Hindawi
Journal of Electrical and Computer Engineering
Volume 2018, Article ID 8309450, 9 pages
https://doi.org/10.1155/2018/8309450

Research Article

A Security Monitoring Method Based on Autonomic Computing for the Cloud Platform

Jingjie Zhang, Qingtao Wu, Ruijuan Zheng, Junlong Zhu, Mingchuan Zhang, and Ruoshui Liu

Information Engineering College, Henan University of Science and Technology, Luoyang 471023, China

Correspondence should be addressed to Qingtao Wu; wqt8921@haust.edu.cn

Received 16 November 2017; Accepted 5 February 2018; Published 5 March 2018

Academic Editor: Vincent C. Emeakaroha

With the continuous development of cloud computing, cloud security has become one of the most important issues in cloud computing. For example, data stored in the cloud platform may be attacked, and its security is difficult to be guaranteed. Therefore, we must attach weight to the issue of how to protect the data stored in the cloud. To protect data, data monitoring is a necessary process. Based on autonomic computing, we develop a cloud data monitoring system on the cloud platform, monitoring whether the data is abnormal in the cycle and analyzing the security of the data according to the monitored results. In this paper, the feasibility of the scheme can be verified through simulation. The results show that the proposed method can adapt to the dynamic change of cloud platform load, and it can also accurately evaluate the degree of abnormal data. Meanwhile, by adjusting monitoring frequency automatically, it improves the accuracy and timeliness of monitoring. Furthermore, it can reduce the monitoring cost of the system in normal operation process.

1. Introduction

Resource monitoring in cloud computing environment is an important part of resource management of cloud computing platform. It provides the basis for resource allocation, task scheduling, and load balancing. With the extensive use of cloud computing services, users have made increasing demands on the security of cloud computing. Since the cloud computing environment has the characteristics of transparent virtualization and resource flexibility, it is infeasible for a traditional security program to protect the data security in the cloud platform, which hinders further development and application of cloud computing [1]. Therefore, it is of critical importance to develop new tools suitable for monitoring cloud platform data. However, the collection, transmission, storage, and analysis of a large number of monitored data will bring huge resource overhead, directly affecting system performance, timely detection of anomalies, and pinpoint accuracy of problem. In addition, because cloud computing is essentially developed on the basis of current technology, the existing security vulnerabilities will be inherited directly to the cloud computing platform, which may even bring greater security threat. It can be seen that, in the cloud computing environment, users basically lost the control of private information and data, which triggered a series of security challenges, such as cloud data storage location, data encryption mechanism, data recovery mechanism, integrity protection, third-party supervision and auditing, virtual machine security, and memory security. At present, there is not enough research on cloud computing resource monitoring, but there are a lot of researches on distributed computing and grid computing, for instance, DRMonitor [2], Ganglia [3], and MDS (Monitoring and Discovery System) [4]. They play important roles in distributed systems or grid systems. However, if the above methods are applied directly in the cloud computing environment, there will be some shortcomings. On the one hand, the resource in the cloud computing environment is highly virtualized and flexible. Moreover, cloud computing provides services such as IaaS, PaaS, and SaaS, in addition to monitoring the resources of the physical server [5]. Users need to monitor the virtual machine running on it. On the other hand, cloud computing is a business model, and the cloud service provider will charge the user for usage accordingly. Monitoring information in

existing resource monitoring system is not fine granularity, so it is unable to get to the process level of information and track consumption of CPU, memory, storage and other resources in real time during the user task execution process. Cloud computing environment is dynamic, random, complex, and open. Cloud providers need to collect user-related fees based on resource usage; as a result, original resource monitoring methods cannot fully meet the requirements of the cloud computing environment. Therefore, according to the characteristics of cloud computing itself, some resource monitoring methods for current distributed computing, and grid computing, cannot fully adapt to the cloud computing environment.

In order to adapt to the cloud computing environment, combining with abnormal data mining algorithm, we propose a data monitoring method under cloud environment based on autonomic computing model. In order to address the security challenges for data on the cloud platform, the model uses autonomic computing mechanism and the abnormal data mining idea to transmit the monitoring information to each other. The model is mainly composed of five modules: network monitoring module, data analysis module, response strategy module, system implementation module, and knowledge base. In the network monitoring module, the system gathers the data by collecting the data stream and generates the original data. In addition, through the data preprocessing mechanism, the original data are formatted. The data analysis module evaluates these processed data, extracts useful data from it to determine whether they are abnormal, and then feeds the analysis result back to the response strategy module to adjust the monitoring period. The data collection and analysis of storage are the core parts of this model, which provide users with essential data monitoring information. In the local computer deployment monitoring framework, the cloud is connected to data monitoring. Our contributions are as follows:

(i) We propose a safe and effective model that enables the data on the cloud platform to be monitored in time, and the system adjusts the monitoring cycle to autonomously protect the data.

(ii) We design a data mining algorithm, in which, based on an improved chaotic algorithm, data mining method was proposed for the frequently appearing abnormal data in the cloud computing environment. We also design and implement abnormal behavior detection based on the Poisson process to obtain accurate test results.

(iii) We formally analyze the capability of abnormal behavior monitoring and implement all of these data security monitoring models based on autonomic computing. A large number of experiments are carried out in the simulation environment using prepared dataset, and the results show that our system achieves the desired goals.

This paper is organized as follows. Section 2 states the origin of autonomic computing theory, and its related work. Section 3 analyzes how to establish a security monitoring model based on autonomic computing for the cloud platform. Then we analyze the existing security model and safety monitoring method of autonomic computing to the cloud platform oriented metrics and the calculation method. Section 4 analyzes method of simulation and experiment and presents a security monitoring model based on autonomic computing analysis for the cloud platform. Section 5 gives the summary and points out the future research directions.

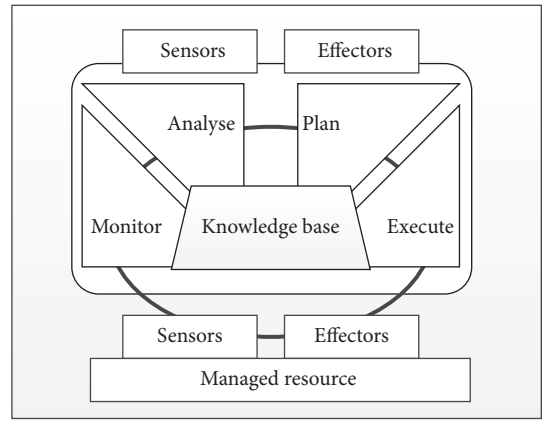

Figure 1: MAPE autonomic computing model diagram.

2. Related Work

The concept of autonomic computing was proposed by IBM's Paul Horn in 2001, which has self-configuring, self-optimization, self-healing, self- protection, and other good features that have been accepted by the computer scientists [6, 7]. Autonomous computing refers to the computing environment with self-management capabilities, dynamically adapting to the increasingly complex environment, and self-discipline since calculation unit (Autonomic Computing Element) is an essential part of the autonomic computing system [8]. At present, the research on autonomic computing is based on a model of control loop proposed by IBM in 2003. It is called the MAPE-K (Monitor, Analyze, Plan, Execute, and Knowledge) cycle. Based on this model, a strategy module has been added to allow IT managers to facilitate the management of autonomic computing units. Its structure is shown in Figure 1. The monitor assembly means collecting information from the managed resource, that is, from the external environment. The analyze assembly is used to analyze the complexity of the internal environment of the system, so that the self-regulatory manager can understand the running state of the system in real time so as to predict the future situation and take right strategy for the future condition. The plan generates a sequence of actions that can be achieved based on the monitoring component and the analysis component's data information from the external and internal environments, as well as previous policies. The execute is given to the effector to adjust the state of the managed resource. The actions generated by the above four components are all based on the knowledge in the knowledge base.

An event classification method used in the fault monitoring of autonomic computing system was introduced by Liu and Zhou in 2010 [9]. In this scheme, the system monitors the status of heterogeneous resources failure. With the self-management system communicating internally, an appropriate strategy is activated to repair the fault for self-repair system. However, there is a lack of research in the field of self-discipline for system performance failures. On the basis of studying the self-monitoring of self-discipline, the team proposed a multipoint detection method in 2011 [10]. Detecting the threshold cross-border and recovery to determine the system performance failure ensures the effectiveness of detection for the system, providing a useful strategy for the repairment failure. On the basis of studying the existing autonomic computing model, they proposed a self-discipline model which is suitable for distributed environment and formalizes the model elements of management resources, resource operation, state and action, and so on, to enhance the application of self-discipline model and practical value [11].

Among these advances of distributed computing, one must take into account the emergence of new paradigms such as cloud computing. In 2012, Yolanda et al. proposed a monitoring tool [12]. The goal of their work is to evaluate existing monitoring tools that can be used in cloud environments and are subsequently included in the monitoring component of the projects.

In order to improve the efficiency of resource management in the cloud environment, Liu and Li proposed a self-regulatory model for the cloud environment [13], which uses the multi-autonomous manager's hierarchical management model to solve the traditional autonomic computing model in dealing with a large number of requests when there is a bottleneck. Moreover, they proposed a hybrid strategy management model to tackle different fault repair requests.

Monitoring is an important factor in improving the quality of service provided in cloud computing. With the increase of cloud system architecture, the workload of data center is also increasing, leading to node failure and performance problems. Suciu et al. proposed a solution for monitoring and service providing cloud computing systems that allows users and providers to optimize the usage of the computational resources according to the constantly changing business requirements inside an organization [14]. The main contribution of their paper consists of the integration of the monitoring system, which is based on Nagios and NConf with test cloud architecture.

Liu and Guo proposed a hardware monitoring system based on cloud platform for data acquisition, storage, and analysis based on cloud computing [15]. Computer hardware monitoring system stores and accesses data based on the cloud platform. Storage hardware in the cloud platform unifies modeling analysis parameters for a large number of data in order to provide users with complete hardware maintenance information.

In 2015, Masmoudi et al. [16] proposed a multitenant monitoring approach based services that keep monitoring service execution at runtime and detecting privacy violations.

In 2016, Peng et al. [17] proposed a sampling method in the compressed sensing theory to implement feature compression for network data flow so that we can gain refined sparse representation. After that SVM (Support Vector Machine) is used to classify the compression results. This method can realize detection of network anomaly behavior quickly without reducing the classification accuracy.

3. Security Monitoring Model Based on Autonomic Computing for the Cloud Platform

With the widespread use of cloud computing services, users have made increasing demands regarding the security of cloud computing systems. The dynamicity, randomness, complexity, and openness of the cloud computing environment make it difficult to apply the conventional security scheme, which also hinders the further development and application of cloud computing [18]. The security requirements of cloud computing include terminal, platform, communication, and information security and confidentiality of the whole cycle. The cycle involves the cloud computing infrastructure layer, platform layer, application layer, and other layers.

Autonomic computing provides an effective way to reduce the complexity of the system management, but what early autonomic computing system mostly considered is physical resource management of distributed heterogeneous environment. It did not include the cloud environment under various restricting factors, such as large-scale virtualized applications, service level agreements, diverse application, and dynamic changes in the deployment environment. The original framework of autonomic computing application cannot be applied directly in the cloud environment [19]. For example, in the virtual layer, monitoring tools usually require on-demand configuration to distinguish the surveillance applications and resources. As a result, the traditional MAPE cycle also needs to be redesigned to fit the cloud. Cloud services have the characteristics of virtualization, liquidity, and boundary ambiguity, so the evaluation of their safety is one of the most important issues. According to the data security problem of the cloud platform, the working mechanism of autonomic computing system model uses the abnormal data mining idea for reference. It contains four self-regulatory elements, which can transmit the monitoring information to one another. The model framework is shown in Figure 2. The model consists of five modules: network monitoring module, data analysis module, response strategy module, system implementation module, knowledge base, and virtual machine (VM).

3.1. Model of the Cloud Security Monitoring. In order to solve the problem of cloud system security, we designed an independent monitoring model based on autonomic computing idea. The model is mainly composed of five modules: network monitoring module, data analysis module, response strategy module, system implementation module, and knowledge base.

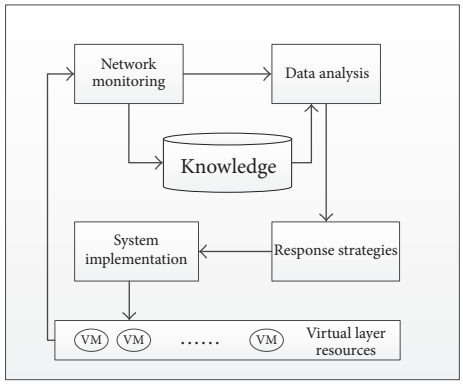

Figure 2: The cloud security monitoring model based on autonomic computing.

In the cloud computing environment, the traditional resource monitoring can be in two modes. (1) Active mode; there is the resource monitoring component in the work node and the virtual machine monitor collecting status information of the virtual machine running on it. The monitor activates the master node by sending its own monitoring information. (2) Passive mode: the master node sends request to the work node; then the work node returns their monitoring data back to the main node.

Network monitoring module is the monitoring agent deployed on the physical host, virtual machine, or other containers. It is used to collect monitored data of system at all levels and its persistent storage. Based on the historical monitored data, the data analysis module establishes the correlation of the data to form the metric correlation graph for evaluating the importance degree. It uses the PCA (Principal Component Analysis) to calculate the eigenvector of the monitored data and computes the linear regression equation of the data source in the cloud computing environment to quantitatively evaluate system anomalies. The response strategy module selects the object to be monitored in the next stage according to the importance of the monitored object. The Poisson process is used to establish the software system reliability model, and the probability of the system failure is predicted based on the abnormality degree to adjust the monitoring period. System implementation module is to use the monitoring agent to perform the dynamic adjustment of monitoring objects and monitoring cycle. Knowledge base is the record of operation in the process of learning load patterns and corresponding eigenvectors, so that the system normal operation can be depicted.

The resources of the cloud computing environment monitoring with real-time information need to adopt the strategy of polling-cyclical or event-driven way. Periodic mode refers to two cases. In the first case, the work nodes periodically send their own monitored information to the master node. In the second case, the node that is the main resource monitoring component will send a periodic request to work nodes; then these work nodes will collect and feed back information about themselves to the master node. The event-driven way refers to the fact that the old work nodes will produce a series of events, and the generation of each event is triggered by monitoring of the corresponding terminal resource state for a check and compared with the last check of the data. If the change between the two events is greater than the threshold set, then the job is in either active or passive mode.

3.2. Monitored Data Collection. In the cloud computing environment, the data collection work is a comprehensive and coordinated process and needs to coordinate with the data storage and analysis work. Then it establishes a data acquisition and analysis model, including data collection, data preprocessing, data analysis, storage, and other components.

After the data collection process, the original data stream is classified and the original data are generated. Then the data preprocessing mechanism is used to format these original data. Finally, through the data analysis and storage parts, the useful data are extracted from the massive data. By these procedures and software, users can use the data directly. This data acquisition and analysis model achieves an integrated data processing process. In this model, the data collection and analysis of storage are two core parts, giving customers meaningful information.

Data Collection. In the cloud computing environment, the data collection consists of three steps, namely, data capture, data filtering, and data classification. Through these three steps, original data flow deals with customer meaningful data [20]. After capturing the data, it needs to filter and remove some useless information and categorize the rest. Then the classification of the data is sent to the preprocessing part.

Data Preprocessing. Data preprocessing includes several methods, that is, data cleaning, data integration, data transformation, and data reduction [21]. After selecting appropriate attributes as the data mining attributes from the original data, the principle of selection process should refer to giving attribute names and attribute values as clear meaning as possible, unifying multiple data source attribute code, and removing the unique attribute. So it can select the appropriate monitoring data for analysis.

Data Analysis. Data collection model must be established with relevant functions and appropriate conditions. Firstly, the data acquisition model needs to understand the expected results. Measuring and verifying the initial model are also required. According to the validation results, the model will be effectively adjusted. Secondly, the data acquisition model needs to be practical. At last, the causal relationship between the data will be revealed in order to make the collected data statistically significant. The establishment of the data model also needs to be improved continuously.

(1) The length of the sliding window is n; multiple measures for monitored data are collected as $x = (x_1, x_2, \ldots, x_m)$, among which each collected monitoring data includes m metrics. The operation and

maintenance personnel can set m value according to their needs. Here, m is a positive integer; x_i is the value of the ith metric. The monitored data slides into the sliding window in the order of time. Monitored data in the sliding window form the matrix A_{nm} (n rows and m columns).

(2) Each column of A_{nm} is standardized with a mean of 0 and the variance of 1. $Z_i = (x_i - u_i)/\sigma_i$, u_i is the mean of the ith column data set, and σ_i is the standard deviation of the ith column data set.

(3) The covariance matrix is obtained:

$$C = \begin{bmatrix} \sigma_{11}^2 & \cdots & \sigma_{1m}^2 \\ \vdots & \ddots & \vdots \\ \sigma_{m1}^2 & \cdots & \sigma_{mm}^2 \end{bmatrix}. \tag{1}$$

(4) Calculate the eigenvector of C, the main direction of the data distribution $\overrightarrow{u}$.

(5) When new monitored data arrives, in order to enlarge the influence of outliers on the main direction of change, the sample will be copied nr times, $r \in [0, 1]$, which is a copy of the current samples with the current sample size. The proportion of the updated matrices are shown as follows:

$$A = A \cup \{x_t, x_t, \ldots, x_t\}. \tag{2}$$

(6) Update matrix mean and covariance matrix:

$u = (u + rx_t)/(1 + r)$. Update the eigenvector in the main direction:

$$u_t = \frac{\sum_A u}{\left\| \sum_A u \right\|}. \tag{3}$$

(7) Evaluate abnormality of newly collected monitored data by using cosine similarity. The lower the similarity between the two main eigenvectors, the larger the deviation and the higher the abnormality. This paper describes the degree of anomaly:

$$\text{AutoCos} = \frac{u_t \cdot u}{\|u_t\| \times \|u\|}. \tag{4}$$

In this formula, "·" represents the dot product.

Data Storage. Data storage is the most important security issue, which needs to be addressed to ensure the integrity of the data. The access to data can only be granted to those with right authority.

3.3. Monitored Data Analysis. When the traditional mining algorithm is used to excavate the frequent abnormal data, it cannot identify the abnormal data of frequent abnormalities, and there is a big problem of frequent anomaly of data mining. Based on the historical monitored data, the data analysis module establishes the correlation between the metrics to form the metric graph, so that the degree of importance of the metric can be evaluated. A mining method for anomaly data based on the improved chaos algorithm is proposed. The eigenvector of the monitored data is calculated to quantify the system anomalies. Firstly, the data source in the observed cloud computing environment is fused to the partial least squares method to be cleaned and dimensionless processed, and a normalized data matrix, as well as a normalized dimension vector, is obtained. The matrix and the vector are defined respectively. The anomalous data predicting variables and the determinants are observed, and the principal component analysis is extracted to establish the linear regression equation of the data source in the cloud computing environment. The concrete steps are as follows:

(1) We will fuse the observed data to partial least squares, so that the original data can be the cleaned with nondimension. The formula is as follows:

$$\begin{aligned} x_{ij}^* &= \frac{x_{ij} - \overline{x}_j}{\sqrt{S_j S(D)}}, \\ y_i^* &= \frac{y_i - \overline{y}}{\sqrt{S_y}} \times x_{ij}^*. \end{aligned} \tag{5}$$

Here, x_{ij} is state space of data source in the cloud computing environment with intrinsic mode function, S_j represents weight vector of each data in the cloud computing environment, S_y represents the steady-state probability of the global data in the cloud computing environment, y_i represents a multivariable time series of given exception data, $\overline{y}$ represents the local thinning ratio, x_{ij}^* represents the source data after cleaning in the cloud computing environment, and y_i^* represents dimensionless processing of data source in cloud computing environment.

(2) We calculate the standardized data matrix and the normalized dimensional vector and provide the data basis for the main component analysis in the next step. We obtain a normalized matrix of $m \times n$ order represented by X_0 and the m-dimensional vector represented by Y_0, as follows:

$$\begin{aligned} Y_0 &= y_i^* \times \begin{bmatrix} y_1^* \\ y_2^* \\ \vdots \\ y_m^* \end{bmatrix}, \\ X_0 &= \begin{pmatrix} x_{11}^* & \cdots & x_{1n}^* \\ \vdots & \ddots & \vdots \\ x_{m1}^* & \cdots & x_{mn}^* \end{pmatrix} \times Y_0. \end{aligned} \tag{6}$$

y_1^*, y_2^*, and y_m^* represent the main elements of Y_0, and x_{11}^*, x_{1n}^*, x_{m1}^*, and x_{mn}^* represent the main elements of X_0.

(3) X_0 and Y_0 are defined, respectively, as the results of observation value standardization for prediction variables and the determinant of frequent abnormal data in cloud computing environment. The principal component analysis and extraction will be carried on. The following two formulas are used, respectively, to extract principal components Y_0 and X_0 of Y_1 and X_1, making X_1 maximally reflect Y_1, implementing principal component extraction. The calculation formulas are as follows:

$$Y_1 = \frac{t_1 (X_0 \omega_1)}{Y_0},$$
$$X_1 = \frac{P_1 \otimes X_0^T t_1}{Y_1}, \tag{7}$$

where P_1 is the frequent abnormal data on behalf of the cloud computing environment, X_0^T represents the eigenvalue vector data of frequent abnormalities in cloud computing environment, ω_1 is the similarity of characteristic value for frequent abnormal data, and t_1 represents the time complexity of the decomposition of matrix X_0.

(4) After extracting the principal components, we need to establish the regression equation of the source data in cloud computing environment. The expression is as follows:

$$(x^*, y^*) = X_1 \times \frac{\alpha_1 x_1 + \alpha_2 + {x_2}^* + \alpha_p + {x_p}^*}{\varpi_1}. \tag{8}$$

(x^*, y^*) is the linear regression coefficient of data in cloud computing environment, α_1, α_2, and α_p represent relevance between dependent variable and each variable of frequent abnormal data, and ${x_2}^*$ and ${x_p}^*$ represent the correlation between the frequent abnormal data variables.

3.4. Policy Response Method. The nature of the services provided by the cloud computing platform is the Internet access service. Most of the existing methods use the stochastic process model of typical telecommunication access-Poisson process as the basis of system resource scheduling optimization. Poisson process is a class of the most basic independent process in describing random events, in which the time interval of events is considered to be independent random variables, equivalent to an update process [22]. The typical Poisson process is expressed as

$$P(N(t) - N(t-s) = n) = \frac{e^{-\lambda[(t-(t-s))]} [\lambda t - \lambda (t-s)]^n}{n!}. \tag{9}$$

In (9), $N(t)$ is the number of events occurring between 0 and t, and λ is the intensity of the event. In the Internet access services, they can be seen as the number and access strength of Internet services access at time t; $N(t) - N(t-s)$ is the increment of the number of Internet access services in the time $[t-s, t)$.

When the data module is abnormal, it will be sent to the corresponding module of the strategy. The Poisson distribution process is the classical fault prediction model of the reliability engineering. Conventionally, the historical fault data is used to predict the time of the next failure. However, this paper improves the model and introduces the anomaly evaluation to replace the historical fault data and predict the next system failure time. In this way, by assessing the degree of system anomalies, the instantaneous error can be responded to in a timely manner; the monitoring cycle is immediately adjusted to the minimum. The cumulative error can be adjusted according to the degree of abnormal monitoring cycle. We set the probability of a system failure as $F(t) = w$. Then you can calculate the time interval of next failure:

$t = -\ln(1-w)/\lambda$, so the current monitoring period can be adjusted to

$$T = \begin{cases} T_\beta, & 0 \le s_t < \beta \\ -\dfrac{\ln(1-s_t)}{\lambda} + e, & \beta \le s_t \le \alpha \\ T_\alpha, & \alpha < s_t \le 1. \end{cases} \tag{10}$$

In (10), T_β is the minimum monitoring period, T_α is the maximum monitoring period, and λ and e are the adjustment parameters. According to the analysis of the function, we find that, if the monitoring cycle is set between the maximum and the minimum monitoring cycle, abnormal monitoring cycle shortens with the increase in the degree of abnormal system, and the shortness degree increases with the range of increase degree of abnormal monitoring cycle. That is to say, the more serious the abnormality is, the shorter the monitoring period is, which the desired result is.

4. Experiments and Simulation

In order to further verify the feasibility and rationality of the autonomous monitoring model based on autonomic computing, a simulation experiment is carried out. MATLAB software is used to build the frequent abnormal data simulation platform in the cloud computing environment. The computer simulation platform configuration includes Intel (R) Core (TM) i3-2130 3.4 GHz CPU, 4.00 GB memory, windows 10 professional 64-bit operating system, MATLAB 2014b. The cloud platform is built on four different operating system application servers. The experimental environment includes Huawei S9303 multilayer routing switch and MySQL database server to provide data operation.

4.1. Experiment Procedure. In the monitoring of data collection, we monitor three kinds of resource types, shown in Table 1. In this paper, we use sliding window technology to temporarily store the monitoring data, mainly to achieve two functions. The first function is to calculate the current sliding window in the vector set of eigenvectors, compared with the reference feature vector to detect the fault. The second one is to learn the new load mode according to current sliding window in the load vector set, so that the exception

Table 1: Critical monitoring metrics.

Resource type	Interpretation	Metric
CPU	The system calls the CPU time slice	system_cpu_usage
Internet	Number of packets received	r_packets
	Number of packets sent	s_packets
Data content	Integrity	inf_com

Table 2: Frequent abnormal data mining performance.

Number of experiments (times)	Accuracy rate (%)	Error rate (%)	Reliability rate (%)
20	98.2	0.2	98.6
30	97.7	0.1	97.0
40	98.1	0.1	98.3
50	98.2	0.1	98.6
60	98.2	0.1	98.6

in the specific load mode can be detected. When the new monitored data arrives at the end of the sliding window, the larger the length of the first window of the monitoring data to delete is, the more obvious the abnormal performance will be and the better the timeliness of abnormal detection will be. This results in a higher false negative rate. On the contrary, the smaller the window length is, the more obvious the failure performance will be and the greater the impact of noise monitored data will be. The results are in a higher false alarm rate. The experimental results show that when the window length is 17–23, the detection effect is ideal with little difference. So we will set the sliding window size to 20 in this experiment.

In this experiment, the number of concurrent steps between 0 and 2 can be gradually increased to 300. When the number of concurrent steps is set to 5 minutes, and the experiment lasts 500 minutes, then 100 sets of data can be acquired. In the experiment, set the sliding window length to 20. The first 20 data sets in the main direction of the PCA (Principal Component Analysis) feature vector are the benchmark training data, and then the data can be used as an online fault detection.

Based on the improved chaos algorithm, the anomaly data mining method is used to simulate the frequent abnormal data in the cloud environment. The mining accuracy and the mining error rate are compared with different sets of experimental times so as to measure the effectiveness, referring to Table 2.

4.2. Results Analysis. When data analysis is carried out under unchanged condition of load model, the experimental results are shown in Figure 3. It shows that the abscissa is concurrent number and the coordinate is the abnormal degree (1 − |correlation|) with values between 0 and 1. The maximum of abnormal degree is 1, which means that there is a problem in cloud data. The minimum value is 0, implying that cloud

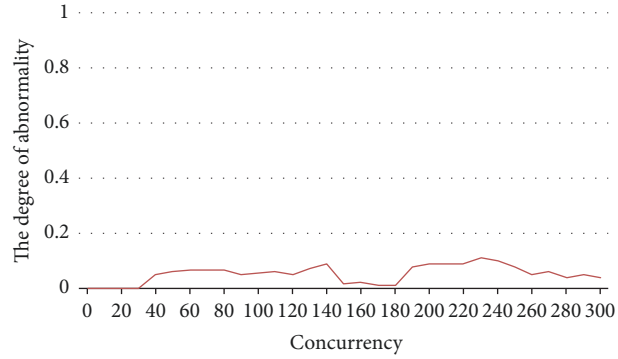

Figure 3: Abnormal degree change with concurrency.

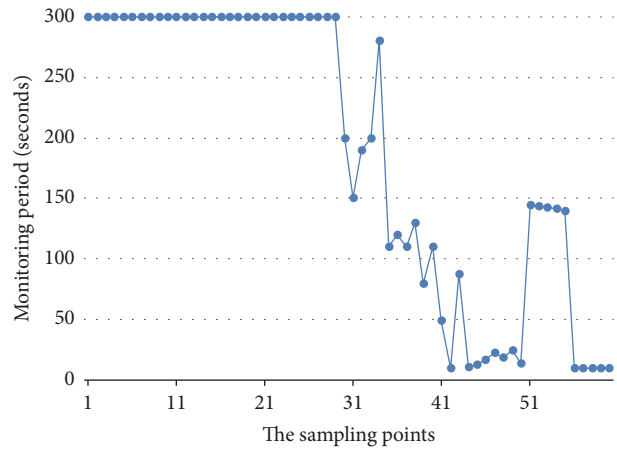

Figure 4: Monitoring period adjustment.

data is normal. In the constant load model, when there is a concurrent number, there is no phenomenon of normal monitoring data to detect anomalies mistakenly. Therefore, the number of concurrent dynamic change situations has good adaptability.

According to the experimental results in this paper, the maximum and minimum monitoring cycles are, respectively, set to 300 seconds and 10 seconds. Initial system abnormal value is 0, and the biggest monitoring cycle is 300 seconds. When the cloud data situation changes, then the monitoring period is adjusted accordingly. When the cycle of sample point is short and the monitoring cost is larger, for example, when the sample point is no more than 20, Figure 4 shows that no matter how the situation changes, monitoring time and the basic monitoring price remain unchanged.

Repeated experiments show that when the degree of network anomaly is low, the method of dynamically adjusting the monitoring period makes the monitoring system adopt a large monitoring period, reduce the number of data collection, reduce the unnecessary overhead, and dynamically adjust the monitoring period to send the network traffic below fixed detection cycle method. Therefore, the model can be adjusted according to the amount of data in cloud platform monitoring cycle, to reduce the monitoring cost.

5. Conclusion and Further Work

Data security in cloud platform is an ongoing challenging issue. We draw lessons from the working mechanism of autonomic computing system and the idea of abnormal data mining. Then we propose a data security monitoring method based on autonomic computing. This method monitors changes of data in the cloud platforms, to ensure the security of these. Simulation results show that the cost can be reduced with the architecture of integrated modules of data collection, analysis, monitoring service, and data volume based monitoring cycle adjustment. However, the ability of repairing the attacked data is not taken into account in the proposed method yet. Therefore, the research on data recovery and related implementation will be considered in the future work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants no. 61602155, no. U1604155, no. 61370221, and no. U1404611, in part by the Program for Science & Technology Innovation Talents in the University of Henan Province under Grant no. 16HASTIT035, in part by Henan Science and Technology Innovation Project under Grants no. 164200510007 and no. 174100510010, in part by the Industry university research project of Henan Province under Grant no. 172107000005, and in part by the support program for young backbone teachers in Henan Province under Grant no. 2015GGJS-047.

References

[1] G. Komal and G. Parul, *Cloud computing security issues: An analysis*, Institute of Electrical and Electronics Engineers Inc., 2016.

[2] J. J. Jiang and Z. F. Ding, "The design and implementation of cloud computing model and platform," *Applied Mechanics and Materials*, vol. 631-632, pp. 210–217, 2014.

[3] J. Kar and M. R. Mishra, "Mitigating threats and security metrics in cloud computing," *Journal of Information Processing Systems*, vol. 12, no. 2, pp. 226–233, 2016.

[4] I. Diaz, G. Fernandez, M. Martin, P. Gonzalez, and J. Tourino, "Integrating the common information model with MDS4," in *Proceedings of the 2008 9th IEEE/ACM International Conference on Grid Computing (GRID)*, pp. 298–303, Tsukuba, September 2008.

[5] Y. Q. Zhang, X. F. Wang, X. F. Liu, and L. Liu, "Survey on cloud computing security," *Journal of Software. Ruanjian Xuebao*, vol. 27, no. 6, pp. 1328–1348, 2016.

[6] T. B. Mathias and P. J. Callaghan, "Autonomic computing and IBM System z10 active resource monitoring," *IBM Journal of Research and Development*, vol. 53, no. 1, pp. 13:1–13:11, 2009.

[7] Z. Wang, H. Wang, G. Feng, and et al., "Research on Autonomic Computing System and its Key Technologies," *Computer Science*, vol. 40, no. 7, pp. 15–18, 2013.

[8] K. Ahuja and H. Dangey, "Autonomic Computing: An emerging perspective and issues," in *Proceedings of the 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques, ICICT 2014*, pp. 471–475, India, February 2014.

[9] W. Liu and Y. Zhou, "Research and Design of Fault Monitoring Mechanism Based Oil Autonomic Computing," *Computer Science*, vol. 37, no. 8, pp. 175–177, 2010.

[10] W. Liu and Z. Li, "Multiplied Data Threshold Detecting Method Based on Autonomic Computing," *Computer Science*, vol. 38, no. 5, pp. 132–134, 2011.

[11] W. Liu and K. Ren, "Research and Design of an Autonomic Computing System Model Based on Distributed Environment," *Journal of Northwestern Polytechnic University*, vol. 29, no. 2, pp. 160–164, 2011.

[12] S. Yolanda, G. Georgina, and C. Mariela, "Evaluation of monitoring tools for cloud computing environments," *IEEE Computer Society*, pp. 569–578, 2012.

[13] W. Liu and Z. Li, "Autonomic Computing Model Based on Hierarchical Management in Cloud Environment," *Computer Science*, vol. 41, no. 3, pp. 189–192, 2014.

[14] G. Suciu, S. Halunga, A. Ochian, and V. Suciu, "Network management and monitoring for cloud systems," in *Proceedings of the 6th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2014*, pp. 1–4, Romania, October 2014.

[15] M. Liu and P. Guo, "Design and Implementation of Computer Hardware Monitoring System Based on Cloud Computing," in *Proceedings of the 2016 International Conference on Electronic, Information and Computer Engineering, ICEICE 2016*, Hong Kong, April 2016.

[16] F. Masmoudi, M. Loulou, and A. H. Kacem, "Multi-tenant services monitoring for accountability in cloud computing," *IEEE Computer Society*, pp. 620–625, 2015.

[17] M. Peng, H. Xiong, and X. Yu, "SVM intrusion detection model based on compressed sampling," *Journal of Electrical & Computer Engineering*, pp. 1–7, December 2016.

[18] Z. ZEng and C. Li, "Study on frequent Abnormal Data Mining Method in Cloud Computing Environment," *Computer Simulation*, vol. 33, no. 3, pp. 339–342, 2016.

[19] J. Ge, B. Zhang, and Y. Fang, "Study on Resource Monitoring Model in Cloud Computing Environment," *Computer Engineering*, vol. 37, no. 11, pp. 31–33, 2014.

[20] D. Li, L. Li, L. Jin, G. Huang, and Q. Wu, "Research of load forecasting and elastic resources scheduling of Openstack platform based on time series," *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*, vol. 28, no. 4, pp. 560–566, 2016.

[21] H. Chen, X. Fu, Z. Tang, and X. Zhu, "Resource monitoring and prediction in cloud computing environments," in *Proceedings of the 3rd International Conference on Applied Computing and Information Technology and 2nd International Conference on Computational Science and Intelligence, ACIT-CSI 2015*, pp. 288–292, Japan, July 2015.

[22] Z. Luo, H. Qian, and Z. Jun, "Abnormal Behavior Detection Based on Poisson Equation," *Science Technology and Enginering*, vol. 14, no. 2, pp. 50–54, 2014.