Received June 13, 2018, accepted August 1, 2018, date of publication August 13, 2018, date of current version September 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2865114

An Anomaly Detector Deployment Awareness Detection Framework Based on Multi-Dimensional Resources Balancing in Cloud Platform

JUN LIU[1], HANCUI ZHANG[2], AND GUANGXIA XU[1,3]

[1]College of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[2]School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China
[3]Information and Communication Engineering Postdoctoral Research Station, Chongqing University, Chongqing 400044, China

Corresponding author: Jun Liu (liujuncqcs@163.com)

This work was supported in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJ1704081, in part by the National Natural Science Foundation of China under Grant 61772099 and Grant 61572090, in part by the Chongqing Research Program of Basic Research and Frontier Technology under Grant cstc2017jcyjAX0270, in part by the Artificial Intelligence Technology Innovation Important Subject Projects of Chongqing under Grant cstc2017rgzn-zdyf0140, in part by the Innovation and Entrepreneurship Demonstration Team Cultivation Plan of Chongqing under Grant cstc2017kjrc-cxcytd0063, in part by the China Postdoctoral Fund under Grant 2014M562282, and in part by the Project Postdoctoral Supported in Chongqing under Grant Xm2014039.

ABSTRACT Anomaly detection has been an important topic in cloud platforms to guarantee the dependability and robustness of services in the cloud. Most research works were proposed to tackle the detection performance problems of detection algorithms caused by the volume of data, the dynamic environment, various types of anomalies, and so on. However, almost all of them take only the optimization of algorithms into account, which leads to a situation that some key features of detector deployment and the scalability and dependability of the detection framework itself are omitted. Therefore, an anomaly detector deployment awareness detection framework based on multi-dimensional resources balance is proposed to address the problems. It balances the multi-dimensional resources by bringing four factors of resources into consideration to deploy detectors quietly, which are the current utilizations, the available capacity, the demands of detectors and the remaining resources. Three experiments and comparative analysis suggest that this framework achieves a higher scalability and detection accuracy than the existing framework.

INDEX TERMS Anomaly detection, detector deployment, resource balance, cloud platform.

I. INTRODUCTION

There is no doubt that cloud computing brings a more efficient and cost-effective way for individuals and communities to access applications and request services quickly and conveniently, no matter when and where [1]–[3]. It makes use of virtualization technology [4] to consolidate existing hardware and software resources, and it allows clients to deploy their own applications in this type of virtual machine on demand. However, with the immense growth of users' requests, virtual machines as the execution unit of the cloud platform are always initialized, migrated or deleted dynamically to meet the specific requirements and tasks. Thus, the cloud service availability may not always be guaranteed. An awareness challenge of the cloud computing platform arising here is the dependability and availability of the cloud platform.

A notable amount of studies have already focused on addressing the problems, such as anomaly detection, intrusion detection, fault tolerance and so on with the growing numbers of anomaly detection algorithms and methods. Statistical-based algorithms are used to assume a distribution or probability model for the data to identify anomalies [5], [6]. Classification-based algorithms discriminate anomalies from the normal data class on the basis of a labeled class [7], while clustering-based algorithms group or cluster data based on similarity to detect an outlier [8], [9]. With the fast-growing scale of the cloud platform and the tremendous amount of running data, some hybrid detection methods are proposed to confront the highly dynamic platform and mass of data [10]–[12]. Nevertheless, detection accuracy improvement is often achieved based on the algorithm

optimization, but it ignores the impact of the detector deployment environment.

In the cloud platform, the large-scale and short-lived characteristics of virtual machines make it difficult to ensure the availability and accuracy of an anomaly detection algorithm with the runtime going on. To effectively improve the scalability and reliability of the anomaly detection mechanism, this paper uses a dedicated server to deploy the detector virtual machines to avoid the impact of the detection performance of users' virtual machines caused by anomaly detector deployment. The detector virtual machine as a special node of anomaly detection can improve the reliability and scalability of the detection system. The same as with the deployment of client virtual machines, the deployment of detector virtual machines requires a dynamic resources configuration according to the size of the detection task (i.e., the size of the detected object). Therefore, one of the major challenges in detector deployment is related to optimizing the resources being allocated to virtual machines and balancing the resource utilization.

In contrast to other virtual machine migration-based deployment policies, this paper focuses on the importance of the initial placement [13]–[16] in order to avoid the consumption of time and computing resources in migration. Meanwhile, in order to maximize the reliability and scalability of the detector virtual machine, the multi-dimensional resources in terms of load balancing is taken into consideration [17], [18].

The contributions of this paper are summarized as follows:

(1) An anomaly detector deployment awareness detection framework based on multi-dimensional resources balance (DDAF for short) is proposed. The detection framework leverages a multi-resource balanced deployment policy to facilitate cloud resource management.

(2) A deployment of the detector virtual machine is presented, which takes more care of the importance of the initial placement rather than live migration. Moreover, the multi-dimensional resources of the cloud platform, such as the CPU, memory, bandwidth, I/O and so on, are taken into account in terms of load balancing in order to achieve the dependability and robustness of the anomaly detector system on the cloud platform.

The rest of this paper is organized as follows. In the related works section, some anomaly detection frameworks are discussed. The architecture of the proposed detection framework is demonstrated in section 3. It is followed by the multi-dimensional resource analysis and the detector deployment policy in section 4. Experiments and results are described in section 5. Section 6 presents the conclusions.

II. RELATED WORK

The anomaly detection of virtual machines is an important topic in the cloud platform running environment. Various factors were considered in the process of anomaly detection, such as the detection latency, accuracy, false alarm rate, anomaly types, the volume and dimension of data and so on.

Lv *et al.* [19] presented a density-based clustering algorithm with attention given to the complex structured datasets, which improved the traditional locality sensitive hashing method to implement the fast querying of the nearest neighbors. Several definitions are redefined on the basis of the influence space of each object and the core density that is reachable based on the influence space is proposed, which aims to distinguish between border objects and noisy objects.

Jindal *et al.* [20] proposed a top-down scheme based on the decision tree and support vector machine methods for theft detection. The data first processed by the DT are fed as an input to the SVM classifier to detect and locate real-time electricity theft.

Garg and Batra [21] presented a hybrid detection method that uses a combination of a fuzzy k-means clustering algorithm, an extended Kalman filter and support vector machines to detect anomalies. Features are extracted by the FKM and the optimization of features is done by the extended Kalman filter. The detection of anomalies is performed by the SVM. This technique leads to well-classified data, a low false positive rate, and a high detection rate compared with the earlier developed techniques.

Guiping and Jiawei [22] proposed an anomaly detection framework in the consideration of imbalanced training data, an increasing number of training data and multiple anomaly category classifications. To cope with these challenges, several support vector machine (SVM) based anomaly detection algorithms are implemented and equipped, including the C-SVM, OCSVM, multi-class SVM, imbalanced SVM, and online learning SVM.

Guan *et al.* [23] proposed a cloud dependability analysis framework to analyze the correlation of various performance metrics with failure occurrences in virtualized and non-virtualized environments. Therefore, the function of this cloud dependability analysis framework is not comprehensive enough. It just gains insight into the impact of virtualization on the cloud dependability instead of automatically detecting anomalous virtual machines.

Pannu *et al.* [24] presented a self-evolving anomaly detection framework to recursively explore newly generated verified detection results for future detection. Statistical learning technologies are exploited in detector determination and working dataset selection. The one-class SVM and SVM are adopted for anomaly detection.

Above all, various anomaly detection methods were adopted to enhance the performance of the accuracy and efficiency. Hybrid algorithms combined with Machine learning, neural networks, probability and statistics, etc. have been used sufficiently along with evolving and self-adaptive frameworks. Although the accuracy and efficiency of the detection algorithm is essential, the dependability and scalability of the framework takes the tendency. Thereby, in addition to the detection algorithm itself, this paper places strong emphasis on the efficiency and scalability of the detection system based on detector deployment in the cloud platform.

III. A FRAMEWORK OF ANOMALY DETECTION BASED ON DETECTOR DEPLOYMENT POLICY

In regard to anomaly detection, what people pay the most attention to is the accuracy and efficiency of the anomaly detection algorithm or the method itself and there is a lack of concern about the scalability and dependability of the detection system.

To gain insight into the impact of the deployment of the detector virtual machine itself, this paper proposes a detector deployment awareness anomaly detection framework in the cloud platform, which leverages a multi-resource balanced deployment policy to facilitate cloud resource management.

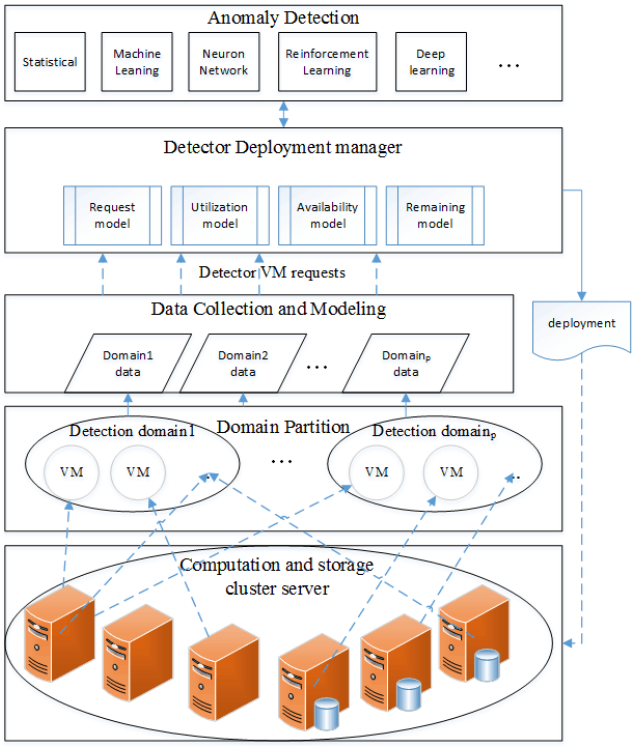


FIGURE 1. The framework of anomaly detection based on the detector deployment policy.

As illustrated in Figure 1, the proposed anomaly detection framework includes 4 modules. A set of client virtual machines that runs on top of physical servers are clustered according to the runtime environment in the detection domain partition module. Each virtual machine encapsulates the execution states of the specific running applications and cloud servers. The runtime performance metrics as well as the running environment's attribute data are monitored and collected by the data collection and modeling module. When a set of collections is submitted to the detector deployment module with its computation and storage requirements in the detection domain, it evaluates the qualifications of the requirements, the availability and the utilization ratio of multi-dimensional resources on the cloud server.

As is known, a number of virtual machines deployed on the same host will share and compete with resources. To eliminate the influence of the deployment of a detector during performance metrics collection and anomaly detection, it is necessary to deploy the detector on the most suitable physical server on the basis of multi-dimensional resource balance. The detector deployment manager is responsible for the resource management and detector deployment. The four models including request, utilization, availability and remaining are the main factors that matter to the deployment policy. In each detector, a variety of algorithms and methods for detecting anomalies are dispatched as a core component of the anomaly detection module that conducts the anomaly detection processes.

IV. DETECTOR VIRTUAL MACHINE DEPLOYMENT POLICY

In contrast to other placement methods, the deployment of the detector virtual machine presented in this paper takes more care of the importance of the initial placement rather than live migration. Meanwhile, in particular, the multi-dimensional resources of the cloud platform, such as the CPU, memory, bandwidth, I/O and so on, are taken into account in terms of load balancing in order to achieve the dependability and robustness of the anomaly detector system on the cloud platform.

Suppose that there are two physical machines PM_a and PM_b, and the normalized values of the residual resources under two dimensions (CPU and Memory) are (0.66,0.25) and (0.4, 0.36), respectively. The available capacity of these two physical machines can be simply calculated as the product of each dimension of residual resources, which is 0.165 and 0.144. Normally, when a new request of detector virtual machine deployment comes with the normalized value (0.2, 0.2), the physical machine with the larger available capacity is given greater priority than the smaller one. However, after comparing the deployment situation of detector virtual machines on these two physical machines in Figure 2, it can be found that there are 46 percent of CPU resources available in PM_a, while the memory resource is almost gone. In contrast, PM_b, the smaller available capacity one, remains relatively balanced before and after the deployment of the detector, and the remaining resources can be further used.

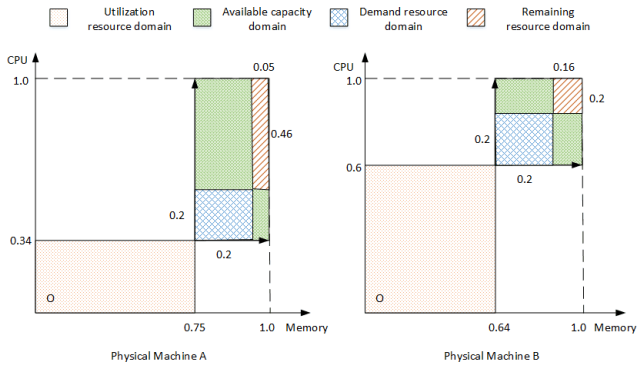


FIGURE 2. The resources consumptions of PM_a and PM_b.

Thus, in order to improve the dependability and scalability of the anomaly detection system on the basis of load balancing, in addition to the available capacity and demand resources, the ratio relationships of the utilized resources and the remaining resources also play an important role to retain resource balancing and servers' availability and dependability. This detector virtual machine deployment is much more than a client virtual machine deployment. The deployment requests are connected with the running environment and it is essential to ensure the running environment balancing before and after deployment to reduce the false alarm rate.

Hence, to deploy the detector virtual machines quietly, four factors are addressed to manage the deployment of detectors: the ratio relationships and values of the utilization of resources among each dimension, the ratio relationships and values of the availability of resources among each dimension, the ratio relationships and values of the remaining resources among each dimension, and the ratio relationships and values of the demanded resources among each dimension.

The steps are as follows.

Step 1: Get a set of requests of detector virtual machine deployment demand $I = \{DVM_1, DVM_2, DVM_3, \ldots, DVM_m\}$, $DVM_i =< CPU_i, Mem_i, BandWidth_i, IO_i >$. Based on the demand resources of detector virtual machines, some candidate physical machines $PH_{candidate}$ are collected by the formula from the detection domain.

$$PH_{candidate} = \{PH_i | CPU_{dem} < CPU_{rem} \cap < Mem_{dem} Mem_{dem} \cap BandWidth_{dem} < BandWidth_{rem} \cap IO_{dem} < IO_{rem}, i = 1, 2, \ldots, n\} \quad (1)$$

Step 2: According to the previous analysis, the more unbalanced the ratio relationships of remaining resources among each dimension after deployment, the more likely that it may fall into resource wastage and the service being unavailable. Therefore, it is an effective way to calculate the probable remaining resources vector $PROBABLE_{PHi}$ after the deployment of the detector virtual machine.

$$PROBABLE_{Phi} = REMAINDER_{PHi} - DVM_j \quad (2)$$

where $REMAINDER_{PHi}$ is the current remainder of physical machine PH_i in $PH_{candidate}$ before detector deployment.

Step 3: Use the cosine similarity between the utilization resources vector and demand resources vector to update the candidate physical machines in $PH_{candidate}$.

$$PH_{cand_update} = \{PH_i | UTILIZE_{PHi} \times DVM_j \leq Threshold\} \quad (3)$$

where $UTILIZE_{PHi}$ is the current utilization vector of the physical machine PH_i in $PH_{candidate}$ and DVM_j is the demanded resources vector.

It is obvious that the resource vectors are positive since they are all included in a multi-dimensional resources cube with the angle in the range of $[0, \pi/2]$. With the cosine similarity, a smaller angle between two vectors results in more similarity. Therefore, if the utilization resource vector is more similar to the demand resource vector, an imbalance of multi-dimensional resources will emerge. Accordingly, the updating method with a predefined threshold is helpful to reduce the search scope and retain the resource balancing.

Step 4: Pinpoint the most suitable physical machine PH_i to deploy the detector in consideration of the relationships between the probable remaining resources vector $PROBABLE_{PHi}$ in PH_{cand_update} and the total resource vector Vtotal.

$$PH_{suit} = \{PH_i | max(PROBABLE_{PHi} \times V_{total}), \forall PH_i \in PH_{cand\_update}\} \quad (4)$$

A physical machine PH_i with the maximum score of the dot product indicates that the probable remaining resource has much more similarity with the total resource vector, which means that the idea balancing of the multi-dimensional resources will be reached. Thus, it eases the influence of the detector deployment on the accuracy of the anomaly detection due to the change of the runtime environment related to the multi-dimensional resources.

V. EXPERIMENTS AND RESULTS

To evaluate the performance of the proposed detector virtual machine deployment-based detection framework (DDAF), three experiments are conducted on a private cloud platform built with OpenStack [25], [26]. 6 physical servers with Intel Xeon 3.3 GHz CPUs and 8 Gb of RAM are used, all of them use the CentOS7 operating system and the virtualization management program is Xen [27]. The runtime performance metrics data of the client virtual machines in the private cloud platform is collected by the tools xentop and libvirt [28]. The detector virtual machine requests are produced by the different model of runtime performance metrics data due to different types of fault injections, which are CPU hog, memory hog or network hog [29]–[31].

Experiment 1: The requirements of detector virtual machines' deployment were collected every 10 minutes by the data collection module under the different detection domains. The experiment is done to evaluate the load balance of the utilizations of CPU and Memory resources before and after detector virtual machines' deployment.

As shown in Figure 3 and Figure 4, along with the increasing of the detector virtual machines requests, the utilizations of the CPU and memory also increase. Nonetheless, what is worth paying close attention to is that although they are rising in different spaces, they remain relatively balanced. It is very helpful to keep the runtime environment of the client virtual machines stable during the detector virtual machines' deployment to maintain the detection accuracy.

Experiment 2: The average completion time and the demand satisfaction of the detector virtual machines requests are used to estimate the performance of the scalability and dependability of the proposed detection framework. A various number of requests and types will be collected by the collection module. How efficient and effective the proposed

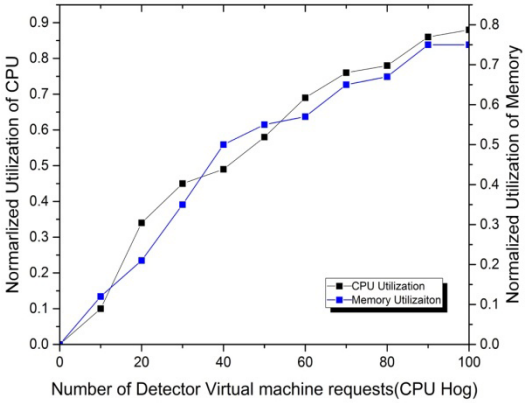


FIGURE 3. The resource utilizations of the CPU and Memory over the number of requests (CPU Hog).

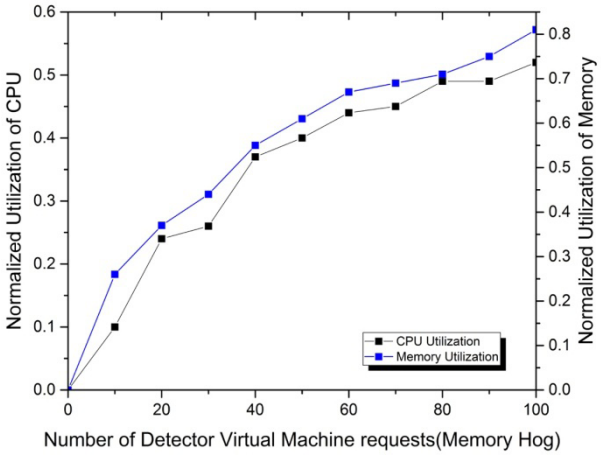


FIGURE 4. The resource utilizations of the CPU and Memory over the number of requests (Memory Hog).

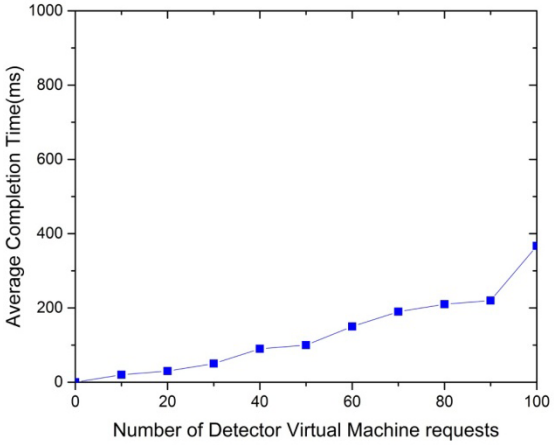


FIGURE 5. The average completion time over the various numbers of requests.

deployment policy is at satisfying demands is critical to the performance of the service. As shown in Figure 5, we vary the number of detector virtual machine requests from 0 to 100.

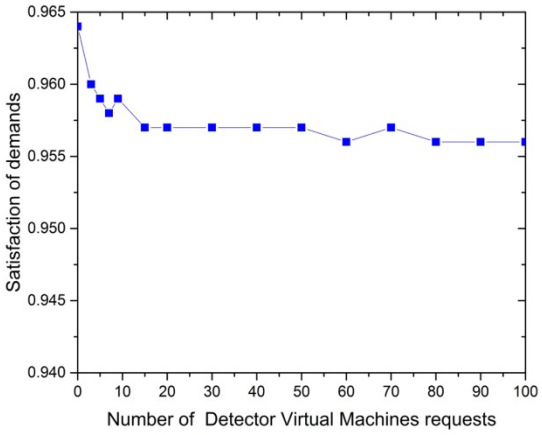


FIGURE 6. The satisfaction of demands over the various numbers of requests.

The proposed detector deployment policy takes less than 300 ms to finish all the requests. To illustrate the satisfactions of requests, the fraction of the total application demands, which is described as $(\sum_i^m DVM_i \sum_j^N REMAINDER_j)/\sum_i^m DVM_i$, is used and the results are plotted in Figure 6. The demand satisfaction remains approximately 0.956 as the requests increase, thus indicating that the deployment policy can produce high-quality solutions regardless of the problem size.

Experiment 3: The influence of detector deployment on the anomaly detection is taken into consideration. We compare this proposed detector deployment-based detection framework (DDAF) with the environment awareness detection framework (EaAD) presented in [22]. First, we deploy the same operation environment and the number of client virtual machines (100), and then simulate an anomalous environment by injecting a fault (memory leak, CPU Hog and network Hog) into them randomly to produce concurrent requests. Figure 7 shows the comparison of the detection accuracy between these two detection frameworks as the detection virtual machines' deployment requests increase.

The experimental result shown in Figure 7 varies as the number of detector virtual machines requests varies from 0 to 100 due to the fault injection. As the request size increases, the proposed DDAF outperforms EaAD. More importantly, the fluctuation of the detection performance changes as the EaAD increases dramatically, while the DDAF remains stable.

In summation, the proposed detector deployment-based detection framework significantly and consistently outperforms others in all three aspects: resource balancing, scalability, and detection accuracy. The ability to achieve a higher balancing of multi-dimension resources is mainly due to its combination with four ratio factors of resources, which are utilization, demand, availability and remaining. The fast speed and high scalability are mainly due to the strategy that refines the searching range by the cosine similarity of the utilization and demand resources vectors. Meanwhile, due to

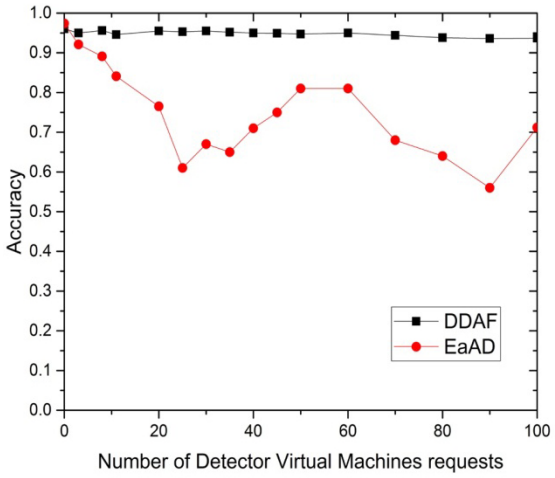


FIGURE 7. The detection accuracy over various numbers of requests.

the multi-resources balancing and scalability of the detection framework, it achieves a perfect detection accuracy.

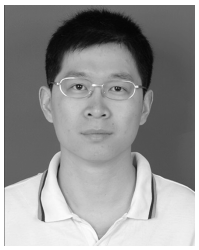
VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a detector deployment-based detection framework in response to the low scalability and dependability of cloud services. The framework preserves cloud service availability through the fast and balanced deployment of detector virtual machines. Four factors are considered: the current utilization ratio and value of multi-dimensional resources, the available capacity, the demand ratio and value of requests and the possible remaining ratio and value of multi-dimensional resources. Three experiments are conducted to evaluate the efficacy and scalability of the proposed framework and the results are encouraging.

As part of future work, to reflect a more complete running environment of the cloud platform of the short-lived feature of virtual machines, proactive migrations will be incorporated into the system based on some prediction algorithms. Another direction is to incorporate multi source data acquisition and fusing technology into detection applications to optimize detection.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," *Commun. ACM*, vol. 53, no. 6, p. 50, Jun. 2010.
[2] S. Marston *et al.*, "Cloud computing—The business perspective," *Decis. Support Syst.*, vol. 51, no. 1, pp. 176–189, Apr. 2011.
[3] W. Bai and W. Geng, "Research on operation management under the environment of cloud computing data center," *Psychol. Rev.*, vol. 119, no. 3, pp. 573–616, 2012.
[4] P. Barham *et al.*, "Xen and the art of virtualization," in *Proc. 19th ACM Symp. Oper. Syst. Principle (SOSP)*, 2003, pp. 164–177.
[5] C. Wang *et al.*, "Statistical techniques for online anomaly detection in data centers," in *Proc. 12th IFIP/IEEE Int. Symp. Integr. Netw. Manage.*, May 2011, pp. 385–392.
[6] M. Solaimani, M. Iftekhar, L. Khan, and B. Thuraisingham, "Statistical technique for online anomaly detection using spark over heterogeneous data from multi-source VMware performance data," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2014, pp. 1086–1094.
[7] M. Fugate and J. R. Gattiker, "Computer intrusion detection with classification and anomaly detection, using SVMs," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 17, no. 3, pp. 441–458, 2003.
[8] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 551–556.
[9] J. Liu, S. Chen, Z. Zhou, and T. Wu, "An anomaly detection algorithm of cloud platform based on self-organizing maps," *Math. Problems Eng.*, vol. 2016, Mar. 2016, Art. no. 3570305.
[10] G. Wang, S. Chen, and J. Liu, "Anomaly-based intrusion detection using multiclass-SVM with parameters optimized by PSO," *Int. J. Secur. Appl.*, vol. 9, no. 6, pp. 227–242, 2015.
[11] H. Zhu, Y. Xin, and F. Wang, "A novel framework for anomaly detection based on hybrid HMM-SVM model," in *Proc. IEEE Int. Conf. Broadband Netw. Multimedia Technol.*, Oct. 2011, pp. 670–674.
[12] B. Li, S. Zhang, and K. Li, "Towards a multi-layers anomaly detection framework for analyzing network traffic," *Concurrency Comput. Pract. Exper.*, vol. 29, no. 14, p. e3955, Jul. 2017.
[13] X. Li, Q. He, J. Chen, K. Ye, and T. Yin, "Informed live migration strategies of virtual machines for cluster load balancing," in *Proc. 8th IFIP Int. Conf. Netw. Parallel Comput. (NPC)*, 2011, pp. 111–122.
[14] P. Lu, A. Barbalace, R. Palmieri, and B. Ravindran, "Adaptive live migration to improve load balancing in virtual machine environment," in *Proc. Eur. Conf. Parallel Process.*, 2013, pp. 116–125.
[15] Y. Zhao and W. Huang, "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud," in *Proc. Int. Joint Conf. INC, IMS IDC*, Aug. 2009, pp. 170–175.
[16] T. Wood *et al.*, "Black-box and gray-box strategies for virtual machine migration," in *Proc. 4th USENIX Conf. Netw. Syst. Design Implement.*, 2009, p. 17.
[17] Z. Zhu and Q. Zhang, "Resource scheduling with load balance based on multi-dimensional QoS and cloud computing," *Comput. Meas. Control*, vol. 1, pp. 263–265, Jan. 2013.
[18] S. Singh and I. Chana, "QRSF: QoS-aware resource scheduling framework in cloud computing," *J. Supercomput.*, vol. 71, no. 1, pp. 241–292, 2015.
[19] Y. Lv *et al.*, "An efficient and scalable density-based clustering algorithm for datasets with complex structures," *Neurocomputing*, vol. 171, pp. 9–22, Jan. 2016.
[20] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, and S. Mishra, "Decision tree and SVM-based data analytics for theft detection in smart grid," *IEEE Trans. Ind. Informat.*, vol. 12, no. 3, pp. 1005–1016, Jun. 2016.
[21] S. Garg and S. Batra, "A novel ensembled technique for anomaly detection," *Int. J. Commun. Syst.*, vol. 30, no. 11, p. e3248, Jul. 2017.
[22] G. Wang and J. Wang, "An anomaly detection framework for detecting anomalous virtual machines under cloud computing environment," *Int. J. Secur. Appl.*, vol. 10, no. 1, pp. 75–86, 2016.
[23] Q. Guan, C. C. Chiu, and S. Fu, "CDA: A cloud dependability analysis framework for characterizing system dependability in cloud computing infrastructures," in *Proc. IEEE Pacific Rim Int. Symp. Dependable Comput.*, Nov. 2012, pp. 11–20.
[24] H. S. Pannu, J. Liu, and S. Fu, "A self-evolving anomaly detection framework for developing highly dependable utility clouds," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2012, pp. 1605–1610.
[25] J. M. A. Calero and J. G. Aguado, "MonPaaS: An adaptive monitoring platformas a service for cloud computing infrastructures and services," *IEEE Trans. Services Comput.*, vol. 8, no. 1, pp. 65–78, Jan. 2015.
[26] L. Sha, J. Ding, X. Chen, X. Zhang, Y. Zhang, and Y. Zhao, "Performance modeling of openstack cloud computing platform using performance evaluation process algebra," in *Proc. IEEE Int. Conf. Cloud Comput. Big Data*, Nov. 2015, pp. 49–56.
[27] S. G. Soriga and M. Barbulescu, "A comparison of the performance and scalability of Xen and KVM hypervisors," in *Proc. Int. Conf. Netw. Educ. Res. (RoEduNet)*, Sep. 2014, pp. 1–6.
[28] J. Ye and Y. Shang, "An overview of open-source virtualization technology," in *Proc. Int. Conf. Smart Sustain. City*, 2013, pp. 221–224.
[29] X. Pan *et al.*, "Ganesha: BlackBox diagnosis of MapReduce systems," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 37, no. 3, pp. 8–13, 2010.
[30] J. E. Marynowski, A. O. Santin, and A. R. Pimentel, "Method for testing the fault tolerance of MapReduce frameworks," *Comput. Netw.*, vol. 86, pp. 1–13, Jul. 2015.
[31] T. Miyachi *et al.*, "Fault injection on a large-scale network testbed," in *Proc. Asian Internet Eng. Conf.*, 2011, pp. 4–11.



JUN LIU received the B.S. degree in software engineering and the Ph.D. degree in computer science and technology from Chongqing University, China, in 2008 and 2016, respectively. Since 2016, he has been with the School of Software Engineering, Chongqing University of Posts and Telecommunications, China. His research interests include anomaly detection, machine learning, big data analytics, NAND flash memory, information security, and cloud computing.

HANCUI ZHANG received the Ph.D. degree from the School of Big Data and Software Engineering, Chongqing University, China, in 2018. Since 2018, he has been with the School of Information Science and Technology, Zhejiang Sci-Tech University, China. Her current interests include cloud computing, anomaly detection, large scale data mining, and NAND flash memory.

GUANGXIA XU received the M.S. and Ph.D. degrees in computer science from Chongqing University, China. She is currently a Professor at the Chongqing University of Posts and Telecommunications. She is also the Research Vice Director at the Network and Information Security Engineering Center, Chongqing, China. Her research interests include block-chain, big data analytics, and security AI. She is a Committee Member at the Fault Tolerant Computing of the China Computer Federation and a Vice Chairman of the Information Security Association, Chongqing, China.