ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Anomaly detection and identification scheme for VM live migration in cloud infrastructure☆

Tian Huang [a], Yongxin Zhu [a,*], Yafei Wu [a], Stéphane Bressan [b], Gillian Dobbie [c]

[a] School of Microelectronics, Shanghai Jiao Tong University, China

[b] School of Computing, National University of Singapore, Singapore

[c] Department of Computer Science, University of Auckland, New Zealand

HIGHLIGHTS

- We extend Local Outlier Factors to detect anomalies in time series and adapt to smooth environmental changes.
- We propose Dimension Reasoning LOF (DR-LOF) that can point out the "most anomalous" dimension of the performance profile. DR-LOF provides clues for administrators to pinpoint and clear the anomalies.
- We incorporate DR-LOF and Symbolic Aggregate ApproXimation (SAX) measurement as a scheme to detect and identify the anomalies in the progress of VM live migration.
- In experiments, we deploy a typical computing application on small scale clusters and evaluate our anomaly detection and identification scheme by imposing two kinds of common anomalies onto VMs.

ARTICLE INFO

Article history:
Received 3 March 2015
Received in revised form
11 May 2015
Accepted 9 June 2015
Available online xxxx

Keywords:
Anomaly detection
Security
Virtualization

ABSTRACT

Virtual machines (VM) offer simple and practical mechanisms to address many of the manageability problems of leveraging heterogeneous computing resources. VM live migration is an important feature of virtualization in cloud computing: it allows administrators to transparently tune the performance of the computing infrastructure. However, VM live migration may open the door to security threats. Classic anomaly detection schemes such as Local Outlier Factors (LOF) fail in detecting anomalies in the process of VM live migration. To tackle such critical security issues, we propose an adaptive scheme that mines data from the cloud infrastructure in order to detect abnormal statistics when VMs are migrated to new hosts. In our scheme, we extend classic Local Outlier Factors (LOF) approach by defining novel dimension reasoning (DR) rules as DR-LOF to figure out the possible sources of anomalies. We also incorporate Symbolic Aggregate ApproXimation (SAX) to enable timing information exploration that LOF ignores. In addition, we implement our scheme with an adaptive procedure to reduce chances of performance instability. Compared with LOF that fails in detecting anomalies in the process of VM live migration, our scheme is able not only to detect anomalies but also to identify their possible sources, giving cloud computing operators important clues to pinpoint and clear the anomalies. Our scheme further outperforms other classic clustering tools in WEKA (Waikato Environment for Knowledge Analysis) with higher detection rates and lower false alarm rate. Our scheme would serve as a novel anomaly detection tool to improve security framework in VM management for cloud computing.

1. Introduction

Virtualization is the fundamental technology that powers cloud computing [1] as a middleware that separates physical infrastructures to create various dedicated resources, e.g. virtual machine (VM). Researchers and engineers have developed distributed computing systems on large clusters of low-cost commodity machines running VMs [2]. Examples of such initiatives include Google's

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author.
E-mail addresses: ian_malcolm@sjtu.edu.cn (T. Huang), zhuyongxin@sjtu.edu.cn (Y. Zhu), wuyf0406@sjtu.edu.cn (Y. Wu), steph@nus.edu.sg (S. Bressan), gill@cs.auckland.ac.nz (G. Dobbie).

http://dx.doi.org/10.1016/j.future.2015.06.005
0167-739X/

MapReduce, Hadoop from the open-source community, and Cosmos and Dryad from Microsoft. Applications running in VMs provide easy deployment, flexible resource provisioning, high utilization of server resources, and simpler administration. They can take advantage of different services and capabilities provided by the virtualization infrastructure such as live migration, elastic scale out, and better sharing of physical resources [3–5].

As an underlying technique of virtualization, live migration refers to the process of moving a running VM between different physical machines without disconnecting the client or application. Live migration brings the benefits of workload balancing, elastic scaling, fault tolerance, hardware maintenance, sharing infrastructures and transparency to users. However, the dynamic nature of VMs, especially live migration, makes them difficult to maintain the consistency of security.

For example, VMs' live migration among physical servers could spread security vulnerabilities and human negligence with an ignorant and rapid way. This will be a disaster against a pool of virtualized servers for production use, because there are generally no physical firewalls separating the VMs in a virtual environment [6]. Additionally, VM live migration adds difficulties to anomaly detection because the Inconsistent performance of VM before, in the course of and after the live migration cover anomalies and attacks [7].

Anomaly (or behavioral) detection is concerned with identifying events that appear to be anomalous with respect to normal system behavior [8]. Anomaly based approach involves the collection of data relating to the behavior of legitimate users over a period of time, and then applies statistical tests to the observed behavior, which determines the legitimacy of the behavior. It has the advantage of detecting attacks which have not been found previously. Recently, anomaly detection is extensively researched for cloud computing area [6,9,10] where complete knowledge of security issues are not available because of the heterogeneity and diversity of cloud environments. However existing anomaly detection methods are not ready to cope with the challenges VM live migration brings to the security management of cloud computing [7].

Existing methods detect anomalies but do not identify the root cause of anomalies. As security issues may spread rapidly through live migration, knowing the reason of the security issues and taking action in time will delay the spreading and reduce Mean Time To Recovery (MTTR). The heterogeneity and diversity of cloud environment hinder staff in manually identifying the cause. We believe that along with generating the alarm, Anomaly detection methods ought to provide staff the clues to pinpoint and clear the anomalies.

Besides, existing methods do not distinguish anomalies from the impacts of environmental changes. When VMs are migrated to a destination host whose workload, memory size, network condition, I/O speed and many other infrastructure settings are different from the origin host, the applications provisioned by the VMs may behave differently, making detection methods difficult to distinguish the anomalies and attacks from the impacts of environmental factors.

In this paper, we mitigate these security issues by proposing an anomaly detection and identification scheme for VM live migration. Our scheme incorporate time series analysis methods and Local Outlier Factor (LOF) algorithm to detect and identify the anomalous behaviors through the performance profile of virtual machine. This scheme works based on the fact that an application keeps its behavioral consistency and consumes similar amount of resources in the progress of VM live migration. Live migration and environmental changes impose consistent impacts on the performance profile of VM. Therefore our scheme identifies the legitimacy of behaviors of VM according to the similarity of its performance profile. Our major contributions can be summarized as follows:

- We extend Local Outlier Factors (LOF) [11] to detect anomalies in time series and adapt to smooth environmental changes.
- We propose Dimension Reasoning LOF (DR-LOF) that can point out the most anomalous dimension of the performance profile. DR-LOF provides clues for administrators to pinpoint and clear the anomalies.
- We incorporate DR-LOF and Symbolic Aggregate ApproXimation (SAX) [12] measurement as a scheme to detect and identify the anomalies in the progress of VM live migration.

In experiments, we deploy a typical computing application on small scale clusters and evaluate our anomaly detection and identification scheme by imposing two kinds of common anomalies onto VMs. We compare our scheme with other methods in the scenario live migration. Experiments show that our scheme detects anomalies more accurately and effectively over VM live migration and other environmental changes.

The rest of the paper is organized as follows. In Section 2, we discuss the related work and the background classic LOF and SAX algorithms. In Section 3, we give an overview and present detailed implementation of our scheme. In Section 4, we comparatively analyze the performance of our proposed scheme in contrast with classic LOF and with various clustering algorithms. Finally, we summarize our contribution and results and conclude in Section 5.

2. Related work and background

2.1. Related work

As a classic issue, anomaly detection has been explored by researchers in the areas of intrusion detection, machine learning, web semantics as well as relevant statisticians. The problem arose again with new technical and operational features in cloud computing systems, though there have been well formed methods for operations on classic computer systems.

Many efforts on intrusion detection draw on the experience of machine learning. Classification, clustering and Markov chain [13–16] are well studied for classic datacenters, while anomaly detection methodology in cloud computing system is still in its early stage. For example, Lo and co-authors [17] presented a scheme which successfully reduced the attack of DoS (Deny of Service) by collecting and mining network packets. However, the training phase of the method settles down the dividing line between normal and abnormal data. Therefore the effectiveness highly depends on the long term stability of infrastructure settings, which is contradictory to the characteristics of cloud computing.

Among methods handling anomaly detection in cloud systems [6,9,10], there are a few efforts on security issues in VM live migration. Adamova et al. [7] stated that virtual service migration can adversely affect state of the art anomaly detection techniques, potentially rendering them unusable. Wang et al. [18] proposed a role-based mechanism with remote attestation to leverage Intel vPro and TPM to improve security. VM migration is controlled by specific policies that are protected in seal storage under the mechanism. Xianqin2009seamless and co-authors [19] built a prototype system of the framework based on stateful firewall enabling Xen hypervisor. Alshawabkeh and co-authors [20] focused on feature selection algorithm to enhance detection accuracy in VM, but it did not consider VM migration scenario. Biedermann and co-authors [21] detected live migrations from the inside of an affected VM and measured security status of the VM before the migration finished. Oberheide and co-authors [22] investigated threats to VM migration and discussed strategies to address the deficiencies in virtualization software. These works focused on generic frameworks to detect anomalies and keep VM from attacks. However, none of them gave feasible ways to detect and fix anomalies. As the virtualization technology may incur anomalies that cannot be

Please cite this article in press as: T. Huang, et al., Anomaly detection and identification scheme for VM live migration in cloud infrastructure, Future Generation Computer Systems (2015), http://dx.doi.org/10.1016/j.future.2015.06.005

identified by previous experience, the ability of identifying the root cause of anomalies becomes a must-have for security tools in cloud computing.

In our previous work [23], we proposed to implement LOF for anomaly detection in cloud computing. The concept in the work viewed data as separated points in feature space loses time information. As a result, The method is weak in understanding the relationships between two successive points and mistakes some normal behaviors for anomalies. In [24], we presented some preliminary results on anomaly detection for VM migration in cloud infrastructure. New contributions are made in this paper with reinforcement of the proof of our algorithm, sensitivity analysis of our scheme to input parameters, as well as by comparison of our scheme with major clustering algorithms.

In this paper, we take into account the changing of behaviors of VM due to the characteristics of cloud computing. We detect anomaly in the scenario of VM migration by utilizing the time information of VM behaviors. By doing this we not only improve the detecting rate but also reduce the false alarm rate. We also aim at identifying the most anomalous dimension of behaviors so that administrators can pinpoint and clear the anomalies more easily.

2.2. Background

In this section, we first describe the concept of anomalies of virtual machines in cloud infrastructure. As our algorithms are extensions and combinations of LOF and SAX, we then give a brief description of these basic algorithms.

2.2.1. Concept of anomalies of VM

According to the definition presented in [25], anomalies are patterns in data that do not conform to a defined notion of normal behavior. Our definition for anomalies of VMs follows the same idea behind the classic definition. We define the normal behaviors of a VM as patterns in observed behaviors that conform to the expectations of cloud providers and users. Observed behaviors, such as the traces of resource utilizations and system calls, are reasonable representations of the behaviors of a VM. The anomalies of VM that our scheme detects and identifies are the patterns in observed behaviors that do not conform to the normal behaviors of the VM.

Behaviors of VM can be anomalous from collective and contextual view. Collective anomaly occurs when a collection of related data instances is anomalous, whereas contextual anomaly indicates that an individual data instance is anomalous within a context. Our scheme handles two kinds of anomalies with LOF and SAX algorithms.

2.2.2. Detect anomaly with LOF and SAX

We choose LOF as our basic strategy to detect collective anomaly. LOF [11] is a semi-supervised anomaly detecting algorithm. As it outperforms support vector machine (SVM), neural network, and many other mainstream anomaly detecting algorithm [26]. In our scheme, we accelerate LOF algorithm with R-tree [27], which has good performance in storing multidimensional data [28].

Major definitions in LOF algorithm include *kdistance*, *reachability distance*and *local reachability*density. $kdistance(A)$ denotes the distance between point A and A's kth nearest neighbor. $N_k(A)$ is the set of A's k nearest neighbors and $|N_k(A)|$ is the size of the set, which equals to k. $dist(A, B)$ represents the Euclidean distance between A and B. Reachability distance between A and B, $rd_k(A, B)$, is defined as:

$$rd_k(A, B) = \max kdistance(A), dist(A, B). \tag{1}$$

Local reachability density, $lrd_k(A)$, is defined as:

$$lrd_k(A) = \frac{1}{\frac{\sum_{B \in N_k(A)} rd_k(A,B)}{|N_k(A)|}}. \tag{2}$$

LOF value, $LOF_k(A)$, is defined as:

$$LOF_k(A) = \frac{\sum_{B \in N_k(A)} \frac{lrd_k(B)}{lrd_k(A)}}{|N_k(A)|}. \tag{3}$$

The LOF value describes the degree of anomaly. The higher the LOF value is, the further the point A is from its neighborhood. As LOF handles with collective anomaly, we further need SAX algorithm to detect contextual anomaly.

SAX [12] is an approximated representation of time series. It has the advantage of lower computational complexity and closed tightness compared to sophisticated approximation methods such as DFT (Discrete Fourier Transformation) and DWT (Discrete Wavelet Transformation). These features make SAX more favorable in the domain of anomaly detection.

We extend SAX as an algorithm to detect contextual anomaly. The process of SAX algorithm consists of three phases: Normalization, Symbolization and Similarity Comparison. We would like to introduce the following parameters for discussion in our paper.

P and Q represent two time series respectively. p_i and q_i denote the ith element in time series P and Q. $\tilde{P}$ and $\tilde{Q}$ refer to corresponding symbolic series obtained from P and Q. $\tilde{p}_i$ and $\tilde{q}_i$ represent the ith element in symbolic series $\tilde{P}$ and $\tilde{Q}$.

The symbolic distance between $\tilde{P}$ and $\tilde{Q}$ is defined as:

$$SymDist(\tilde{P}, \tilde{Q}) = \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} dist_2(\tilde{p}_i, \tilde{q}_i)} \tag{4}$$

where $dist(a, b)$ is defined as:

$$dist(a, b) = b_{\max(a,b)-1} - b_{\min(a,b)} \text{ when } |a - b| > 1 \tag{5}$$

$dist()$ is not Euclidean distance since the symbolic distance of the letters should be the lower bound of Euclidean distance of raw time series [29]. The detailed definitions, deductions and loop-up-table can be found in [12].

3. Scheme design

We propose our adaptive anomaly detection and identification scheme whose framework is shown in Fig. 1. Our scheme manipulates the statistical data sampled from cloud platform to discover the hostile attacks and report the anomalies with their possible sources to the administrators.

We start our scheme with adaptive LOF algorithm to detect collective anomalies from runtime VM behaviors. If a runtime VM behavior deviates from the normal behaviors in database, we use DR-LOF to figure out the behavioral data that has the most significant impacts on the abnormal condition. Due to possible misinterpretation of contextual information, the intermediate results on anomaly detection may contain unexpected false alarms. We further proceed to a validation process in which we compare the anomalous behavior with normal behaviors from the contextual perspective. This is achieved by comparing the SAX distances of the behaviors before and after migration. A small distance indicates that the VM operates regularly in a new environment. Therefore we consider the report by adaptive LOF algorithm as a false alarm. A large distance confirms the behavior as an anomaly. Our scheme sets off the alarm with the clue of the most anomalous behavioral data.

Please cite this article in press as: T. Huang, et al., Anomaly detection and identification scheme for VM live migration in cloud infrastructure, Future Generation Computer Systems (2015), http://dx.doi.org/10.1016/j.future.2015.06.005

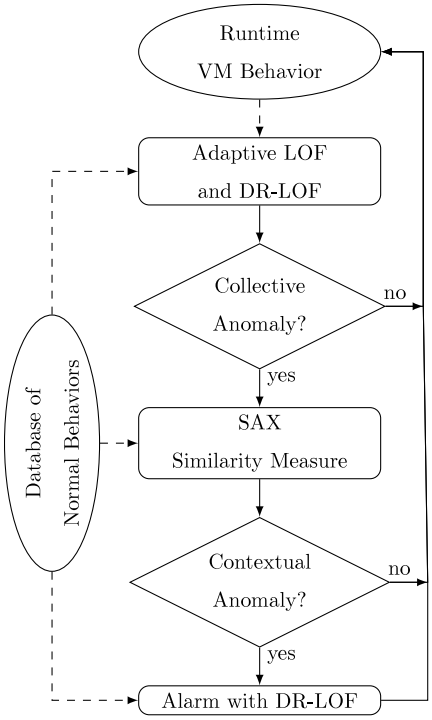


Fig. 1. Adaptive anomaly detection and identification scheme overview.

3.1. Data collection

The inputs of our scheme, which are shown as ellipses in Fig. 1, are the behavioral data of a VM. The inputs consist of runtime VM behaviors and a database of normal behaviors. We create a database of normal behaviors with collected behavior data of a VM in observation during the process of deployment, function testing and performance testing of applications running on the VM. After launching applications into practical operations, we collect the runtime VM behaviors as testing input. We also update the database with runtime VM behaviors observed so that we can learn individual difference and changes of the applications in different VMs [23].

We use resources utilization statistics because they are reasonable representations of the behaviors of a VM. We use SAR (System Activity Reporter) to monitor system activities and collect training data. It reports a lot of statistical indicators, such as file read/write, system call, CPU usage, disk I/O and so on. Although choosing proper indicators is important, our detection scheme has the ability to measure the contribution of each indicator to anomalies. All we need is to choose representative statistics of resource utilization as the input and let our scheme determine which is responsible for the anomalies.

3.2. Adaptive extension to classic LOF

During the construction of database of LOF approach, raw data records are fixed as read-only reference values. In scenario of VM migration, if we do not update reference values, changes brought by migration or other factors would make the results less relevant. Therefore, those methods similar to classic LOF [11], whose reference values are not updated, need to be extended in order to handle the VM migration scenario discussed in this paper.

To explain our adaptive extension to LOF algorithm clearly, let us consider a 2-dimensional case for simplicity. Fig. 2 shows a scenario that a normally behaved cluster moves slowly from C_1 to C_2 due to certain environmental changes. In this case, if we do not update knowledge base and deem C_1 as the normal cluster from beginning to end, data point P_1 will be considered as a regular data point while P_2 will not. Since the cluster representing normal behaviors has moved to C_2, P_2 is actually the normal point while P_1 is not. Moreover, updating is even more important in our migration scenario since the infrastructure settings may change a lot during migration.

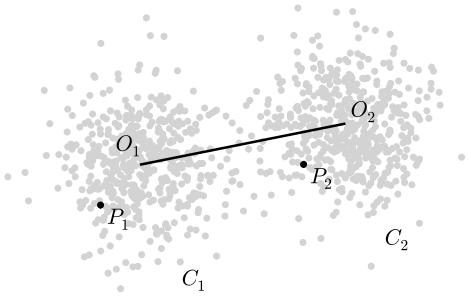


Fig. 2. Illustration of movement of a cluster representing normal behaviors due to environmental changes.

Our experiments show that without updating, if data volume grows linearly or exponentially (e.g. $D_1 = 1, 2, 3,$ or $D_2 = 1, 10, 100, 1000,$), classic LOF value increases rapidly to infinity, which means classic LOF algorithm fails in this scenario. To handle this situation, if we update our storage whenever we figure out a new normal testing point, the LOF value will converge to a valid finite value.

In our adaptive extension to classic LOF algorithm, we identify more operations to avoid potential instability. Data should be normalized before it is put into storage, i.e. R-tree. As the normalization rule is generated in training phase, it may no longer be valid for testing data. Hence, every time a new data record is put in, inspection should be done to all the multi-dimensional data in R-tree to check if their outlines are still similar to a hypercube. If not, the entire R-tree should be re-normalized and rebuilt.

3.3. Dimension reasoning extension to LOF

To identify possible sources of anomalies after detection, we further propose our dimension reasoning extension to classic LOF algorithm, denoted as DR-LOF. The basic idea of DR-LOF is that by removing one dimension data of each data point and recalculating LOF, the ratio of the new LOF value to total LOF value reflects the status of both this data point and the removed dimension of data. If the removed dimension is the most responsible to the anomalous condition, the new LOF value declines. Therefore the smallest value of DR-LOF indicates that the corresponding dimension has the most contribution to the anomalous behavior.

We define DR-LOF as the ratio of LOF values without and with the nth dimension data. The definition can be represented as:

$$DR\text{-}LOF_k^n(A) = \frac{LOF_k^n(A)}{LOF_k(A)} \tag{6}$$

in which $LOF_k^n(A)$ represents the LOF values of A without taking nth dimension data into account. Supposing C is the set of data points of point B's k nearest neighbors. According to Eqs. (1)–(3), we have:

$$\frac{LOF_k^n(A)}{LOF_k(A)} = \frac{\dfrac{\sum_{B\in N_k(A)} \frac{lrd_k^n(B)}{lrd_k^n(A)}}{|N_k(A)|}}{\dfrac{\sum_{B\in N_k(A)} \frac{lrd_k(B)}{lrd_k(A)}}{|N_k(A)|}} = \frac{\sum_{B\in N_k(A)} \dfrac{\sum_{B\in N_k(A)} rd_k^n(A,B)}{\sum_{C\in N_k(B)} rd_k^n(B,C)}}{\sum_{B\in N_k(A)} \dfrac{\sum_{B\in N_k(A)} rd_k(A,B)}{\sum_{C\in N_k(B)} rd_k(B,C)}}$$

Please cite this article in press as: T. Huang, et al., Anomaly detection and identification scheme for VM live migration in cloud infrastructure, Future Generation Computer Systems (2015), http://dx.doi.org/10.1016/j.future.2015.06.005

$$= \frac{\sum_{B\in N_k(A)} rd_k^n(A,B)}{\sum_{B\in N_k(A)} rd_k(A,B)} \times \frac{\sum_{B\in N_k(A)} \frac{1}{\sum_{C\in N_k(B)} rd_k^n(B,C)}}{\sum_{B\in N_k(A)} \frac{1}{\sum_{C\in N_k(B)} rd_k(B,C)}}. \quad (7)$$

Based on Eq. (7), we can obtain the following properties.

Property 1. *If data point A is normal, removing any dimension of A and re-computing LOF with the other dimensions of data will get approximately the same LOF value.*

Property 1 can be explained by making the first assumption: all A's k nearest neighbors are on the hyper-sphere centered at A with a radius of $kdistance(A)$; Therefore, reachability distance between A and B is defined as:

$$rd_k(A,B) = \sqrt{(A_1-B_1)^2 + (A_2-B_2)^2 + \cdots + (A_N-B_N)^2}. \quad (8)$$

A_n and B_n are A and B's projection on the nth dimension coordinate. N is the dimensionality of the data. If we remove the nth dimension, what we get is:

$$(rd_k^n(A,B))^2 = (rd_k(A,B))^2 - (A_n-B_n)^2.$$

The ratio of $rd_k^n(A,B)$ and $rd_k(A,B)$ is:

$$\frac{rd_k^n(A,B)}{rd_k(A,B)} = \sqrt{1-\left(\frac{A_n-B_n}{rd_k(A,B)}\right)^2}. \quad (9)$$

As each dimension usually contributes similar shares to reachability distance, we can make the second assumption for clear calculation that the nth dimension contributes an averaged share of reachability distance, i.e. $1/N$ of $rd_k(A,B)$. With this assumption, we can further simplify Eq. (9) as:

$$\frac{rd_k^n(A,B)}{rd_k(A,B)} = \sqrt{1-\left(\frac{1}{N}\right)}. \quad (10)$$

Consequently, Eq. (7) can be further deduced to:

$$\frac{LOF_k^n(A)}{LOF^k(A)} = \sqrt{1-\frac{1}{N}} \times \frac{1}{\sqrt{1-\frac{1}{N}}} = 1 \quad (11)$$

which means Property 1 holds true.

Property 2. *If data point A is an anomaly:*

- *LOF value without the nth dimensional data decreases if the nth dimension is anomalous;*
- *LOF value without the nth dimensional data increases if the nth dimensional data behaves normally.*

We would like to explain Property 2 as below. Since A is anomalous, it is away from the normally behaving cluster C. Suppose A's projections on U of N dimensions coordinates fall into cluster C and V dimensions do not into cluster C, where $U+V=N$. To be more precise, we give two definitions:

Def. 1. $a = Avg_{j\in U}\{|A_j - B_j|\}$, U is the set of normal dimensions of A;

Def. 2. $b = Avg_{i\in V}\{|A_i - B_i|\}$, V is the set of normal dimensions of A;

B represents some normal points deep in Cluster C. As V is the set of abnormal dimensions, we can make an intuitive assumption that $a < b$.

If the removed nth dimension data is anomalous, which means $n \in V$ for n in Eq. (9), so Eq. (9) can be further deduced to:

$$\sqrt{1-\left(\frac{A_n-B_n}{rd_k(A,B)}\right)^2} = \sqrt{1-\left(\frac{b}{rd_k(A,B)}\right)^2} = \sqrt{1-\frac{b^2}{ua^2+vb^2}} < \sqrt{1-\frac{1}{N}}. \quad (12)$$

The first equal sign holds true based on the assumption that the removed dimension is about the mean of all abnormal dimensions. Consequently,

$$\frac{LOF_k^n(A)}{LOF_k(A)} = \sqrt{1-\left(\frac{A_n-B_n}{rd_k(A,B)}\right)^2} \times \sqrt{\frac{N}{N-1}} < \sqrt{1-\frac{1}{N}} \times \sqrt{\frac{N}{N-1}} = 1 \quad (13)$$

which means the first half of Property 2 holds true. If the removed nth dimension data is normal, Eq. (9) can be similarly deduced to:

$$\sqrt{1-\left(\frac{A_n-B_n}{rd_k(A,B)}\right)^2} = \sqrt{1-\left(\frac{a}{rd_k(A,B)}\right)^2} = \sqrt{1-\frac{a^2}{ua^2+vb^2}} > \sqrt{1-\frac{1}{N}} \quad (14)$$

and,

$$\frac{LOF_k^n(A)}{LOF_k(A)} = \sqrt{1-\left(\frac{A_n-B_n}{rd_k(A,B)}\right)^2} \times \sqrt{\frac{N}{N-1}} > \sqrt{1-\frac{1}{N}} \times \sqrt{\frac{N}{N-1}} = 1 \quad (15)$$

which means the second half of Property 2 holds true.

In our implementation, we remove each dimension and re-compute LOF value to identify which dimension of data leads to biggest decrease in LOF value if we remove it. For example, if the dimension %memused leads to the most significant drop in LOF, it is most likely that memory leakage is the main reason for the abnormal status. Dimension reduction is an additional feature of DR-LOF: only one dimension of data will be the input to SAX subsystem.

3.4. Combining SAX to adaptive LOF

In our design, we adopt SAX to work as a validation process after DR-LOF. As mentioned before, LOF algorithm neglects the time information in data. In the context of VM live migration, the same application working on different hosts shows different characteristics. LOF cannot adapt to such changes and it will mistake normal behaviors for anomalous and vice versa.

SAX reads in the dimension of data which is the most anomalous. After symbolizing the data, it compares the distance before and after migration. If the symbolic distance between them is less than threshold, the anomalies reported by DR-LOF after migration will not be reckoned as real anomalies and the system will not raise alarms. If the symbolic distance is large enough, alarm along with the reason of the anomaly will be reported to the supervisors. Threshold can be determined by the variance of several subsequences in certain testing environment.

Please cite this article in press as: T. Huang, et al., Anomaly detection and identification scheme for VM live migration in cloud infrastructure, Future Generation Computer Systems (2015), http://dx.doi.org/10.1016/j.future.2015.06.005

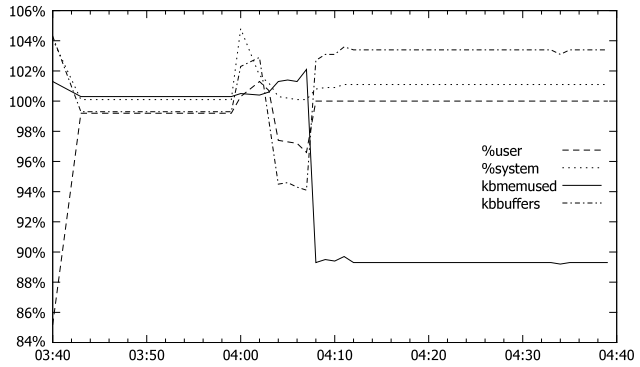


Fig. 3. Ratio of LOF values with/without specific dimensions of data and all dimensions of data in the experiment to evaluate DR-LOF.

4. Performance evaluation

Our testing infrastructure consists of 2 Dell R410 Rack servers (named as SERVER0 and SERVER1) powered by Intel Xeon CPU E5606 @2.13 GHz. Four Xen virtual machines (VMs) are running on the servers with CentOS Linux operating systems. We used Xen hypervisor as the virtualization tool. ScaLapack [30] parallel computation programs are deployed as benchmark on the VMs (VM0–VM3). This application performs large scale matrix decomposition calculation, which causes large consumption of CPU and disk resource.

Our detection scheme does not impose any hypothesis on the infrastructure, platform and software. Our scheme is supposed to be able to detect anomalies in a cloud infrastructure whose size is much larger than our experimental setup.

4.1. Evaluation of DR-LOF

To evaluate our DR-LOF algorithm, we deploy used performance monitoring tool SAR to obtain logs of our testing infrastructure. We sampled a 9-dimensional data set every minute from 03:40 to 04:40. Each dimension, as shown in Fig. 3, are representative indicator of resource utilizations. Three different types of anomaly were injected to see if DR-LOF can successfully recognize the type of anomalies: (1) CPU emulation at 03:40; (2) routine maintenance at 04:02; and (3) memory leakage at 04:08. For demonstration purpose, we did not collect VM behaviors in the scenario of CPU emulation and routine maintenance when constructing the database of normal behaviors. Our scheme will identify them as anomalies.

Fig. 3 shows segments of traces for $LOF_k^n(A)/LOF_k(A)$ from 03:40 to 04:40. We did not show all the results and only 4 most important dimensions are represented in Fig. 3 for legibility.

The dimension of data whose new LOF value achieves the minimal among all dimensions is picked up and recorded in the second column of Table 1. By comparing them with the third column, which shows the real cause of anomalies, we can see that DR-LOF correctly identifies which dimension takes the most responsibility for the anomalous situation. All the dimensions with minimal LOF ratio are relevant to the type of anomaly injected. Besides, routine maintenance is also seen as an anomaly by LOF, and we can find out that in the process of the routine maintenance, the system performs a large number of I/O operations followed by buffer transmissions.

4.2. Anomaly detection in VM migration scenarios

To demonstrate that our proposed scheme can distinguish anomalies without raising false alarms over normal data on new hosts after VM live migration, we use our scheme to figure out when the VM works normally or bears attacks by malicious port scanning. Virtual machine VM0 was live migrated to SERVER0 at time point t_1. After working on SERVER0 for around 1 h, it was migrated back to SERVER1 at time point t_2.

Table 1
Result of DR-LOF and real anomaly injected.

Time period	Attribute with minimal LOF ratio	Anomaly injected
03:40	%user	CPU emulation
03:44–03:59	None	None
04:00–04:07	kbbuffers	Routine maintenance
04:08–04:40	%memused	Memory leakage

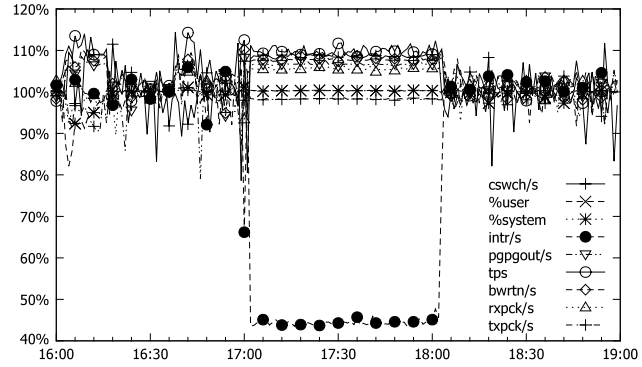


Fig. 4. DR-LOF values of all performance statistics in experiment 1. The DR-LOF value of intr/s largely deviates from 100% during the migration of the VM, indicating that intr/s is the most responsible for the alternated behavior of the VM.

Table 2
Experiment 1: symbolic distance.

Segments	Symbolic distance
(1) and (2)	4.5057
(2) and (3)	4.3429
(1) and (3)	4.3138

4.2.1. VM worked properly on new host

During the period from 17:00 to 17:59, VM0 worked on the destination server (SERVER0). No anomaly was injected during this period. We implemented this experiment to test whether our scheme can successfully adapt to the new infrastructures and do not regard the performance on new hosts as anomaly. Testing data is collected from 16:00 to 18:59.

First, as a reference to compare, classic LOF algorithm was implemented. Most of the anomalies detected by LOF occur between 17:00 and 18:00 which was just the time when VM0 was working on the new host. This phenomenon implies that classic LOF algorithm cannot adapt properly to the new host caused by VM live migration.

Fig. 4 shows the result of DR-LOF. intr/s (number of intrusion per second) is the dimension reflecting the anomaly detected by DR-LOF on the new host as it has the minimal new LOF value. Then we implemented SAX with intr/s and computed the distance between any two of the three periods: (1) 16:00–16:59, (2) 17:00–17:59 and (3) 18:00–18:59, which represent the periods before, during and after VM was migrated to a new server and returned back. SAX results are shown in Table 2. The variance of the Symbolic distances is 2.14×10^{-2} only, which is much lower than a threshold of sequence matching of 0.25. Hence we conclude that no anomaly happens on the new server and refuse to report the anomalies reported by LOF during period (2).

Please cite this article in press as: T. Huang, et al., Anomaly detection and identification scheme for VM live migration in cloud infrastructure, Future Generation Computer Systems (2015), http://dx.doi.org/10.1016/j.future.2015.06.005

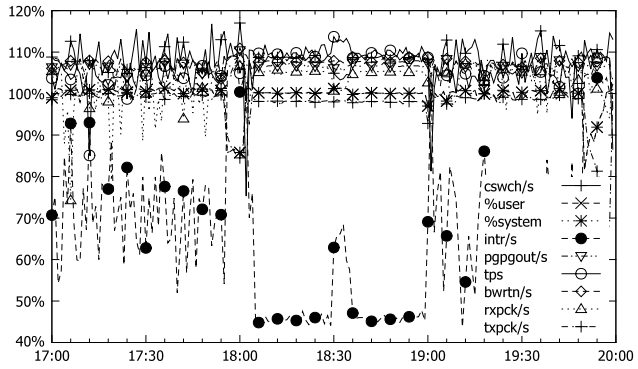


Fig. 5. DR-LOF values of all performance statistics in experiment 2. intr/s is the most responsible for changed behaviors of the VM during the migration. The spike of DR-LOF values of intr/s at 18:30 indicates another anomaly that stems from other dimensions, which are rxpck/s and cswch/s in this case.

Table 3
Experiment 2: symbolic distance.

Segments	Symbolic distance
(1) and (2)	8.0383
(2) and (3)	8.1466
(1) and (3)	4.5757

4.2.2. Malicious port scanning on new host

Port scanning is the act of systematically scanning a computer's ports. Port scanning has legitimate uses in managing networks, but port scanning also can be malicious in nature if someone is looking for a weakened access point to break into target computer. In this experiment, we perform malicious port scanning towards VM0. The experimental settings are as follows. VM0 worked on SERVER0 from 18:00 to 18:59, and returned back to SERVER1 at 19:00. During the time from 18:30 to 18:35, malicious port scanning was performed. We conduct this experiment to demonstrate the ability of our scheme to detect anomalies on new hosts.

DR-LOF and SAX were implemented. Fig. 5 shows that intr/s (number of interrupts per second) is the most significant dimension, followed by rxpck/s, txpck/s, and system%. We can infer from the figure that the anomaly is a kind of network attack that incurs a lot of interrupts and thus CPU time for system. The symbolic distances among any two of the periods (1) 17:00–17:59, (2) 18:00–18:59 and (3) 19:00–19:59 are shown in Table 3. The variance of the symbolic distances is 8.251, so the anomalies reported by LOF in period (2) should be considered as real anomalies.

4.2.3. Heartbleed exploitation on new host

The Heartbleed vulnerability took the Internet by surprise in April 2014. The vulnerability, one of the most consequential since the advent of the commercial Internet, allowed attackers to remotely read protected memory from an estimated 24%–55% of popular HTTPS sites [31].

In this experiment we construct the situation of repeated heartbleed attack. We set up a web server and established 20 TLS links from guest machine to the web server. We issue 100 malformed heartbeat requests, ten at a time. We count the difference between the packet size of heartbeat requests and replies every 5 min. We apply both discovery methods on the record of difference to detection the exploitation.

DR-LOF and SAX were implemented. Fig. 6 shows that txpck/s (number of package sent per second) is the most significant dimension. We can infer from this figure that the attacks are a kind of network attacks that incur a lot of outward traffics from the target machine. The symbolic distances among any two of the periods (1) 11:00–11:59, (2) 12:00–12:59 and (3) 13:00–13:59 are shown in Table 4. The variance of the symbolic distances is 7.560, so the anomalies reported by LOF in period (2) should be considered as real anomalies.

Table 4
Experiment 3: symbolic distance.

Segments	Symbolic distance
(1) and (2)	7.4362
(2) and (3)	8.0910
(1) and (3)	4.4443

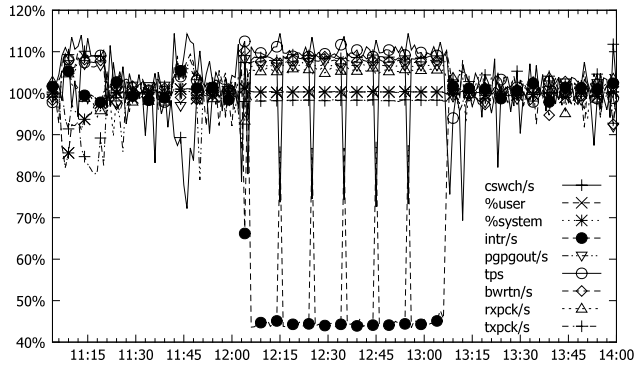


Fig. 6. DR-LOF values of all performance statistics in experiment 3. intr/s is the most responsible for the alternated behavior of the VM during the migration. The DR-LOF values of intr/s and cswch/s change inversely as time goes on during the migration. The five pairs of inverted spikes of the curves indicate that cswch/s becomes the major cause to the alternated behavior of the VM.

4.3. Sensitivity of our scheme with respect to major parameters

In this section, we present sensitivity analysis of our scheme with respect to major parameters in our scheme, i.e. k and threshold, as informal robustness analysis of our scheme.

4.3.1. k

The calculations of LOF-related algorithms require the parameter k, which represents the number of nearest neighbors. We theoretically and experimentally analyzed the impact parameter k of LOF to our scheme as follows.

Parameter k affects the LOF value by determining the tolerance of anomalies in training data. Unsupervised detection method requires training data to build model of normal behaviors. Generally it is difficult to ensure zero anomalies in training data. LOF has the ability to tolerate small number of anomalies in training data. Parameter k determines the tolerance of anomalies in training data. If we happen to record an anomalous behavior m times into training data, we adjust parameter k to be greater m to ensure that the anomalous behavior will not be recognized as a normal behavior during testing phase. In most cases, it is impossible to build model of normal behaviors using a training data set with an anomaly occurs tens of times during the process of collecting training data. Therefore it should be safe to use k value of 30 in our scheme.

According to [11], k should be between 10 and 50 to ensure the proper functionality of LOF. We observed that an empirical value of k between 20 and 40 works fine for most applications. We choose values between 20 and 40 to examine the sensitivity of our scheme to the parameter k. Fig. 7 shows the LOF value of testing data with parameter $k = 20$, $k = 30$ and $k = 40$ in experiment 2.

The points marked with plus and triangle are the LOF values calculated with $k = 20$ and $k = 40$ respectively. These points are very close to points marked with circle ($k = 30$). The points during the period of migration (18:00–19:00) are all above the threshold

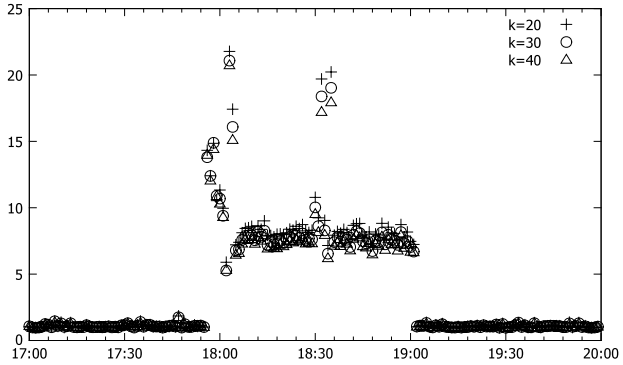


Fig. 7. The sensitivity of LOF to its parameter k. Y axis stands for LOF values while X axis stands for time.

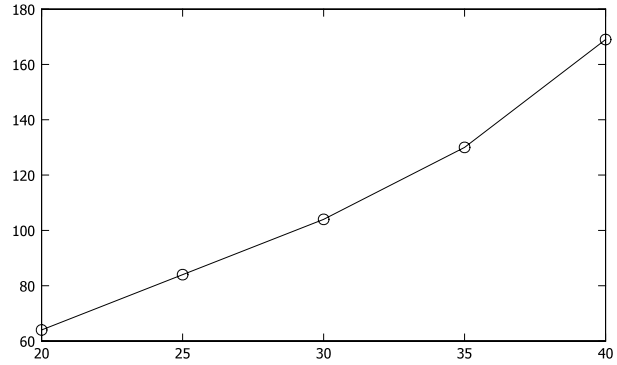


Fig. 8. Time consumption of DR-LOF with different parameter k.

of 2.5 we set in our scheme. The figure indicates that the parameter k has trivial impacts on the LOF calculations in our scheme.

Parameter k also has trivial impacts on DR-LOF since DR-LOF is based on LOF. Fig. 5 shows DR-LOF values calculated with $k = 30$, we plotted Fig. 5 again with $k = 20$ and $k = 40$, and obtained similar results compared to those calculated with $k = 30$. As such, the details of DR-LOF calculated with $k = 20$ and $k = 40$ will not be repeated here.

Though producing little impacts on accuracy of our scheme, parameter k has obvious impacts on the time consumption of our scheme. The worst-case complexity of LOF with respect to k is $O(k^3)$. In our experiments, we implemented caches of lookup tables for KNN queries to reduce the average case complexity of DR-LOF. Fig. 8 shows the wall clock time consumed by DR-LOF detection scheme on 2160 points using an Intel Core i5 processor and Java implementation.

As shown in Fig. 8, calculation time increases with k. However, DR-LOF algorithm is still able to handle more than 10 points of samples per second with a parameter $k = 40$. The performance is fast enough for an online anomaly detection method since there are less than 10 points of samples within a second.

With analytical and experimental results, we would like to draw the conclusion that changing parameter k within the range from 20 to 40 has trivial impacts on our scheme in terms of accuracy. Performance wise, there is no need in general to adjust parameter k to optimize the performance of our scheme since the performance appears to be sufficient for online detection.

4.3.2. Threshold

We introduce a threshold of LOF values to classify all behaviors into normals and anomalies. The threshold is a critical parameter to the detection rate and the false alarm rate of our method. We present the method to determine proper value of the threshold and analyse the sensitivity of our scheme to the threshold.

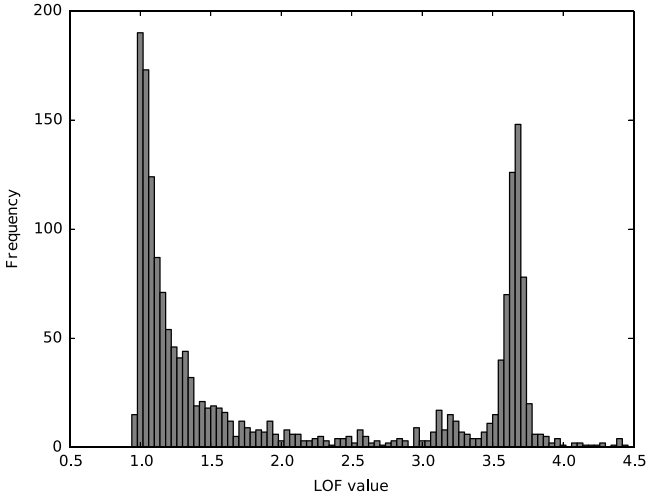


Fig. 9. A histogram of LOF values in experiment 2.

Table 5
Comparison of our scheme and classic LOF scheme.

Experiment 1	Our scheme	Classic LOF
(1) and (2)	NA	NA
(2) and (3)	0.3%	18.4%

We determine the threshold based on the distribution of LOF values of normal behaviors. We ran a two-day test of the ScaLapack to collect observations of normal behaviors. Anomalies were not injected during the test. We calculated the LOF values of all observations and select a threshold that is greater than 99.9% LOF values. In this case the threshold is 2.5. This threshold accepts 99.9% behaviors observed in the two-day test and reject others as anomalies. To verify the effectiveness of the threshold, we plotted a histogram of LOF values of all points in experiment 2.

Fig. 9 shows the histogram of LOF values of all points in experiment 2. X axis represents the LOF values. Y axis represents the frequency of observation of corresponding LOF values. The bars on the left side represent the observations of normal behaviors, whereas the bars on the right represent observations of anomalous behaviors. There is few observation of behaviors whose LOF value is between two and three. According to the distribution of the LOF values shown in Fig. 9, a threshold between two and three is able to divide observations of behaviors into normal ones and anomalies. Therefore, the threshold of 2.5 works properly in the experiments of our scheme.

4.4. Comparison with other anomaly detection methods

To determine technical quality of our scheme more accurately, we would like to compare our scheme against existing peer methods. The comparisons against classic LOF schemes and clustering schemes are presented as follows.

4.4.1. Our scheme vs. classic LOF

The differences between our scheme and classic LOF [11] lie in: (1) we introduce adaptive LOF by updating R-tree; (2) we propose DR-LOF which can identify the reason of anomalies; (3) SAX algorithm is implemented to make the scheme adaptive to VM migration and other infrastructure changes. Detection rates and false alarm rates are shown in Table 5.

Since we have not injected anomaly into the data of experiment 1, the detection rate of classic LOF and our scheme is not available. Classic LOF simply regards the whole part of data during migration as anomalies. So the false alarm rate of classic LOF is high. In

Please cite this article in press as: T. Huang, et al., Anomaly detection and identification scheme for VM live migration in cloud infrastructure, Future Generation Computer Systems (2015), http://dx.doi.org/10.1016/j.future.2015.06.005

Table 6
Performance of clustering algorithms in experiment 1/2/3.

Experiment 1/2/3	Out of migration recognition	During migration recognition	Incorrect clustered instances
EM	91%/79%/78%	91%/84%/100%	8.5%/19.4%/13.8%
Density based	100%/79%/78%	100%/84%/100%	1%/18.8%/14.4%
X-means	92%/82%/81%	91%/88%/100%	8.3%/16.8%/12.7%

Table 7
Performance of clustering algorithms in experiment 2/3.

Experiment 2/3	Detection rate	False alarm rate
EM	79%/0%	84%/14%
Density	79%/0%	84%/6.1%
X-means	82%/100%	88%/10.2%
Our scheme	98.3%/100%	16.9%/1.1%

our scheme, we use SAX to evaluate the similarity of normal data sequence and the false positive one. The data sequence before migration and during migration is similar, which means there is no anomaly during migration. So we reduce false alarm rate through SAX phase of our scheme.

4.4.2. Our scheme vs. clustering algorithms

Clustering is one of the most popular anomaly detection methods. We use three clustering algorithms: EM (Expectation Mean), Density based and X-means clustering tool provided by WEKA (Waikato Environment for Knowledge Analysis) [32] to perform anomaly detection on the same groups of data (see Table 6). The detecting results of experiment 1/2/3 are shown in Table 7.

Clustering algorithms over data in experiment 1. Density based and X-means clustering over the testing data set of experiment 1. The data set is mainly clustered into two parts, i.e. the part taken within migration and the other part taken out of migration. Density based method works the best among the three. It clearly divides the data set into two parts with almost no mistakes. The results implies that clustering method works well on identifying migration of virtual machine.

Clustering algorithm over data in experiment 2. Density based and X-means clustering over the testing data set of experiment 2. In this experiment, anomalies due to malicious port scanning were injected during the migration phase. All these methods barely identify the changing phase within and out of the migration process. They have incorrectly clustered many data instances because of the disturbance of anomalies. Most data instances during migration, including anomalies, are clustered into a group, which implies that these methods cannot distinguish anomalies from the normal behaviors during the migration process.

It is worth noting that we attained these results by iteratively adjusting the number of clusters until we get an optimized answer. Without knowing the migration state and the types of anomalies, these algorithms cannot work properly.

Table 7 shows the performance comparison of our scheme and these clustering algorithms. From Table 7 we know that these clustering algorithms have the ability to identify migration of virtual machine. But their performance drops when anomaly occurs during the migration. They can neither distinguish anomaly from normal behavior during the migration. Their detection rate and false alarm rate are not satisfactory. Our scheme has better detection rate and it does not compromise much false alarm rate. In addition, clustering mixes all the dimensions of data, which makes it impossible to detect the reason of anomalies. To conclude, comparing with our scheme, clustering is not appropriate for anomaly detection in the scenario of VM migration.

5. Conclusion

To tackle a critical security issue that arose from VM live migration in a shared cloud environment, we proposed an adaptive anomaly detection scheme which is robust to detection of collective and contextual anomalies. Our scheme consists of a combination of the adaptive DR-LOF and SAX. It not only adapts to new host with different infrastructures on the context of VM live migration, but also identifies the possible sources of anomaly. Combination of DR-LOF with SAX takes the time dimension into account to handle further contextual anomalies and thus reduces false alarm rates.

Our empirical performance evaluation shows that our scheme is capable of detecting and identifying such anomalies as CPU emulation, memory leakage, and malicious port scan as well other anomalies. Experimental results show that our scheme detects anomalies more effectively than classic LOF and clustering methods in popular WEKA package. The detection rates were improved from 79% to more than 98% while false alarm rates were reduced from 84% to 16.9%. More importantly, our DR-LOF scheme is able to identify possible sources of the anomalies, which classic LOF fails, giving cloud computing operators important clues to the anomalies.

Acknowledgments

The work presented in this paper was funded in part by the Shanghai International Science and Technology Collaboration Program under Grant 13430710400, and Campus for Research Excellence and Technological Enterprise (CREATE) program of Singapore National Research Foundation under the joint project on Energy and Environmental Sustainability Solutions for Megacities (R-706-000-101-281 and R-252-000-496-281).

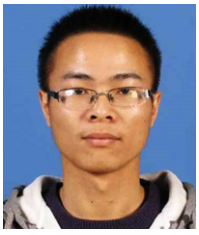
References

[1] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al., A view of cloud computing, Commun. ACM 53 (4) (2010) 50–58.
[2] D. Ridge, D. Becker, P. Merkey, T. Sterling, Beowulf: harnessing the power of parallelism in a pile-of-PCs, in: Aerospace Conference, 1997. Proceedings., IEEE, Vol. 2, IEEE, 1997, pp. 79–91.
[3] A. Beloglazov, J. Abawajy, R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, Future Gener. Comput. Syst. 28 (5) (2012) 755–768.
[4] J. Tordsson, R.S. Montero, R. Moreno-Vozmediano, I.M. Llorente, Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers, Future Gener. Comput. Syst. 28 (2) (2012) 358–367.
[5] A.J. Ferrer, F. Hernández, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R.M. Badia, K. Djemame, et al., Optimis: A holistic approach to cloud service provisioning, Future Gener. Comput. Syst. 28 (1) (2012) 66–77.
[6] H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: A comprehensive review, J. Netw. Comput. Appl. 36 (1) (2013) 16–24.
[7] K. Adamova, D. Schatzmann, B. Plattner, P. Smith, Network anomaly detection in the cloud: The challenges of virtual service migration, in: 2014 IEEE International Conference on Communications (ICC), IEEE, 2014, pp. 3770–3775.
[8] Y. Liu, Z. Yuan, C. Xing, B. Gong, Y. Xiao, H. Liu, A behavioral anomaly detection strategy based on time series process portraits for desktop virtualization systems, Cluster Comput. (2015) 1–10.
[9] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in cloud, J. Netw. Comput. Appl. 36 (1) (2013) 42–57.
[10] A. Milenkoski, S. Kounev, A. Avritzer, N. Antunes, M. Vieira, On benchmarking intrusion detection systems in virtualized environments, ArXiv Preprint arXiv:1410.1160.

[11] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: ACM SIGMOD Record, Vol. 29, ACM, 2000, pp. 93–104.
[12] J. Lin, E. Keogh, S. Lonardi, B. Chiu, A symbolic representation of time series, with implications for streaming algorithms, in: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, ACM, 2003, pp. 2–11.
[13] T. Wang, W. Zhang, J. Wei, H. Zhong, Workload-aware online anomaly detection in enterprise applications with local outlier factor, in: 2012 IEEE 36th Annual Computer Software and Applications Conference (COMPSAC), IEEE, 2012, pp. 25–34.
[14] P. Kumar, N. Nitin, V. Sehgal, K. Shah, S. Shukla, D. Chauhan, A novel approach for security in cloud computing using hidden Markov model and clustering, in: 2011 World Congress on Information and Communication Technologies (WICT), IEEE, 2011, pp. 810–815.
[15] H. Gao, D. Zhu, X. Wang, A parallel clustering ensemble algorithm for intrusion detection system, in: 2010 Ninth International Symposium on Distributed Computing and Applications to Business Engineering and Science (DCABES), IEEE, 2010, pp. 450–453.
[16] Z. Xie, T. Quirino, M.-L. Shyu, S.-C. Chen, L. Chang, UNPCC: A novel unsupervised classification scheme for network intrusion detection, in: 18th IEEE International Conference on Tools with Artificial Intelligence, 2006, ICTAI'06, IEEE, 2006, pp. 743–750.
[17] C.-C. Lo, C.-C. Huang, J. Ku, A cooperative intrusion detection system framework for cloud computing networks, in: 2010 39th International Conference on Parallel Processing Workshops (ICPPW), IEEE, 2010, pp. 280–284.
[18] W. Wang, Y. Zhang, B. Lin, X. Wu, K. Miao, Secured and reliable VM migration in personal cloud, in: 2010 2nd International Conference on Computer Engineering and Technology (ICCET), Vol. 1, IEEE, 2010, pp. 705–709.
[19] C. Xianqin, W. Han, W. Sumei, L. Xiang, Seamless virtual machine live migration on network security enhanced hypervisor, in: 2nd IEEE International Conference on Broadband Network & Multimedia Technology, 2009, IC-BNMT'09, IEEE, 2009, pp. 847–853.
[20] M. Alshawabkeh, J.A. Aslam, D. Kaeli, J. Dy, A novel feature selection for intrusion detection in virtual machine environments, in: 2011 23rd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2011, pp. 879–881.
[21] S. Biedermann, M. Zittel, S. Katzenbeisser, Improving security of virtual machines during live migrations, in: 2013 Eleventh Annual International Conference on Privacy, Security and Trust (PST), IEEE, 2013, pp. 352–357.
[22] J. Oberheide, E. Cooke, F. Jahanian, Empirical exploitation of live virtual machine migration, in: Proc. of BlackHat DC Convention, Citeseer, 2008.
[23] T. Huang, Y. Zhu, Q. Zhang, Y. Zhu, D. Wang, M. Qiu, L. Liu, An LOF-based adaptive anomaly detection scheme for cloud computing, in: 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops (COMPSACW), IEEE, 2013, pp. 206–211.
[24] Q. Zhang, Y. Wu, T. Huang, Y. Zhu, An intelligent anomaly detection and reasoning scheme for VM live migration via cloud data mining, in: 2013 IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, 2013, pp. 412–419.
[25] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. (CSUR) 41 (3) (2009) 15.
[26] A. Lazarevic, L. Ertöz, V. Kumar, A. Ozgur, J. Srivastava, A comparative study of anomaly detection schemes in network intrusion detection, in: SDM, SIAM, 2003, pp. 25–36.
[27] A. Guttman, R-Trees: A Dynamic Index Structure for Spatial Searching, Vol. 14, ACM, 1984.
[28] S. Hwang, K. Kwon, S.K. Cha, B.S. Lee, Performance evaluation of main-memory r-tree variants, in: Advances in Spatial and Temporal Databases, Springer, 2003, pp. 10–27.
[29] E. Keogh, J. Lin, A. Fu, HOT SAX: Efficiently finding the most unusual time series subsequence, in: Fifth IEEE International Conference on Data Mining, IEEE, 2005, pp. 226–233.
[30] J. Choi, J.J. Dongarra, R. Pozo, D.W. Walker, ScaLAPACK: A scalable linear algebra library for distributed memory concurrent computers, in: Fourth Symposium on the Frontiers of Massively Parallel Computation, 1992, IEEE, 1992, pp. 120–127.
[31] Z. Durumeric, J. Kasten, D. Adrian, J.A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, et al., The matter of heartbleed, in: Proceedings of the 2014 Conference on Internet Measurement Conference, ACM, 2014, pp. 475–488.
[32] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, ACM SIGKDD Explor. Newslett. 11 (1) (2009) 10–18.

Tian Huang was born in Shanghai, China in 1986. He received the B.S. degree in Electronics Science and Technology from Shanghai Jiao Tong University in 2008. He is currently pursuing a Ph.D. degree in Electronics Science and Technology at Shanghai Jiao Tong University. His research interest includes Computer Architecture, Embedded System, Multi-media Signal Processing, Parallel Processing, and System on Chip.

Yongxin Zhu is an Associate Professor with the School of Microelectronics, Shanghai Jiao Tong University, China. He is a senior member of China Computer Federation and a senior member of IEEE. He received his B.Eng. in EE from Hefei University of Technology, and M. Eng. in CS from Shanghai Jiao Tong University in 1991 and 1994 respectively. He received his Ph.D. in CS from the National University of Singapore in 2001. His research interest is in computer architectures, embedded systems, medical electronics and multimedia. He has authored and co-authored over 60 English journal and conference papers and 30 Chinese journal papers. He has filed 20 Chinese patents among which 8 have been approved.



Yafei Wu was born in Zhejiang, China in 1990. He received the B.S. degree in Microelectronics from Shanghai Jiao Tong University in 2013. He is currently pursuing an M.Eng degree in CS at Shanghai Jiao Tong University. His research interest includes Computer Architecture, Anomaly Detection and Distributed Computing.

Stéphane Bressan is an Associate Professor at the School of Computing of the National University of Singapore. He joined the National University of Singapore in 1998. He is also adjunct an Associate Professor at the Malaysia University of Science and Technology (MUST) since 2004. His domain of research is the integration and the management of disparate information, i.e. the integration and management of multi-modal and multimedia information from distributed, heterogeneous, and autonomous sources. He is the author or co-author of more than 100 papers in peer reviewed international journals and conferences. He is member of the program and editorial committees of several conferences and journals.

Gillian Dobbie is a Professor in the Department of Computer Science at the University of Auckland, New Zealand. She received a Ph.D. from the University of Melbourne, an M.Tech.(Hons) and B.Tech.(Hons) in Computer Science from Massey University. She has lectured at Massey University, the University of Melbourne and Victoria University of Wellington and held visiting research positions at Griffith University and the National University of Singapore. Her research interests include formal foundations for databases, object oriented databases, semi-structured databases, logic and databases, data warehousing, data mining, access control, e-commerce and data modeling. She has published over 100 international refereed journal and conference papers.

Please cite this article in press as: T. Huang, et al., Anomaly detection and identification scheme for VM live migration in cloud infrastructure, Future Generation Computer Systems (2015), http://dx.doi.org/10.1016/j.future.2015.06.005