening.

## 5 Numerical Experiments

In order to quantify the impact of the sleeping parameter on the energy saving rate of the system for the different number of the VMs in Module II and the different thresholds  $N$ , we provide numerical experiments. Referencing to [8], we set the experimental parameters as follows:  $c + d = 50$ ,  $\lambda = 7.00$  (requests/ms),  $\mu_1 = 0.20$  (requests/ms),  $\mu_2 = 0.10$  (requests/ms),  $\omega = 0.50$  mW,  $\omega_s = 0.10$  mW,  $\omega_m = 0.50$  mW and  $\omega_l = 0.15$  mW.

Figure 2 examines the influence of the sleeping parameter  $\theta$  on the energy saving rate  $E$  of the system for the different number  $d$  of the VMs in Module II and the different thresholds  $N$ .

From Fig. 2(a), we notice that when the sleeping parameter  $\theta$  and the threshold  $N$  are given, a larger number  $d$  of the VMs in Module II will lead to a higher



**Fig. 2.** Energy saving rate  $E$  of the system.

energy saving rate  $E$  of the system. As the number of the VMs in Module II increases, more VMs have the opportunity to take a sleep, so the energy saving rate of the system improves.

From Fig. 2(b), we notice that when the sleeping parameter  $\theta$  and the number  $d$  of the VMs in Module II are given, a bigger threshold  $N$  will lead to a higher energy saving rate  $E$  of the system. The higher the threshold  $N$  is, the later a VM in Module II will wake up from sleeping, so the VMs in Module II will stay in the sleep state for longer. This results in a higher energy saving rate of the system.

Combining Figs. 2(a) and 2(b), we also observe that for any number  $d$  of the VMs in Module II and any value for threshold  $N$ , the energy saving rate  $E$  of the system decreases as the sleeping parameter  $\theta$  increases. On the one hand, the larger the sleeping parameter is, the shorter the time length of a sleep period is, and the later a VM in Module II will wake up from sleeping, so less energy will be saved. On the other hand, the larger the sleeping parameter is, the more frequently the VM in Module II listens to the system buffer, so additional energy will be consumed. Therefore, the energy saving rate of the system will decrease.

## 6 Conclusions

In this paper, we proposed a clustered VM allocation strategy. Considering an  $N$ -threshold sleep-mode with the proposed strategy, we established an  $N$ -policy partially asynchronous multiple vacations queueing model. The queueing model quantified the effects of the number of VMs in Module II, the threshold  $N$  and the sleeping parameter on the energy saving rate of the system. In future research, we aim to extend our study to investigate the average latency of requests and to optimize the proposed strategy by making trade-offs between different performance measures.



**Acknowledgements.** This work was supported in part by National Natural Science Foundation (No. 61472342), Hebei Province Natural Science Foundation (No. F2017203141), China, and was supported in part by MEXT, Japan.

## References

1. Hintemann, R., Clausen, J.: Green Cloud? the current and future development of energy consumption by data centers, networks and end-user devices. In: 4th International Conference on ICT for Sustainability, pp. 109–115 (2016)
2. Jin, X., Zhang, F., Vasilakos, A., Liu, Z.: Green data centers: a survey, perspectives, and future directions (2016). <https://arxiv.org/pdf/1608.00687v1.pdf>
3. Fan, L., Gu, C., Qiao, L., Wu, W., Huang, H.: GreenSleep: a multi-sleep modes based scheduling of servers for cloud data center. In: International Conference on Big Data Computing and Communications, pp. 368–375 (2017)
4. Duan, L., Zhan, D., Hohnerlein, J.: Optimizing cloud data center energy efficiency via dynamic prediction of CPU idle intervals. In: 8th IEEE International Conference on Cloud Computing, pp. 985–988 (2015)
5. Chou, C., Wong, D., Bhuyan, L.: DynSleep: fine-grained power management for a latency-critical data center application. In: International Symposium on Low Power Electronics and Design, pp. 212–217 (2016)
6. Luo, J., Zhang, S., Yin, L., Guo, Y.: Dynamic flow scheduling for power optimization of data center networks. In: 5th International Conference on Advanced Cloud and Big Data, pp. 57–62 (2017)
7. Jiang, M., Hu, J., Zhao, R., Wei, X., Nie, Z.: Hybrid IE-DDM-MLFMA with Gauss-Seidel iterative technique for scattering from conducting body of translation. *Appl. Comput. Electromagn. Soc. J.* **30**(2), 148–156 (2015)
8. Jin, S., Ma, X., Yue, W.: Energy-saving strategy for green cognitive radio networks with an LTE-advanced structure. *J. Commun. Netw.* **18**(4), 610–618 (2016)