International Journal of Future Generation Communication and Networking
Vol. 9, No. 3 (2016), pp. 255-268
http://dx.doi.org/10.14257/ijfgcn.2016.9.3.23

A Virtual Machine Instance Anomaly Detection System for IaaS Cloud Computing

Mingwei Lin[1*], Zhiqiang Yao[1], Fei Gao[1] and Yang Li[1]

[1]*Faculty of Software, Fujian Normal University, Fuzhou 350108, China*
linmwcs@163.com

Abstract

Infrastructure as a Service (IaaS) is one of the three important fundamental service models provided by cloud computing. It provides users with computing resource and storage resource in terms of virtual machine instances. Because of the rapid development of cloud computing, more and more application systems have been deployed on the IaaS cloud computing platforms. Therefore, once anomalies incur in the IaaS cloud computing platforms, all the application systems cannot work normally. In order to enhance the dependability of IaaS cloud computing platform, a virtual machine instance anomaly detection system is proposed for IaaS cloud computing platform to detect virtual machine instances that exhibit abnormal behaviors. The proposed virtual machine instance system consists of four modules that are the data collection, the data transmission, the data storage, and the anomaly detection. In order to reduce the computing complexity and improve the detection precision, the anomaly detection module introduces the principal components analysis to reprocess the collected data and then adopts the Bayesian decision theory to detect the abnormal data. Experimental results show that the proposed virtual machine instance anomaly detection system is effective.

Keywords*: IaaS cloud computing, Anomaly detection, Principal components analysis, Bayesian decision theory*

1. Introduction

With the rapid development of computer technology and the continuous innovation of the application model, cloud computing has become the research focus of the industrial and academic circles [1]. So far, a number of international IT companies such as Google, Microsoft, and Amazon have released their own cloud computing products [2]. They consider cloud computing as the key development direction in the near future.

Through cloud computing, users can obtain various resources and services on demand without worrying about the construction of hardware infrastructure and the investment of hardware resources [3]. Therefore, cloud computing reduces the cost of realizing the information management for small and medium-sized enterprises and improve the business efficiency of enterprises [4].

However, cloud computing still has a number of key problems that need to be solved [5]. In recent years, many famous cloud computing service providers such as Google and Amazon exhibit various faults frequently. In this case, peoples begin to worry about the dependability of cloud computing. For example, the Amazon Elastic Compute Cloud servers exhibited faults in April 2011 and famous applications deployed on the Amazon Elastic Compute Cloud could not be accessed for four days [6]. Emerging fault events block the rapid development and the spread of cloud computing significantly, so it is very important to improve the dependability of cloud computing. Cloud computing is a new service model and changes the use method of traditional software and

ISSN: 2233-7857 IJFGCN
Copyright ⓒ 2016 SERSC

hardware technologies. Existing systems designed for improving the dependability of host and network cannot be used for cloud computing directly [7].

In order to improve the dependability of IaaS cloud computing platform, this paper proposes a virtual machine instance anomaly detection system for IaaS cloud computing to detect virtual machine instances that show abnormal behaviors. The proposed system consists of four modules, which are the data collection, the data transmission, the data storage, and the anomaly detection. In order to reduce the computing complexity and improve the detection precision, the anomaly detection module uses the principal components analysis to preprocess the original collected data and then introduces the Bayesian decision theory to detect the abnormal data. Experimental results show that the proposed virtual machine instance anomaly detection system is effective.

The remainder of this paper is organized as follows. Section 2 reviews existing works on improving the dependability of cloud computing. Section 3 presents the architecture of the cloud data center and our proposed virtual machine instance anomaly detection system. Section 4 describes the detailed implementation of the anomaly detection module. Section 5 shows the performance evaluation. Finally, conclusions are drawn in Section 6.

2. Related Works

Cloud computing is a new commercial service model [8]. Due to its huge commercial value, it has been paid great attention by peoples over the whole world. However, emerging fault events of cloud computing not only make peoples be worried about the dependability of cloud computing, but also block the migration process of core business services from traditional IT infrastructure to cloud computing platform [9]. Currently, the research in cloud computing field focuses on functionality, performance, application, and energy consumption. Because of emerging fault events in cloud computing platforms, the research on improving the dependability of cloud computing is focused gradually [10].

Smith et al performed anomaly detection for cloud computing platforms by collecting data of performance metrics that are related with the health status of each physical node [11]. Wang et al proposed an entropy-based anomaly testing (EbAT) for online anomaly detection in cloud computing platforms [12]. The EbAT introduces entropy to measure the performance metric distributions. Li et al proposed a fast anomaly detection scheme for large scale data center [13]. The proposed scheme exploits the distributions of IP addresses and finds out the abnormal data that are caused by the network attacks.

In the engineering area, Amazon has designed and developed the CloudWatch for the Amazon Elastic Compute Cloud [14]. The CloudWatch provides a visual interface to monitor the change of performance metrics such as the virtual machine instance status, the resource utilization, the network traffic, the disk, and the I/O. Google also has developed the Dapper framework for the Google search engine to monitor its status, but the Dapper framework does not show the anomaly detection function [15]. Jerome Boulon et al developed an anomaly detection system called Chukwa for Hadoop platform [16]. The Chukwa system can monitor and analyze the running status of cloud services.

3. Proposed System

3.1. Architecture of Cloud Data Center

The typical architecture of cloud data center is shown in Figure 1.

 Copyright ⓒ 2016 SERSC

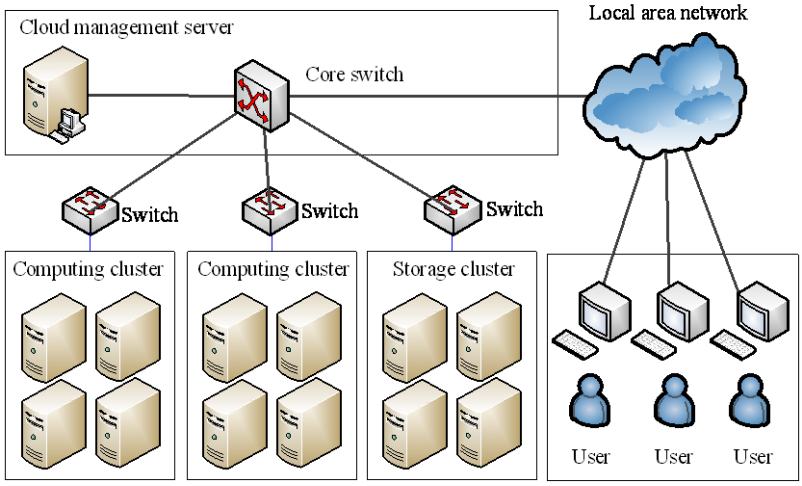

Figure 1. Architecture of Cloud Data Center

In order to facilitate management, servers in the cloud data center are grouped into clusters such as computing cluster and storage cluster. The computing cluster consists of a number of high-performance servers and makes up the computing resource pool. Using the virtual technology such as the virtual machine monitor, a number of virtual machine instances can be created from the computing resource pool and computing resources are provided to users in terms of virtual machine instances. The storage cluster contains many network storage devices and makes up the storage resource pool. Each cluster is connected to the cloud management server through a core switch. The cloud management server is responsible for allocating and managing the virtual computing resource and the virtual storage resource, such as creating virtual machine instances, deleting virtual machine instances, migrating virtual machine instances. Users can access their own virtual machine instances in the cloud data center through the Internet network and deploy their operating systems and application systems on their virtual machine instances.

For each physical node in the computing cluster, a number of virtual machine instances run on each physical node and they share all the hardware resources of the physical node. Each virtual machine instance is created according to users' requirements. If the virtual machine instance cannot satisfy performance requirements of current application systems, the cloud computing service providers will allocate more hardware resources to the virtual machine instance according to users' requirements. If the resources of the virtual machine instance are free most of the time, the cloud computing service providers could reduce hardware resources of the virtual machine instance according to users' requirements. Users deploy application systems on their virtual machine instances and virtual machine instances running on the virtual machine monitor are independent from each other.

3.2. Architecture of Proposed System

As shown in Figure 2, the proposed virtual machine instance anomaly detection system for IAAS cloud computing is divided into four modules, which are the data collection, the data transmission, the data storage, and the anomaly detection.

Copyright ⓒ 2016 SERSC

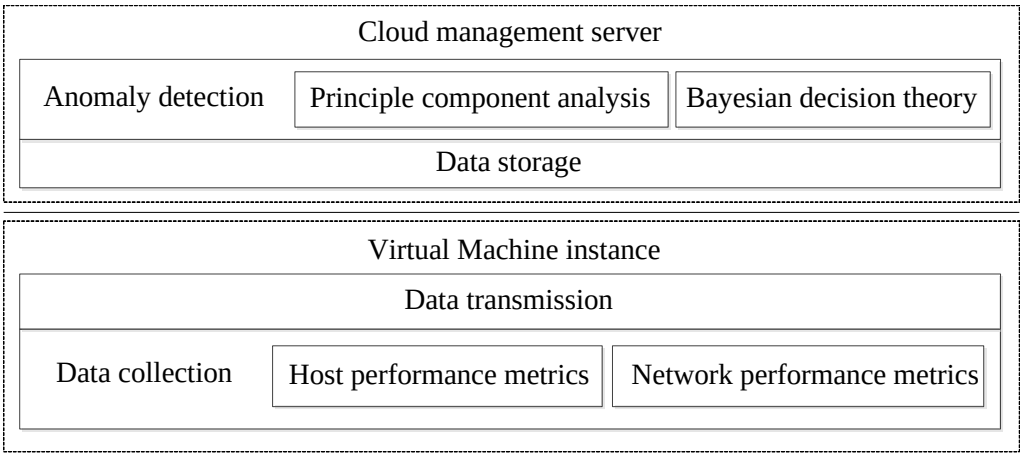

Figure 2. Architecture of Proposed System

(1) Data collection

The collected data should be able to reflect current status of virtual machine instances. The virtual machine instance owns many performance metrics that are related with the health status of the virtual machine instance. All the performance metrics can be divided into two main types, which are host performance metrics of the virtual machine instance and network performance metrics of the virtual machine instance. The host performance metrics show the host running status of a virtual machine instance and the host performance metrics consist of CPU, main memory, hard disk I/O, process, and so on. The host performance metrics can determine whether the virtual machine instance suffers from malicious software, while the network performance metrics can determine whether the virtual machine instance suffers from malicious network attacks.

For host performance metrics of each virtual machine instance, the data of each guest operating system in the virtual machine instance and the data of the running virtual machine instance should be collected. The data of each guest operating system can be collected by reading corresponding files or executing related commands. The Linux kernel provides a special /proc virtual file system, which contains runtime system information. For example, the CPU information can be obtained from the /proc/cpuinfo directory through commands such as vmstat, uptime, and top. The data of the running virtual machine instance can be collected by using the xm virtual machine management tool in the virtual machine monitor and executing related commands.

For network performance metrics of each virtual machine instance, the data of each guest operating system in the virtual machine instance and the data of each virtual machine instance should be collected. This paper collects the network data of each guest operating system through the command tcpdump and saves the collected network data into files for analysis. Because each virtual machine instance running on the virtual machine monitor shares the same network resource, the network data of each virtual machine instance could only be collected from the virtual machine monitor. However, each virtual machine instance running on the virtual machine monitor shares the same physical IP address. Therefore, it is difficult to determine that network packets that flow through the physical network card belong to which virtual machine instance. The virtual network structure of the Xen virtual machine monitor is shown in Figure 3.

 Copyright ⓒ 2016 SERSC

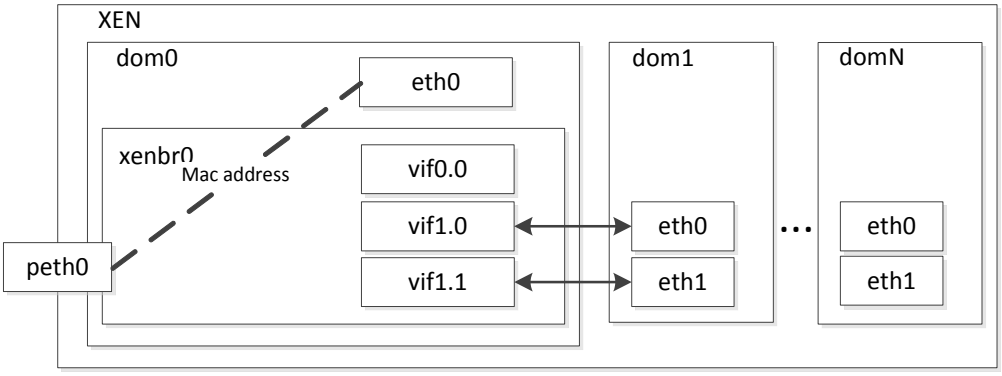

Figure 3. The Virtual Network Structure of Xen

In the virtual network structure of Xen, the peth0 is a physical network card installed in the physical node [17]. The eth0 and eth1 are virtual network devices in each virtual machine instance. The xenbr0 is the network bridge of the virtual machine instance dom0. The virtual network interface vifm.n connects to the nth virtual network device in the mth virtual machine instance. For example, the virtual network interface vif1.1 connects to the virtual network device eth1 in the virtual machine instance dom1. All the network packets have to be sent to the virtual network bridge xenbr0 through the physical network card peth0. Using destination addresses of network packets, network packets are forwarded to each virtual machine instance through corresponding virtual network interfaces. In this work, the network data are collected as follows.

The MAC address is extracted from a network packet. Using the MAC address, the virtual network interface can be obtained. Finally, the domain ID of the virtual machine instance running on the Xen can be obtained. Using this process, the network data of each virtual machine instance can be obtained accurately.

(2) Data transmission

The data transmission module forwards the collected data from the data collection module to the data storage module that is deployed in the cloud management server. In order to reduce the resource consumption of virtual machine instances due to network connections, the UDP network transmission protocol is adopted to transmit the collected data. The UDP network transmission protocol is a kind of connectionless protocol and it could reduce the performance degradation significantly [18].

The network packet structure is defined as shown in Figure 4.

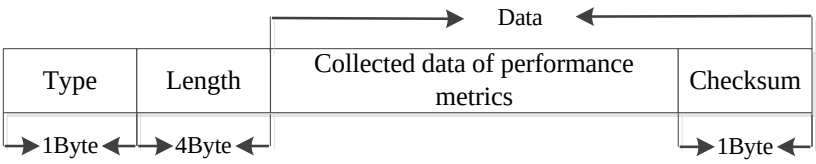

Figure 4. The Network Packet Structure

The type field shows the type of the network packet and the length field represents the length of the data field. The collected data of performance metrics are stored in the third field. The last byte of the third field contains checksum that is used to verify the validation of the network packet.

The receiving process of network packets is shown in Figure 5.

Copyright ⓒ 2016 SERSC

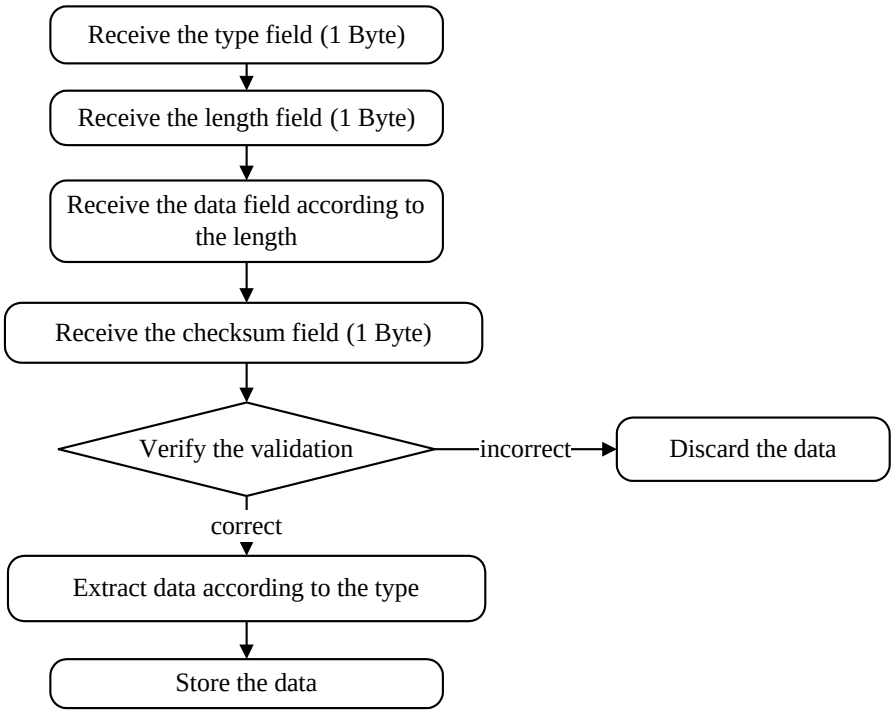

Figure 5. The Receiving Process of Network Packets

The type field and the length field are received according to the occupied bytes. Then, the data field is received according to the value of the length field. Finally, the validation of the received network packet is verified. If it is incorrect, the packet will be discarded. If it is correct, the collected data within the data field will be extracted according to the data type and stored to the local disk.

(3) Data storage

The data storage module saves the collected data of performance metrics into the local disk of the virtual machine instance.

(4) Anomaly detection

The anomaly detection module is a core component of our proposed system. It uses the principle components analysis and Bayesian decision theory to process and analyze the collected data.

4. Detailed Implementation of Anomaly Detection Module

In order to reduce the dimensionality of collected data and the computing complexity, the anomaly detection module first introduces the principle components analysis to preprocess the collected data [19]. Then, the Bayesian decision theory is introduced to analyze the preprocessed data and detect the abnormal data [20].

4.1. Principle Components Analysis

Assume that the collected data of performance metrics are represented as a matrix $X_{m\times n}$, where m is the number of data samples and n is the number of performance metrics of virtual machine instances. The feature extraction process is presented as follows.

(1) The collected data in the matrix $X_{m\times n}$ are normalized and a new matrix $Z_{m\times n}$ is obtained.

 Copyright ⓒ 2016 SERSC

$$z_{ij} = \frac{x_{ij} - \overline{x_j}}{\sqrt{\text{var}(x_j)}} (i = 1, 2, \cdots, m; j = 1, 2, \cdots, n) \quad (1)$$

where $\overline{x_j}$ and $\sqrt{\text{var}(x_j)}$ are the mean value and the standard deviation of the *j*th performance metric. They are calculated as

$$\overline{x_j} = \frac{1}{m} \sum_{i=1}^{m} x_{ij} (j = 1, 2, \cdots, n) \quad (2)$$

$$\text{var}(x_j) = \frac{1}{m} \sum_{i=1}^{m} (x_{ij} - \overline{x}_j)^2 (j = 1, 2, \cdots, n) \quad (3)$$

(2) The covariance matrix *R* of the new matrix *Z* is calculated as

$$R = \frac{ZZ^T}{m-1} \quad (4)$$

(3) Eigenvalues belonging to the covariance matrix *R* can be obtained by solving the following equation:

$$Rx = \lambda x \quad (5)$$

The obtained eigenvalues are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_c$ and their corresponding eigenvectors are $P_1, P_2, \cdots, P_c$.

(4) Principle components could be obtained by solving the following equation

$$Y_i = XP_i, i = 1, 2, \cdots, c \quad (6)$$

where Y_i is the *i*th principle component.

(5) *c* principle components could be obtained and the first *k* principle components should be selected according to their cumulative contribution rates. The cumulative contribution rate of each principle component could be calculated as

$$count(i) = \frac{\lambda_i}{\sum_{i=1}^{c} \lambda_i} \times 100\% \quad (7)$$

where *count*(*i*) is the cumulative contribution rate of the *i*th principle component. The larger the value is, the more important of the principle component is.

The sum of the cumulative contribution rates of first *k* principle components is calculated as

$$\sum_{i=0}^{k} count(i) = \sum_{i=0}^{k} \frac{\lambda_i}{\sum_{j=1}^{c} \lambda_j} \times 100\% \quad (8)$$

The processed data could retain the most important information from the collected data only when the cumulative contribution rate sum is equal to or larger than 95% and then the value of *k* can be obtained.

4.2. Bayesian Decision Theory

Assume the processed data at some point in time are denoted by a vector *X*. This work only needs to detect the normal data and the abnormal data, so there are only two data classes, which are the normal class denoted by C_n and the abnormal class denoted by C_a. The Bayesian decision theory assumes that the class-conditional probability

Copyright ⓒ 2016 SERSC

density function of the vector *X* and the prior probability $P(C)$ are known. The Bayesian decision theory determines whether the collected data are normal or abnormal according to the following decision rule:

$$\begin{cases} P(C_n \mid X) > P(C_a \mid X), \textit{ Normal} \\ P(C_n \mid X) < P(C_a \mid X), \textit{ Abnormal} \\ P(C_n \mid X) = P(C_a \mid X), \textit{ Randomly decided} \end{cases} \quad (9)$$

Let $P(X \mid C)$ be the class-conditional probability density function of *X* given a state of nature *C* and $P(C)$ be the prior probability of the status of nature *C*. Then, the posterior probability $P(C \mid X)$ can be calculated by using the following Bayesian formula:

$$P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)} \quad (10)$$

The equation (9) can be transformed to

$$\begin{cases} \dfrac{P(X \mid C_n)P(C_n)}{P(X)} > \dfrac{P(X \mid C_a)P(C_a)}{P(X)}, \textit{ Normal} \\ \dfrac{P(X \mid C_n)P(C_n)}{P(X)} < \dfrac{P(X \mid C_a)P(C_a)}{P(X)}, \textit{ Abnormal} \\ \dfrac{P(X \mid C_n)P(C_n)}{P(X)} = \dfrac{P(X \mid C_a)P(C_a)}{P(X)}, \textit{ Randomly decided} \end{cases} \quad (11)$$

so,

$$\begin{cases} \dfrac{P(X \mid C_n)P(C_n)}{P(X \mid C_a)P(C_a)} > 1, \textit{ Normal} \\ \dfrac{P(X \mid C_n)P(C_n)}{P(X \mid C_a)P(C_a)} < 1, \textit{ Abnormal} \\ \dfrac{P(X \mid C_n)P(C_n)}{P(X \mid C_a)P(C_a)} = 1, \textit{ Randomly decided} \end{cases} \quad (12)$$

The equation (12) shows that the decision rule depends on the class-conditional probability density function of the vector *X*. The term $P(X \mid C_n)P(C_n)$ and $P(X \mid C_a)P(C_a)$ are considered as likelihood functions. Let $P(X \mid C_n)P(C_n) / P(X \mid C_a)P(C_a)$ be the likelihood ratio that is denoted by $LR(X)$.

Then the equation (12) is transformed to

$$\begin{cases} LR(X) > 1, \textit{ Normal} \\ LR(X) < 1, \textit{ Abnormal} \\ LR(X) = 1, \textit{ Randomly decided} \end{cases} \quad (13)$$

The collected data of each performance metric is analyzed and it is found that the collected data of each performance metric satisfies the Gaussian distribution. Therefore,

 Copyright ⓒ 2016 SERSC

the vector X satisfies the Gaussian distribution and its class-conditional probability density function is calculated as

$$P(X \mid C) = \frac{1}{(2\pi)^{n/2} |K|^{1/2}} \exp\left[\frac{-(X-M)^T K^{-1} (X-M)}{2} \right] \tag{14}$$

so,

$$\begin{aligned} LR(X) &= \frac{P(X \mid C_n) P(C_n)}{P(X \mid C_a) P(C_a)} \\ &= \frac{\frac{1}{(2\pi)^{n/2} |K_n|^{1/2}} \exp\left[\frac{-(X-M_n)^T K_n^{-1} (X-M_n)}{2} \right] P(C_n)}{\frac{1}{(2\pi)^{n/2} |K_a|^{1/2}} \exp\left[\frac{-(X-M_a)^T K_a^{-1} (X-M_a)}{2} \right] P(C_a)} \\ &= \frac{|K_a|^{1/2} P(C_n)}{|K_n|^{1/2} P(C_a)} \\ &\exp\left[-\frac{1}{2} (X-M_n)^T K_n^{-1} (X-M_n) + \frac{1}{2} (X-M_a)^T K_a^{-1} (X-M_a) \right] \end{aligned} \tag{15}$$

We take the logarithm of both sides of the equation (15) and obtain the equation (16).

$$\begin{aligned} &\ln\left[LR(X) \right] \\ &= \ln\left\{ \frac{|K_a|^{1/2} P(C_n)}{|K_n|^{1/2} P(C_a)} \right\} \\ &+ \ln\left\{ \exp\left[-\frac{1}{2} (X-M_n)^T K_n^{-1} (X-M_n) + \frac{1}{2} (X-M_a)^T K_a^{-1} (X-M_a) \right] \right\} \\ &= -\frac{1}{2} (X-M_n)^T K_n^{-1} (X-M_n) + \frac{1}{2} (X-M_a)^T K_a^{-1} (X-M_a) \\ &+ \ln\left(\frac{|K_a|^{1/2}}{|K_n|^{1/2}} \right) + \ln\left[\frac{P(C_n)}{P(C_a)} \right] \end{aligned} \tag{16}$$

so, the equation (13) can be transformed to

$$\begin{cases} \ln\left[LR(X) \right] > 0, \textit{ Normal} \\ \ln\left[LR(X) \right] < 0, \textit{ Abnormal} \\ \ln\left[LR(X) \right] = 0, \textit{ Randomly decided} \end{cases} \tag{17}$$

5. Performance Evaluation

This section presents the experimental environment and experimental results to evaluate the performance of the proposed virtual machine instance anomaly detection system for IaaS cloud computing.

Copyright ⓒ 2016 SERSC

5.1. Experimental Environment

The proposed virtual machine instance anomaly detection system is deployed on a private cloud computing platform that is built by using the OpenStack open source cloud computing software [21]. The experimental environment consists of five high-performance computers. One of them is selected as the cloud management server, while the rest four computers are used to be computing nodes. The hardware configuration of cloud management server and computing node is listed in Table 1.

Table 1. Hardware Configuration of Cloud Management Server and Computing Node

CPU	Intel(R) Core(TM) i5-2400 CPU 3.10GHZ
Main memory	4G DDR3
Hard disk	SATA1TB/7200
Network card	100Mbps

Eight virtual machine instances have been created from four computing nodes and each computing node has two virtual machine instances running on it. The CentOS 6.5 open source operating system is installed in each virtual machine instance [22]. The system architecture of this experimental environment is shown in Figure 6.

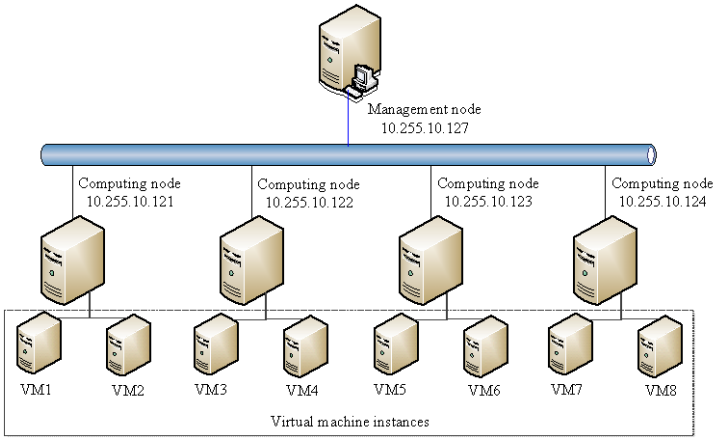

Figure 6. System Architecture of Experimental Environment

In order to make the experimental environment simulate the running status of real cloud computing environments such as the Amazon Elastic Compute Cloud, two different kinds of application systems are deployed on the experimental environment. A Hadoop-based parallel computing framework is installed on the virtual machine instances VM1, VM2, VM3, and VM4 [23]. The Hadoop-based parallel computing framework reads words from a 10 G file continuously and counts the number of each word within the file. A Web application system is deployed on the rest four virtual machine instances [24].

In order to evaluate the effectiveness of the proposed virtual machine instance anomaly detection system, four types of faults are injected into virtual machine instances to simulate abnormal behaviors.

 Copyright ⓒ 2016 SERSC

(1) Computing resource consumption. CPU resource is consumed continuously by computing the number π, which is the ratio of a circle's circumference to its diameter.

(2) Main memory resource consumption. Programs running on the virtual machine instances continuously call the Malloc() function to apply for dynamic memory and never release the allocated memory. In this case, it will result in memory leak and consuming the main memory resource continuously.

(3) Network resource consumption. The LoadRunner software runs on virtual machine instances and accesses the Web application server concurrently. In this case, a large number of http connections will be generated to simulate abnormal network behaviors.

(4) Storage resource consumption. Reading big files from the hard disk continuously incurs a large number of hard disk I/O operations and simulates abnormal storage access behaviors.

5.2. Experimental Results

For experiments, 2500 data records are collected when virtual machine instances work normally and the first 2000 data records are selected to be the training sample. 500 data records are collected when each kind of fault is injected into virtual machine instances. There are four kinds of faults, so 2500 abnormal data records will be obtained. Each kind of abnormal data records and the rest 500 normal data records make up a testing sample, so there will be four testing samples.

Performance metrics used in our experiments are the false positive rate and the false negative rate. Our proposed virtual machine instance anomaly detection system is compared with the Entropy-based Anomaly Testing (EbAT), which was proposed for online anomaly detection in cloud computing platforms.

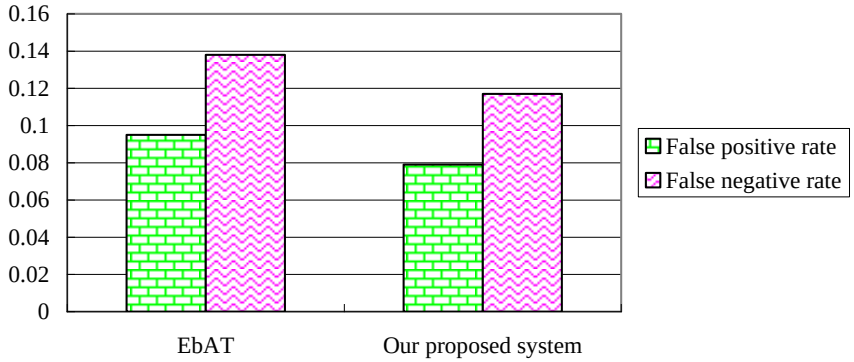

Figure 7. Experimental Results When the Type 1 Fault Is Injected

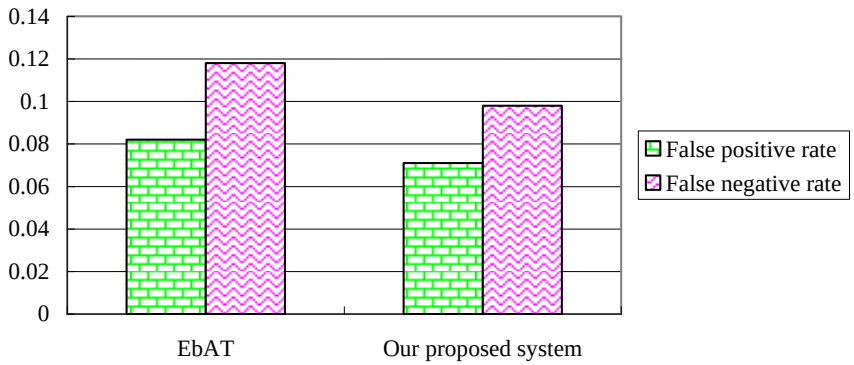

Figure 8. Experimental Results when the Type 2 Fault Is Injected

Copyright ⓒ 2016 SERSC

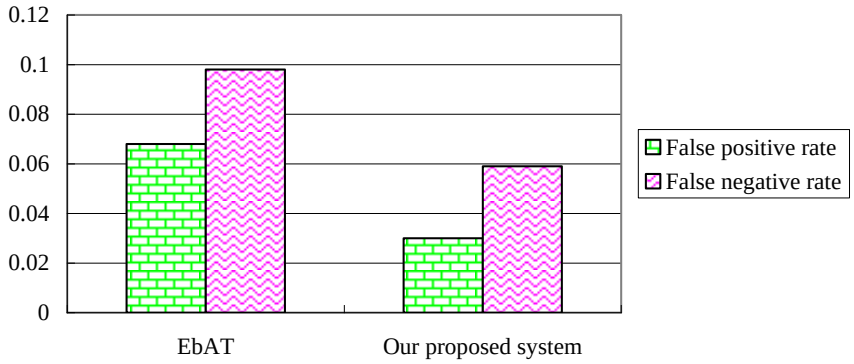

Figure 9. Experimental Results when the Type 3 Fault is Injected

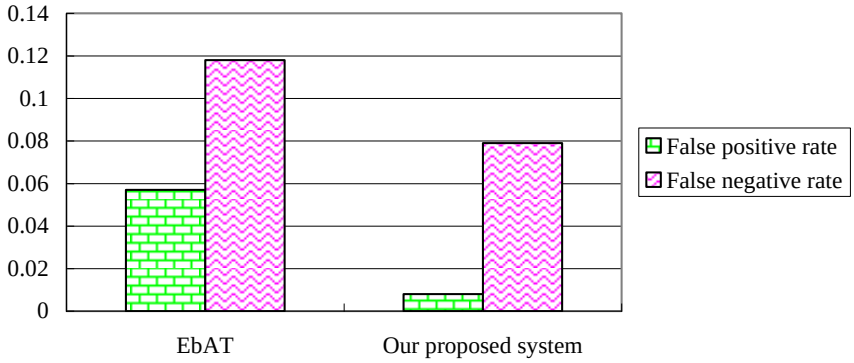

Figure 10. Experimental Results when the Type 4 Fault is Injected

Figure 7-10 show the false positive rates and false negative rates of two tested anomaly detection system. Experimental results show that our proposed virtual machine instance anomaly detection system has lower false positive rate and false negative rate than the EbAT.

6. Conclusions

An efficient virtual machine instance anomaly detection system is proposed for IaaS cloud computing in this paper. The proposed virtual machine instance anomaly detection system consists of four modules, which are the data collection, the data transmission, the data storage, and the anomaly detection. Our proposed virtual machine instance anomaly detection system first uses the principle components analysis to extract the main information from the collected data and then introduces the Bayesian decision theory to analyze the processed data and detect the abnormal data. A series of experiments are conducted on a private cloud computing platform that is built by using the OpenStack open source cloud computing software and experimental results show that our proposed virtual machine instance anomaly detection system performs better than the Entropy-based Anomaly Testing (EbAT) in terms of false positive rate and false negative rate.

Acknowledgments

The work of this paper is supported by the National Natural Science Foundation of China under Grant No. 61502102, No. 61370078, and No. 61402109, Fujian Province Education Scientific Research Projects for Young and Middle-aged Teachers under

 Copyright ⓒ 2016 SERSC

Grant No. JA15122, National Undergraduate Training Programs for Innovation and Entrepreneurship under Grant No. 201510394021, and Fujian Normal University Undergraduate Training Programs for Innovation and Entrepreneurship under Grant No. cxxl-2015163.

References

[1] Y. Xiao, G. Xu, Y. Liu, and B. Wang, "A metadata-driven cloud computing application virtualization model", Journal of Computers (Finland), vol. 8, no. 6, **(2013)**, pp. 1571-1579.
[2] G. Tajadod, L. Batten and K. Govinda, "Microsoft and Amazon: A comparison of approaches to cloud security", Proceedings of 2012 4th IEEE International Conference on Cloud Computing Technology and Science, **(2012)**, pp. 539-544.
[3] K. Lee, C. Park and H.-D. Yang, "Development of service verification methodology based on cloud computing interoperability standard", International Journal of Smart Home, vol. 7, no. 5, **(2013)**, pp. 57-66.
[4] F. Koch, M. Assuncao, D. Marcos and M.A.S. Netto, "A cost analysis of cloud computing for education", Lecture Notes in Computer Science, **(2012)**, vol. 7714, pp. 182-196.
[5] W. Liu, "Research on cloud computing security problem and strategy", Proceedings of 2012 2nd International Conference on Consumer Electronics, Communications and Networks, **(2012)**, pp. 1216-1219.
[6] D. Zissis and D. Lekkas, "Addressing cloud computing security issues", Future Generation Computer Systems, **(2012)**, vol. 28, no. 3, pp. 583-592.
[7] M. D. Ryan, "Cloud computing security: The scientific challenge, and a survey of solutions", Journal of Systems and Software, vol. 86, no. 9, **(2013)**, pp. 2263-2268.
[8] I. Son and D. Lee, "Assessing a new IT service model: Cloud computing", 15th Pacific Asia Conference on Information Systems: Quality Research in Pacific, **(2011)**.
[9] P. Garraghan, P. Townend, and J. Xu, "An empirical failure-analysis of a large-scale cloud computing", Proceedings of 2014 IEEE 15th International Symposium on High-Assurance Systems, **(2014)**, pp. 113-120.
[10] K. R. Joshi, G. Bunker, F. Jahanian, A. Van Moorsel, and J. Weinman, "Dependability in the cloud: Challenges and opportunities", Proceedings of the International Conference on Dependable Systems and Networks, **(2009)**, pp. 103-104.
[11] D. Smith, Q. Guan, and S. Fu, "An anomaly detection framework for autonomic management of compute cloud systems", Proceedings of 34th Annual IEEE International Computer Software and Applications Conference Workshops, **(2010)**, pp. 376-381.
[12] C. Wang, V. Talwar, K. Schwan, and P. Ranganathan, "Online detection of utility cloud anomalies using metric distributions", 2010 IEEE/IFIP Network Operations and Management Symposium, **(2010)**, pp. 96-103.
[13] A. Li, L. Gu, and K. Xu, "Fast anomaly detection for large data centers", 53rd IEEE Global Communications Conference, **(2010)**.
[14] Y. Han, "IAAS cloud computing services for libraries: Cloud storage and virtual machines", OCLC Systems and Services, vol. 29, no. 2, **(2013)**, pp. 87-100.
[15] R. Prodan and M. Sperk, "Scientific computing with Google App Engine", Future Generation Computer Systems, vol. 29, no. 7, **(2013)**, pp. 1851-1859.
[16] B. Jia, T. W. Wlodarczyk and C. Rong, "Performance considerations of data acquisition in Hadoop system", Proceedings of 2nd IEEE International Conference on Cloud Computing Technology and Science, **(2010)**, pp. 545-549.
[17] H. Wu, W. Li, C. Winer and L. Yao, "Research on the security of virtual network with Xen platform", Information Technology Journal, vol. 12, no. 23, **(2013)**, pp. 7774-7777.
[18] J. He and S.-H. Gary Chan, "TCP and UDP performance for Internet over optical packet-switched networks", 2003 International Conference on Communications, vol. 2, **(2003)**, pp. 1350-1354.
[19] H. Men, P. Zhang, C. Zhang, R. Wen and Z. Ge, "An electronic tongue system for recognition of mineral water based on principle component analysis and wavelet neural network", Journal of Computers, vol. 6, no. 12, **(2011)**, pp. 2692-2699.
[20] C.-C. Hsu, K.-S. Wang and S.-H. Chang, "Bayesian decision theory for support vector machines: Imbalance measurement and feature optimization", Expert Systems with Applications, vol. 38, no. 5, **(2011)**, pp. 4698-4704.
[21] A. Corradi, M. Fanelli and L. Foschini, "VM consolidation: A real case based on OpenStack cloud", Future Generation Computer Systems, vol. 32, no. 1, **(2014)**,pp. 118-127.
[22] N. Mirajkar, M. Barde, H. Kamble, R. Athale and K. Singh, "Implementation of private cloud using eucalyptus and an open source operating system", International Journal of Computer Science Issues, vol. 9, no. 3, **(2012)**,pp. 360-364.
[23] G. Xu, F. Xu and H. Ma, "Deploying and researching Hadoop in virtual machines", 2012 IEEE International Conference on Automation and Logistics, **(2012)**, pp. 395-399.

Copyright ⓒ 2016 SERSC

[24] E. Cecchet, J. Marguerite, and W. Zwaenepoel, "Performance and scalability of EJB applications", Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications, **(2002)**, pp. 246-261.

Authors

Mingwei Lin, He received his B.S. and Ph.D. degrees from Chongqing University, China, in July 2009 and December 2014. Currently, he is a lecturer in the Faculty of Software, Fujian Normal University, China. His research interests include anomaly detection, NAND flash memory, Linux operating system, and cloud computing. He got the CSC-IBM Chinese Excellent Student Scholarship in 2012.

Zhiqiang Yao, He received the PhD degree from Xidian University, China, in 2014. Currently, he is a professor in the Faculty of Software, Fujian Normal University, ACM Professional Membership, Senior Member of China Computer Federation (CCF). His current research interests mainly focus on security in cloud computing, multimedia security.

Fei Gao, He is a junior student in the Faculty of Software, Fujian Normal University, Fuzhou, China. He majors in Software Engineering. He has applied for a National Undergraduate Training Program for Innovation and Entrepreneurship successfully. His current research interests include cloud computing, android application development, and flash memory.

Yang Lao, She is a junior student in the Faculty of Software, Fujian Normal University, Fuzhou, China. Her major is Software Engineering. She is hosting a Fujian Normal University Undergraduate Training Program for Innovation and Entrepreneurship. Her current research interests include anomaly detection, cloud computing, and flash memory.

 Copyright ⓒ 2016 SERSC