International Journal of Recent Engineering Science (IJRES),
ISSN: 2349-7157, *volume 2 Issue 5 September to October 2015*

Design and Evaluation of a Method for Enhancing Robustness of MCMC-Based Autonomous Decentralized Mechanism for VM Assignment Problem

Masaya Yokota[1] , Yusuke Sakumoto[1] , and Masaki Aida[1]
[1]*Graduate School of System Design, Tokyo Metropolitan University*
6-6 Asahigaoka, Hino-shi, Tokyo 191-0065 Japan

ABSTRACT *: Autonomous Decentralized Mechanism (ADM) is actively discussed to realize a scalable control scheme for large-scale is wide-area systems. The previous work has proposed the Markov Chain Monte Carlo (MCMC)-based ADM for indirectly controlling the entire system, but has not fully discussed the impact of environmental fluctuation on its robustness. In this paper, we propose a method to enhance the robustness of the MCMC-based ADM against severe environmental fluctuations. In the proposed method, each node adjusts the control parameter of the MCMC-based ADM to absorb environmental fluctuations. In addition, we apply the proposed method to virtual machine (VM) assignment problem in data centers. Through simulation experiment, we clarify the effectiveness of the proposed method for severe environmental fluctuation.*

Keywords *-Autonomous Decentralized Mechanism, Large-Scale and Wide-Area System, Markov Chain Monte Carlo, Robustness, Environmental Fluctuation, Virtual Machine Placement Problem*

1. Introduction

In recent years, progress of Information and Communications Technology (ICT) systems is remarkable [1]. Many ICT systems are supported by large-scale and wide-area data centers to accommodate numerous requests. For instance, Google has distributed a large number of servers composed of its data center around the world [2], and many ICT systems provide service by using Google data center. It is expected such a data center is becoming larger and larger in order to improve computational capability and system reliability. Hence, a scalable control scheme should be realized to handle sustainably large-scale and wide-area systems like data centers.

In general, system control architecture is roughly categorized as Centralized Mechanism (CM) or Autonomous Decentralized Mechanism (ADM). Consider the situation that a system is composed of numerous nodes distributed in a wide area. In a CM, there is a management node and it has to gather state information from the entire system, and controls intensively the state of all nodes in the system. Hence, CM requires a large amount of time to gather state information in a large-scale and wide-area system, and cannot handle such a system against dynamic environment fluctuation. In an ADM, each node autonomously controls its own state on the basis of local information that can be gathered directly by the node. The local information is, in other words, the state information of neighboring nodes. Since each node uses only local information, ADMs can quickly react for an environmental fluctuation. Hence, ADM is being actively discussed in order to build a scalable control scheme for large-scale and wide-area systems (see, e.g., [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]). In ADM, an autonomous node action influences a state of the system. Therefore such an action in ADMs should be designed to be able to tie to the control of the entire system [3].

In [9], the authors have proposed an ADM based on Markov Chain Monte Carlo (MCMC) [13, 14], which is a method to generate a Markov process following a desired probability distribution of a statistical-mechanical variable (e.g., energy). In [9], on the basis of MCMC, the authors designed an autonomous node action to control for the probability distribution of system performance variable that is an amount to quantify of a system state. In addition, they applied the method for solving the virtual machine (VM) assignment problem. In [10], the authors have proposed an advanced function of the MCMC-based ADM as the solution for the VM assignment problem against simple environmental fluctuations, but have not fully shown for its robustness against several environmental fluctuations. There are various kinds of environmental fluctuations for actual systems, so it is crucial to discuss robustness of the MCMC-based ADM and improve its robustness if necessary.

In this paper, we propose a method to enhance its robustness against severe environmental fluctuations. In the proposed method, each node autonomously adjusts its control parameter of the MCMC-based ADM. Then, similarly to [9], we apply the proposed method to the VM assignment problem in data center network (DCN). Through simulation experiment, we clarify the effectiveness of the proposed method for severe environmental fluctuation.

The short version of this paper is presented in [12]. In [12], due to space limitation, we briefly introduced key idea of the proposed method, and only confirmed the effectiveness for fluctuation in traffic rate in DCNs. In addition to above, this paper gives theoretical explanation of the proposed method, and also confirm the effectiveness for fluctuation in number of nodes.

This paper is organized as follows. Section 2 introduces MCMC-based ADM [9]. In section 3, we discuss the impact of environmental fluctuation on the MCMC-based ADM, and design an its adjustment method to absorb environmental fluctuations. Section 4 details the experiments conducted for investigating the robustness of the proposed method in data center networks. Finally, in Section 5, we conclude this paper and discuss future work.

2. The MCMC-Based ADM

2.1. System Model for VM Assignment Problem

We first introduce a system model in this paper. Consider a distributed system in which n nodes interact with each other in order to process tasks. To process tasks, each node uses one of N distributed computational resources. The state of a node is determined by which resource is used in N resources. Fig.1 shows an example of the system model with n = 7 and N = 4. In this figure, we draw interacting nodes by links. Each node prefers to use the same resource or near resources with frequently interacting nodes, so nodes would be concentrated to a few resources. However, for efficient task processing, the load dispersion among resources should be realized. Since there is a trade-off between the node concentration and the load dispersion, the system should adequately adjust the balance of them subject to the trade-off.

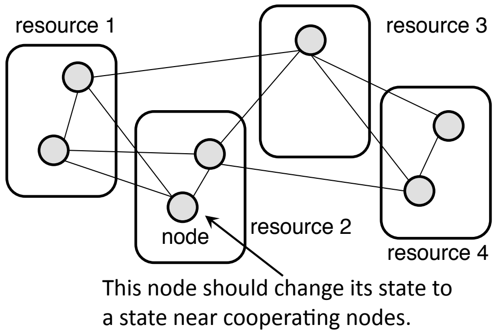

Fig. 1 An example of the system model with n = 7 and N = 4

In [9], the authors formulated VM assignment problem of nodes and resources with taking the trade-off between the node concentration and the load dispersion into consideration. Let x_i be node i's state that is the ID of the resource used by node i ($x_i \in \{1,..,N\}$). We define system state $\boldsymbol{X}$ by combination $(x_1,...,x_n)$ of all node states. System state $\boldsymbol{X}$ influences the node concentration and the load dispersion, so they assume that performance of the system depends on system state $\boldsymbol{X}$. Namely, system performance variable $M(\boldsymbol{X})$ is given by

$$M(\boldsymbol{X}) = \sum_{i=1}^{n} \sum_{j \in \chi_i} r_{ij} d(x_i, x_j), \qquad (1)$$

where χ_i is the set of nodes interacting with node i, r_{ij} is the interaction frequency between nodes i and j, and $d(k,l)$ is distance between resources k and l. Small $M(\boldsymbol{X})$ means strong node concentration but weak load dispersion. By controlling $M(\boldsymbol{X})$, the system adequately adjusts the balance of them.

2.2. Node Action

In the MCMC-based ADM [9], node i changes its state x_i to another state x_i' with state transition probability $T_i(x_i \rightarrow x_i')$, which is designed by MCMC. Specifically, $T_i(x_i \rightarrow x_i')$ is given by

$$T_i(x_i \rightarrow x_i') = \begin{cases} \frac{1}{|\phi_{x_i}|} \exp[-\alpha\lambda\Delta M_i(x_i \rightarrow x_i')] & \text{if } \Delta M_i(x_i \rightarrow x_i') < 0 \\ \frac{1}{|\phi_{x_i}|} \exp[-(1-\alpha)\lambda\Delta M_i(x_i \rightarrow x_i')] & \text{otherwise} \end{cases}, \qquad (2)$$

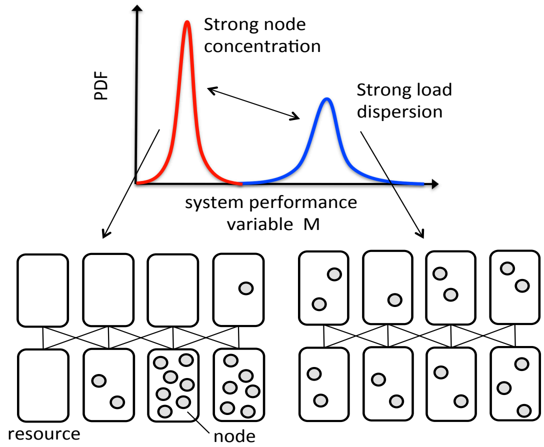

Fig. 2 A relation image between $P(M)$ and resource assignment problem of node2s

where ϕ_k is the set of states that are able to be transited from state k, λ is the control parameter of the MCMC-based ADM, and α is a positive constant ($0 \leq \alpha \leq 0.5$). $\Delta M_i(x_i \to x_i')$ is the amount of change of $M(\boldsymbol{X})$ with respect to the state transition of node i, and is given by

$$\Delta M_i(x_i \to x_i') = \sum_{j \in \chi_i} r_{ij} \left[d(x_i', x_j) - d(x_i, x_j) \right]. \quad (3)$$

If each node uses state transition probability $T_i(x_i \to x_i')$ by Eq. (2), system performance variable M follows the probability distribution

$$P(M) = \frac{G(M)\exp(-\lambda M)}{\sum_{Y \in \Omega_M} G(Y)\exp(-\lambda Y)}, \quad (4)$$

where $G(Y)$ is the number of system state if the system performance variable is equal to Y, and Ω_M is the set of all possible values of system performance variable M. According to Eq. (4), if $\lambda = 0$, $P(M)$ is proportional to $G(M)$. In this case, the MCMC-based ADM is equivalent to the control where each node selects its state at random. In addition, as control parameter λ increases, each node controls its state to lead to the emergence of smaller M. Therefore, the MCMC-based ADM can adjust $M(\boldsymbol{X})$ by changing λ. Fig.2 shows a relation image between $P(M)$ and resource assignment problem of nodes. In this figure, red line means strong node concentration but weak load dispersion and blue line means week node concentration but strong load dispersion. In addition, lower figure shows resources and node distribution. This probability distribution reflects system state, so we realize turning $P(M)$ by changing λ and it correspond with system state.

2.3. Global Property

In statistical mechanics, the probability distribution given by Eq. (4) is called the *Boltzmann distribution*, which is well understood. Hence, in [11], the authors clarified the global property of the MCMC-based ADM on the basis of statistical mechanics. According to the clarified global property, they found that the MCMC-based ADM yields a hierarchical structure that has node-level and system-level layers. At the system-level layer, there is the law between statistics (i.e., average and standard deviation) of M and statistics depending on the external environment of the system.

We explain the law on the system-level layer on the basis of the analysis in [9]. Let μ_λ and σ_λ be the averages and the standard deviation of $P(M)$ when using λ, respectively. Then, we define μ_G and σ_G by the average and standard deviation of $G(M)/|\Omega|$, respectively. They are given by

$$\mu_G = \frac{1}{|\Omega|} \sum_{\boldsymbol{X} \in \Omega} M(\boldsymbol{X}), \quad (5)$$

$$\sigma_G^2 = \frac{1}{|\Omega|} \sum_{\boldsymbol{X} \in \Omega} (M(\boldsymbol{X}) - \mu_G)^2, \quad (6)$$

where Ω is the state space of $\boldsymbol{X}$. Since r_{ij} and $d(x_i,x_j)$ in $M(\boldsymbol{X})$ are given by information of the external environment, μ_G and σ_G depend on the external environment. In [9], the authors derived the relation between these variables at the system-level layer. In the derivation process, they assumed that system performance variable M can be modeled as a continuous quantity. This assumption is valid for large-scale systems. By approximating Eq. (4) around its maximum point using the second-order Taylor series approximation, they showed that μ_λ and σ_λ are joined to μ_G and σ_G as follows

$$\mu_\lambda = \mu_G - \lambda \sigma_G^2, \quad (7)$$

$$\sigma_\lambda = \sigma_G, \quad (8)$$

at the system-level layer. According to the above equations, we find $\mu_0 = \mu_G$ and $\sigma_0 = \sigma_G$. Hence, from Eqs. (7) and (8), we can understand the relation between the external environment and the MCMC-based ADM.

3. Method for Enhancing Robustness of the MCMC-Based ADM

In this paper, we consider robustness, which is defined by the capability to retain system performance (i.e., node concentration) against environmental fluctuations. This robustness is crucial for realizing steady control of the entire system because high node concentration causes heavy load of a few resources, may lead to clash of the system.

We first discuss the impact of environmental fluctuations on the node concentration by the MCMC-based ADM. For instance, we consider a situation where r_{ij} increases due to environmental fluctuation. Note that we can also consider other situations (e.g., n changes) as same as this situation. The increase in r_{ij} produces the increase in $\Delta M_i(x_i \rightarrow x_i')$, and node i would change node behavior by the node action using Eq. (2). Namely, node i prefers to select the same resource or near resources with nodes with large r_{ij}. As the result, nodes are more concentrate to fewer resources. In this sense, such environmental fluctuations would change the node concentration by the MCMC-based ADM, so may become a big threat to the system.

To retain the node concentration, each node should absorb the impact of environmental fluctuations. According to the above discussion, node i should increase/decrease control parameter λ used in Eq. (2) so as to cancel the increasing/decreasing amount in $\Delta M_i(x_i \rightarrow x_i')$ of node i when an environmental fluctuation occurs. Namely, node i adjusts own control parameter λ_i by using

$$\lambda_i = \frac{K}{\sum_{j \in \chi_i} r_{ij}}, \tag{9}$$

where K is the control parameter of the proposed method, and is used to adjust the strength of the node concentration. Note that Eq. (9) can be calculated from only local information of node i, so node i can autonomously reconfigure λ_i.

After that, we discuss whether the reconfiguration by Eq. (9) can absorb the impact of environmental fluctuation, and the MCMC-based ADM with Eq. (9) can retain the strength of the node concentration. First, we derive the condition to retain the strength of the node concentration. Then, we confirm that the MCMC-based ADM with Eq. (9) satisfies the derived condition against environmental fluctuation.

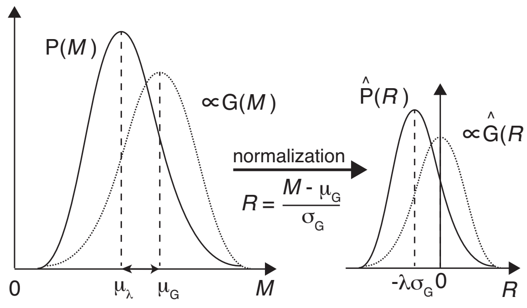

Fig. 3 Distribution $P(M)$ and $G(M)$ vs. Normalized distribution $\hat{P}(R)$ and $\hat{G}(R)$

Before we derive the condition to retain the strength of the node concentration, we quantify the strength. The strength of the node concentration is measured by the difference between states with randomly selected states ($\lambda = 0$) and states controlled by the MCMC-based ADM. We can define such a difference simply by $\mu_0 - \mu_\lambda = \mu_G - \mu_\lambda$. However, environmental fluctuation often changes $M(\boldsymbol{X})$ for each system state $\boldsymbol{X}$. As the result, μ_G would be increased/decreased, so it is difficult to measure the strength given by $\mu_G - \mu_\lambda$ under environmental fluctuation. Hence, we should give an invariant definition for such a difference under environmental fluctuation.

To obtain an invariant definition for the strength of the node concentration under environmental fluctuation, we define normalized variable R by

$$R := \frac{M - \mu_0}{\sigma_0} = \frac{M - \mu_G}{\sigma_G}. \tag{10}$$

The right sight of the above equation means the normalization of M. Let $\mu_\lambda^{(R)}$ and $\sigma_\lambda^{(R)}$ be the average and the standard deviation of R when using λ, respectively. From Eqs. (7), (8), and (10), $\mu_\lambda^{(R)}$ and $\sigma_\lambda^{(R)}$ are given by

$$\mu_\lambda^{(R)} = \frac{\mu_\lambda - \mu_G}{\sigma_G} = \frac{\mu_G + \lambda\sigma_G^2 - \mu_G}{\sigma_G} = \lambda\sigma_G, \quad (11)$$

$$\sigma_\lambda^{(R)} = \frac{\sigma_\lambda}{\sigma_G} = \frac{\sigma_G}{\sigma_G} = 1. \quad (12)$$

To retain the strength of the node concentration, we should keep $\mu_\lambda^{(R)}$ and $\sigma_\lambda^{(R)}$ constant. Therefore, as the condition to retain the strength of the node concentration, we obtain

$$\lambda\sigma_G = \kappa, \quad (13)$$

where $\kappa \geq 0$.

Then, we confirm whether Eq. (9) satisfied the derived condition (13). By substituting Eq. (6) into the left side of (13), we obtain

$$\begin{aligned}\lambda^2\sigma_G^2 &= \frac{\lambda^2}{|\Omega|}\sum_{\boldsymbol{X}\in\Omega}(M(\boldsymbol{X}) - \mu_0)^2 \\ &= \frac{1}{|\Omega|}\sum_{\boldsymbol{X}\in\Omega}(\lambda M(\boldsymbol{X}))^2 - (\lambda\mu_0)^2 \\ &= \frac{1}{N^2}\sum_{x_i=1}^{N}\sum_{x_j=1}^{N}\left(\sum_{i=1}^{n}\lambda\sum_{j\in\boldsymbol{X}_i} r_{ij}d(x_i,x_j)\right)^2 \\ &\quad -\frac{1}{N^4}\left(\sum_{x_i=1}^{N}\sum_{x_j=1}^{N}\sum_{i=1}^{n}\lambda\sum_{j\in\boldsymbol{X}_i} r_{ij}d(x_i,x_j)\right)^2. \quad (14)\end{aligned}$$

In the proposed method, each node has own control parameter λ_i, and λ_i is configured by Eq. (9). By substituting Eq. (9) into λ in Eq. (14), the first term in the right side of Eq. (14) is transformed to

$$\begin{aligned}&\frac{1}{N^2}\sum_{x_i=1}^{N}\sum_{x_j=1}^{N}\left(\sum_{i=1}^{n}\lambda_i\sum_{j\in\boldsymbol{X}_i} r_{ij}d(x_i,x_j)\right)^2 \\ &\approx \left(\overline{d}\sum_{i=1}^{n}\lambda_i\sum_{j\in\boldsymbol{X}_i} r_{ij}\right)^2 \\ &= \left(\overline{d}\sum_{i=1}^{n}K\right)^2 = \overline{d}^2 n^2 K^2, \quad (15)\end{aligned}$$

where d is the average of distances between resources. In the above derivation process, we used the approximation $d(k,l) \approx d$. In a similar way, the second term in the right side of Eq. (14) is transformed to

$$\begin{aligned}&\frac{1}{N^4}\left\{\sum_{x_i=1}^{N}\sum_{x_j=1}^{N}\sum_{i=1}^{n}\lambda_i\sum_{j\in\boldsymbol{X}_i} r_{ij}d(x_i,x_j)\right\}^2 \\ &\approx \left(\overline{d}\sum_{i=1}^{n}\lambda_i\sum_{j\in\boldsymbol{X}_i} r_{ij}\right)^2 \\ &= \left(\overline{d}\sum_{i=1}^{n}K\right)^2 = \overline{d}^2 n^2 K^2. \quad (16)\end{aligned}$$

According to Eqs. (15) and (16), we find that the right side of Eq. (14) is 0. Therefore, Eq. (14) roughly satisfies the derived condition Eq. (13). To obtain the conclusion, we use the approximation $d(k,l) \approx d$, so we should also confirm that the proposed method can retain the strength of the node concentration through experiment.

4. Simulation Experiment

4.1. Experiment Model

In this section, similarly to [9], we apply the proposed method to the VM assignment problem in DCNs (Data Center Networks). In this application, a node and a resource (a node state) correspond to a VM and a PM (Physical Machine) used by a node, respectively. Then, r_{ij} and $d(k,l)$ are the traffic rate between VMs i and j and the communication cost (i.e., the sum of communication costs in the shortest path) between PMs k and l, respectively.

In experiment, we use a network topology shown in Fig. 4. In this network topology, PMs are placed in layer 0, and network equipments (e.g., network switch and router) are placed on the other layers. For scalability reason, PMs are divided into four groups, and each state transition of a VM will be permitted only between PMs within the same or adjacency groups that are connected by a double-headed arrow.

As the metric for the node concentration in a resource assignment, we use PM load coefficient of variation CV[$\boldsymbol{\rho}$] ($\boldsymbol{\rho} = (\rho_1,\ldots,\rho_N)$) of PM loads, which is defined by

$$\mathrm{CV}[\boldsymbol{\rho}] = \frac{1}{\mathrm{E}[\boldsymbol{\rho}]}\sqrt{\frac{1}{N}\sum_{k}^{N}(\rho_k - \mathrm{E}[\boldsymbol{\rho}])^2}, \quad (17)$$

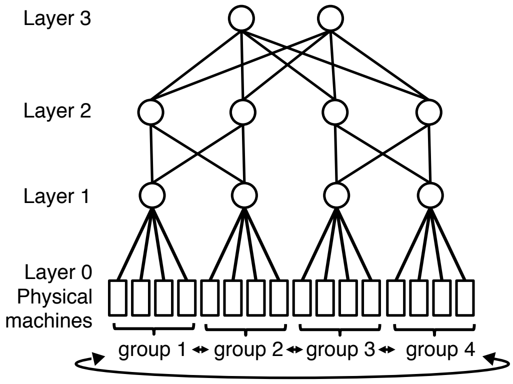

Fig. 4 DCN topology

where $\rho_k = \sum_{i \in \psi k} \rho_i^{(PM)}$, ψ_k is the set of nodes in PM k, and $\rho_i^{(PM)}$ is VM i.'s load. E[$\boldsymbol{\rho}$] is the average of PM loads, which is given by

$$E[\rho] = \frac{1}{N} \sum_{k=1}^{N} \rho_k. \qquad (18)$$

Hereafter, in simulation, we generate environmental fluctuations to evaluate the effectiveness of the proposed method. We focus on fluctuation in traffic rate and the number of VM. In DCN, there are several kinds of fluctuation in traffic rate, but control mechanisms should support them possibly. Hence, we prepare very different traffic rate settings and compare node concentrations when using them. Also we evaluate the proposed method using the different number of VMs as a fluctuation in the number of VM.

In this experiment, we use the parameter configuration shown in Table 1 Each VM communicates with a randomly-chosen N_H (average) VMs with high traffic rate r_H, and other VMs with low traffic rate r_L. At the start of each simulation run, we place n VMs in a randomly-chosen PM. At each simulation time unit, a VM uses the node action Eq. (2) to determine if it should change to another PM according to the MCMC-based ADM.

When we use the proposed method, VM assignment is performed by the MCMC-based ADM, and control parameter λ_i used in node i is adjusted by Eq. (2). When we use the previous method, we do not perform the adjustment of the λ.

Table 1 Parameter configuration

number of VMs, n	[200,800]
number of PMs, N	16
internal cost in a PM	0.0001
link cost	0.1
simulation time	20, 000
high traffic rate, r_H	10
low traffic rate, r_L	0.1
average number of high traffic VMs, N_H	$0.1 \times N$
α	0

4.2. Robustness against Fluctuations

4.2.1. Regarding fluctuation in traffic rate

To evaluate the effectiveness of the proposed method for improving the robustness against traffic rate fluctuations, we compare node concentrations when using the following two traffic rate settings:

traffic rate setting 1
Each VM communicates with a randomly-chosen N_H (average) VMs with high traffic rate r_H, and other VMs with low traffic rate r_L.

traffic rate setting 1
A half VMs communicates with a randomly-chosen $N_H \times 2$ (average) VMs with high traffic rate r_H and others with low traffic rate r_L. Further the other half VMs communicates with a randomly-chosen N_H (average) VMs with high traffic rate r_H and other VMs with low traffic rate r_L.

If the node concentrations for fluctuation models is the same, we can confirm the robustness of the proposed method against fluctuations in the number of VM fluctuations in an indirect way.

Figure. 5 shows percentage of nodes assigned in each PM in different traffic rate settings when using traffic settings 1 and 2 in the proposed method and the previous method. According to Fig. 5, the previous method cannot retain the node concentration for different traffic rate settings. On the contrary, the proposed method can retain the node concentration. Therefore, we can confirm that the proposed method improves the robustness of the MCMC-based ADM for severe fluctuations in traffic rates.

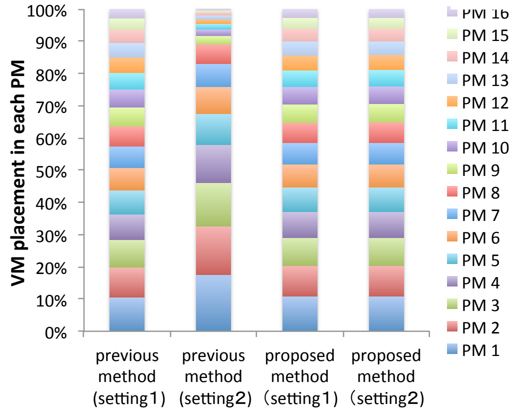

Fig. 2 Percentage of nodes assigned in each PM using traffic settings 1 and 2 in the proposed method and the previous method (n= 200)

Figure. 6 shows percentage of nodes assigned in each PM for different values of parameter K when using the proposed method. This figure shows that the node concentration becomes strong as control parameter K increases. Hence, by changing K, we can adjust the strength of the node concentration against environmental fluctuation.

Figure. 7 shows coefficient of variation CV($\boldsymbol{\rho}$) for different values of control parameter K of the proposed method in traffic rate settings 1 and 2. According to this figure, we confirm CV($\boldsymbol{\rho}$) for different traffic rate settings are almost same regardless the different value of control parameter K. Therefore, the proposed method can retain the strength of the node concentration regardless of the control parameter and traffic rate settings. We also obtain the same conclusion from the result of probability density function shown in Fig. 8.

From Figsure. 5, 7 and 8, the results for very different traffic rate settings 1 and 2 are the same. Hence, we clarify the robustness of the proposed method against traffic rate fluctuations in an indirect way.

4.2.2. Regarding fluctuation in the Number of VM

Figure. 9 shows coefficient of variation CV($\boldsymbol{\rho}$) for different values of parameter K of the proposed method. According to this figure, we confirm coefficient of variation CV($\boldsymbol{\rho}$) takes similar value in the different number of VM. Also, the proposed method can retain the strength of the node concentration regardless of the different value of

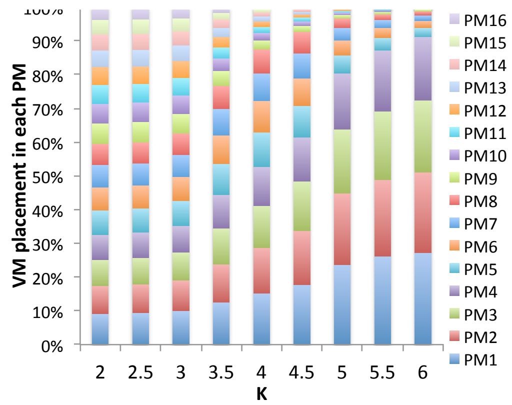

Fig. 6 Percentage of nodes assigned in each PM using traffic settings 1 in the proposed method (n = 200)

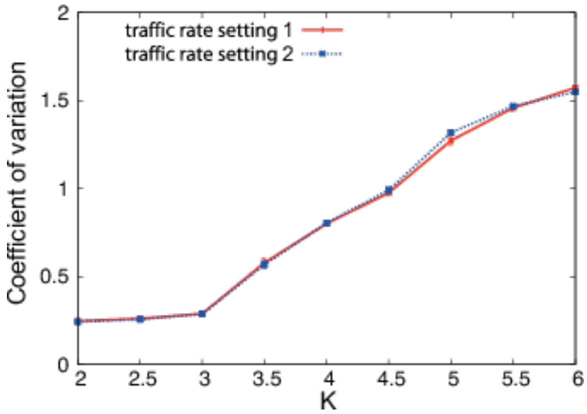

Fig. 7 Coefficient of variation CV($\boldsymbol{X}$) for different values of parameter K of the proposed method in traffic rate settings 1 and 2

control parameter K. Hence, we clarify the robustness of the proposed method against fluctuation in the number of VM.

4.3. Another performance

The previous method can place VMs with high traffic rates r_{ij} into PMs with small $d(x_i,x_j)$. This property of the previous method may be removed by the adjustment of control parameter λ_i using Eq. (2) in the proposed method. Since Eq. (2) normalizes control parameter λ_i by $\sum_{j \in \chi i} r_{ij}$, the difference of traffic rate between VMs may have an insignificant effect on VM assignment. Hence, we confirm that the proposed method has the property of the previous method.

Figure. 10 shows M in the yaxis and coefficient of variation CV($\boldsymbol{\rho}$) in the xaxis when using the proposed method and the previous method. In this

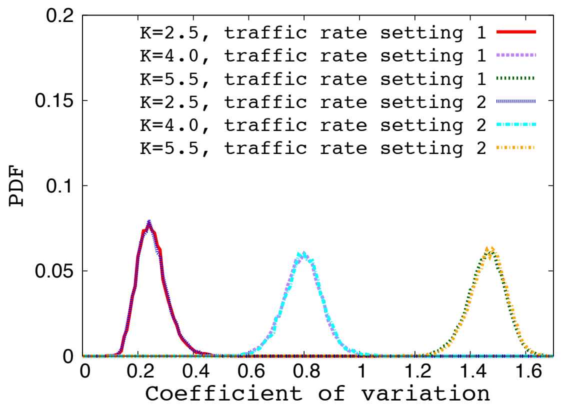

Fig. 8 Probability density function of coefficient of variation CV($\boldsymbol{\rho}$) in fluctuation of traffic rate settings

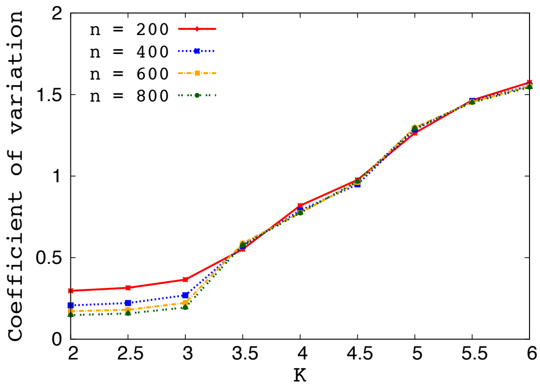

Fig. 9 Coefficient of variation CV($\boldsymbol{\rho}$) for different values of parameter K of the proposed method

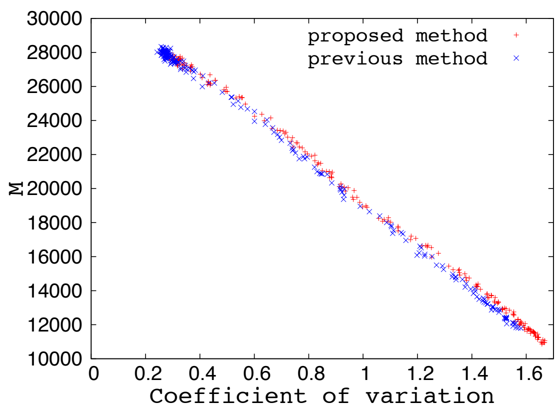

Fig. 10 Coefficient of variation CV($\boldsymbol{\rho}$) vs. M using the proposed method and previous method

figure, we use traffic rate setting 2 and n = 200. Note that we have already got almost the same results using the traffic rate setting 1. If the proposed method has the same property of the previous method, the curve M -CV($\boldsymbol{\rho}$) for both methods are almost same. According to Fig. 10, we can confirm that this is true.

5. Conclusion and Future Work

In this paper, we discussed the impact of environmental fluctuation on the MCMC-based ADM, and proposed a method to enhance the robustness of the MCMC-based ADM against severe environmental fluctuations. In particular, we designed an adjustment method to absorb the impact of environmental fluctuations according to the outcome of our discussion. In the proposed method, each node autonomously adjusts its control parameter of the MCMC-based ADM. Then, similarly to [9], we applied the proposed method to the VM assignment problem in data centers. We performed the simulation experiment using the application, and clarified the effectiveness of the proposed method for severe environmental fluctuations.

As future work, we are planning evaluate the proposed method against several environmental fluctuations (e.g., the change in the number of node, and state distance).

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 15K15985.

REFERENCES

[1] "2014 WHITE PAPER" http://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2014/2014-index.html

[2] "Google Date Centers. " http://www.google.com/about/datacenters/inside/locations/index.html.

[3] M. Aida, "Using a renormalization group to create ideal hierarchical network architecture with time scale dependency," IEICE Transactions on Communications, vol. E95-B, no. 5, May 2012.

[4] C. Takano, M. Aida, M. Murata and M. Imase, "Proposal for autonomous decentralized structure formation based on local interaction and back-diffusion potential," IEICE Transactions on Communications, vol. 95-B, no. 5, pp. 1529-1538, May 2012.

[5] Y. Takahashi, C. Takano, Y. Sakumoto and M. Aida, "Transmission rate control utilizing chaotic nature of coupled oscillators," in Proceedings of INCoS 2012, pp. 126-131, Sep. 2012.

[6] K. Takagi, C, Takano, M. Aida and M. Naruse, "New autonomous decentralized structure formation based on Huygens' principle and renormalization for mobile ad hoc networks,"

International Journal on Advances in Intelligent Systems, vol. 7, no. 1 and 2, pp. 64-73, July 2014.

[7] V. A. Cicirello and S. F. Smith, "Ant colony control for autonomous decentralized shop floor routing," In Proceedings of ISADS 2001, pp. 383-390, Mar. 2001.

[8] K. Leibnitz, N. Wakamiya and M. Murata, "Biologically inspired self-adaptive multi-path routing in overlay networks," Communications of the ACM, vol 49, no. 3, pp. 62-67, 2006.

[9] Y. Sakumoto, M. Aida and H. Shimonishi, "An autonomous decentralized control for indirectly controlling system performance variable in large-scale and wide-area network," in Proceedings of NETWORKS 2014, pp. 1-7, Sep. 2014.

[10] Y. Sakumoto, M. Aida and H. Shimonishi, "An autonomous decentralized adaptive function for retaining control strength in large-scale and wide-area system," in Proceedings of GLOBECOM 2014, pp. 1958-1964, Dec. 2014.

[11] Y. Sakumoto, M. Aida and H. Shimonishi, "An autonomous decentralized control for indirectly controlling system performance variable in large-scale and wide-area network," in Proceedings of Networks 2014, pp. 1-7, Sep. 2014.

[12] M. Yokota, Y. Sakumoto and M. Aida, "A Method for Enhancing Robustness of MCMC-Based Autonomous Decentralized Control," in Proceedings of NOLTA 2015, Dec. 2015.

[13] W. Hastings, "Monte carlo sampling methods using markov chains and their applications," Biometrika, vol. 57, no. 1, pp. 97-109, 1970.

[14] W.R. Gilks, S. Richardson and D. J. Spiegelhalter, "Markov chain Monte Carlo in Practice, " vol. 2. CRC press, 1996.