Consider the Research of Virtual Machine Co-Location on Cloud Computing Platform

Shengchao Xu
School of Date Science Huashang College, Guangdong University of Finance & Ecnomics, Guangzhou, 511300, China

Abstract: The traditional virtual machine co-stationing method adopts the principle of delay change of virtual machine grouping. The sensitivity of co-stationing processing is low, resulting in low co-stationing coverage rate and low co-stationing accuracy. Aiming at the above problems, a virtual machine co-resident method considering cloud computing platform is studied. After constructing the hidden channel in the cloud computing environment, the decision standard of virtual machine co-location is established. By solving the maximum concurrence probability of virtual machine and combining the criteria of concurrence determination, the concurrence probability of virtual machine is realized. In the comparative simulation experiment, it is verified that compared with the stationed method, the accuracy of the proposed method is improved by 13.46%, and the coverage of the latter is larger.

Keywords: Cloud computing platform; Virtual machine in residence; Concealed channel; Virtual machine flooding

1. Introduction

Cloud computing is a kind of distributed computing, which refers to the network "cloud" through the huge data computing processing program into countless small programs, and then through a system composed of several servers to process and analyze these small programs to get the results and return to the user. With this technology, tens of thousands of data can be processed in a very short time to achieve a powerful network service. When the cloud computing platform is deployed on the cloud server, users only need to connect to the cloud server through the terminal to start their work, which eliminates the limitation of time and regin and solves the problem of cross-regional cooperation between departments, which is a great improvement in efficiency for enterprises with scattered departments [1]. Virtual machine refers to a complete computer system running in a completely isolated environment with complete hardware system functions simulated by software. Co-location refers to the fact that cloud service providers often allocate virtual machines from different tenants to the same physical machine in order to effectively utilize physical resources. As a result, the attacker can have the opportunity to host his own virtual machine with the victim's virtual machine, thus threatening the confidentiality and availability of the victim's virtual machine resources [2].

The traditional virtual machine co-stationing method adopts the principle that the round-trip delay of network packet between two virtual machines will change along with the stationing or not, and fails to take into account the complex network environment in the cloud platform, thus causing inaccurate measurement of network packet delay and poor virtual machine co-stationing effect [3]. In order to improve the effective detection rate of virtual machine co-stationing, this paper will study the virtual machine co-stationing method considering cloud computing platform, and verify the performance of the method through simulation experiment.

2. Consider the Research of Virtual Machine Co-Location on Cloud Computing Platform

2.1. Constructing hidden channels in cloud computing environment

In noisy public cloud environment, the hidden channel of microprocessor architecture will face more problems. Virtual machine migration, V CPU scheduling, and Hypervisor activity add a lot of noise, all of which pose challenges to covert channel precise synchronization. Cache is more attractive to attackers than other covert channel media. Because cache's high operating speed can generate high bandwidth and is not limited by the software system, many advanced isolation mechanisms can be bypassed, so cache based covert channels have attracted much attention in recent years [4]. Among them, the most commonly used alternate communication technologies are Prime + Probe and Flush + Relo. However, before implementing an LLC based covert channel, two kinds of address mapping uncertainties need to be resolved. 1) The user layer application manipulates the virtual address, while the cache is physically marked. The location of the data in the cache is determined by its physical memory address. The conversion between the physical address and the virtual address in the operating system is carried out by the hardware MMU, so it is difficult for the user layer to obtain this information. 2) Newer Intel processors map cache addresses using hash functions not declared in official documents, LLC's non-public hash indexing mechanism. The mapping between LLC slices and physical memory addresses is determined by the hash function, and even if an attacker can determine which cache rows are contained in a cache group, he will not know which LLC slices correspond to those cache rows. To address the above issues, the Linux Super Pages mechanism is used to map additional address offsets as much as possible to maintain the mapping between the LLC physical address and the memory virtual address, as shown in the Figure [5].

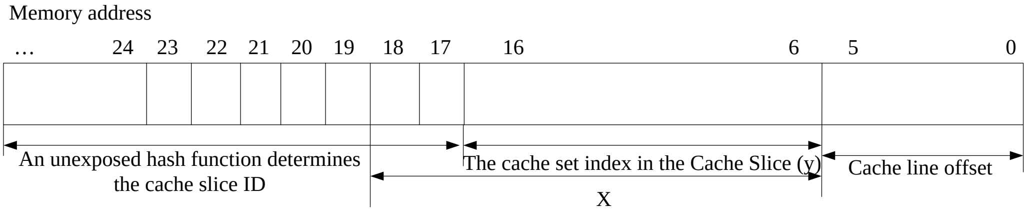

Memory address
…
24
23
22
21
20
19
18
17
16
6
5
0
An unexposed hash function determines the cache slice ID
The cache set index in the Cache Slice (y)
Cache line offset
X

Figure 1. Mapped index of memory with the new LLC

At this point, the communicating parties can only locate the index y of each cache slice, while the slice ID is unknown. By combining the eviction sets with the same elements, the scheme of the eviction sets constructed in this paper is more complete, and the eviction rows that can be evicted beyond the number of group links can be found in the eviction sets corresponding to each slice [6]. In the actual construction of the channel, the sender can access four memory blocks simultaneously during the confict_set construction step to empty the destination cache set. Accessing four slices in a row ensures that the cache set must be occupied by data from the sender. After constructing the hidden channel in the cloud environment, the virtual machine co-resident model is established.

2.2. Establish virtual machine co-location criteria

In the cloud environment, the usage of Shared resources is complex and changeable and transparent to users, and the attacker is only the user of Shared resources, not the owner, and his ability to grasp information is limited. According to the service characteristics of the cloud computing platform, in order to establish the co-location model of the virtual machine, it is assumed that the location of the target virtual machine in the cloud environment is known, but the attacker is not aware of it, but knows clearly what kind of service the target virtual machine provides to the outside world, and can request this service. At the same time, assuming that the target virtual machine provides encryption services to the public, all networked computers can initiate service requests to the target virtual machine and obtain service responses. Then the specific content of the criteria for determining the co-location of virtual machines is as follows [7]:

(1) Create a data space larger than the final Cache size and write it to the Shared Cache through read operation.
(2) Wait for virtual machine 1 and 2 in the same physical server to run for some time in order to replace the contents of the Cache.
(3) Access this set of data again, get time T1, and write the data into the Shared Cache again.
(4) Continuously initiate service requests to target virtual machine 1 through another computer to make it run under high load.
(5) Access this set of data again and get time T2.
(6) Determine whether there is a Shared Cache space between the target virtual machine 1 and the attack virtual machine by comparing the time of T1 and T2, so as to achieve the purpose of simultaneous standing detection.
In this process, if the attacking virtual machine and the target virtual machine are both resident together, the target host will replace a large amount of contents in the Cache during the continuous service request, thus greatly increasing T2 time. If different, there is no obvious time change. However, the characteristics of co-standing time loss will also occur if the following situation occurs: assume that the object of co-standing detection by the attacker is the target virtual machine 2, and in the process of (4), the attacking virtual machine continuously makes service requests to the target virtual machine 2 [8]. At this point, noise virtual machine 1 or 2 suddenly runs under high load, resulting in a large increase in time T2 in (5). According to the above requirements, the virtual machine can be preliminarily determined.

2.3. Virtual machine with resident implementation

Firstly, through the legal way provided by cloud computing platform, apply for a batch of virtual machine cluster simultaneously. In this batch of virtual machine clusters, if a virtual machine falls on each physical host, the probability is $p=(p_1,p_2,\cdots,p_m)$, m is the total number of physical hosts, and $\sum p_i=1$. An estimate of p is made to determine the maximum concurrence probability that the virtual machine flooding can achieve. When the external request is sent to the cloud computing platform, the number of virtual machine instances opened by the request is recorded as n. After the co-location detection, the distribution of all n virtual machine instances on m physical hosts is $x=(x_1,x_2,\cdots,x_m)$, and x_i represents the number of instances on the i th physical machines, $\sum x_i=n$. Then, the maximum congruent probability of the virtual machine is as follows [9]:

$$p(x)=\prod p_i^{x_i} \tag{1}$$

p was estimated by the maximum likelihood method, and then the maximum logarithmic likelihood function was solved by formula (1). The solution process was as follows:

$$\begin{cases}\ln(L(p))=\sum x_i \ln p_i \\ \hat{p}=\arg\max\limits_{p} L(p)\end{cases} \tag{2}$$

Let $g(p)=\sum p_i=1$, introduce the Lagrangian multiplier into formula (2), and get the following formula:

$$\ln(L(p))=\ln(L(p))+\lambda(1-g(p)) \tag{3}$$

In formula (3), λ is the Lagrange multiplier. When the maximum value is obtained in formula (3), the partial derivative of independent variable is 0, which satisfies the following equation:

$$\frac{\partial}{\partial p_i}\ln(L(p)+\lambda(1-g(p)))=0 \tag{4}$$

Combined with the above formula, the following formula can be obtained:

$$\hat{p}=\frac{x_i}{n} \tag{5}$$

After knowing the co-location of the flood virtual machines they hold, attackers can reduce the number of experimental virtual machines on each physical host to only one on each physical host, which will greatly reduce the cost and overhead of subsequent co-location attacks. If the target virtual machine is on the i th physical host at this time, and N_i virtual machines already exist on it, then $\frac{\ln(1-\delta)}{\ln\left(1-\frac{N_i}{m}\right)}$ virtual machines will be opened at the same time, with a probability that the attack virtual machine and the target virtual machine will exist together [10]. The virtual machine held by the attacker running on each physical host is called probe virtual machine, and its cluster is set as $P=\{probe|p_1,p_2,\ldots,p_m\}$; Remote devices owned by the attacker are called remote terminals. If it is found that a virtual machine p_i in P is performing a DoS attack with a service interruption or a great delay, it can be determined that p_i has been residing with the target virtual machine. Through the above steps, virtual machine co-residence is realized and the virtual machine co-residence method considering cloud computing platform is studied.

3. Simulation Results

The virtual machine co-resident method considering cloud computing platform is studied above. This section will verify the performance of this method through simulation experiment.

3.1. Experiment content

This experiment adopts the form of comparison between the virtual machine co-stationing method considering cloud computing platform studied above and the traditional virtual machine co-stationing method. The experimental data of the comparative experiment indexes can be obtained intuitively, so as to evaluate the feasibility and reliability of the co-stationing method studied above. The comparison index of the simulation experiment is the homing coverage rate of the two homing methods when executing the homing operation on the virtual machine, the homing accuracy rate and the homing coverage rate when launching attacks on different numbers of virtual machines. Data processing software is used to process the collected experimental data and analyze the experimental data to complete the simulation experiment.

3.2. Experimental process

The experiment in this paper consists of ten Inspo Infn server NF5280M3 to form a virtualized operating environment based on KVM. Its main configuration information is Intel I5 CPU model, 16G memory, 1TB hard disk and Centos 7.4 system. Virtual machine configuration information is: 2Vcpu, 2G memory, 15GB hard disk, Windows 8.1 operating system. Virtual machine with Eclipse, dec-c ++ and other programming environment, attack the target virtual machine to provide AES file encryption service.

In contrast with two kinds of methods with in correct: in ten servers set up a number of virtual machine, and in all the virtual machines running in the random noise function, the function through the study of the modular arithmetic of random Numbers, according to the situation of remainder random trigger scale, noise and noise by setting the Cache data, filling can be divided into three kinds of load noise size: weak noise, the noise and the strong noise, its data repetition rate were 75%, 50% and

25% respectively. 10 groups of experiments were conducted respectively. The noise virtual machines in the 10 groups of experiments were 1, 2, 3... And ten. For each group, 50 times of co-stationary judgment were performed, and the average accuracy of co-stationary processing was taken for comparison of accuracy.

Compare the same resident coverage of different number of virtual machines launching attacks: set the targeted launch attacks of different number of virtual machine user accounts, and compare the effective coverage of virtual machines under the flood total number of different virtual machines, so as to compare the protection effect of the two virtual machines co-resident methods on the physical host. The final conclusion of the simulation experiment is obtained by synthesizing the above two experimental data analysis conclusions.

3.3. Experimental results

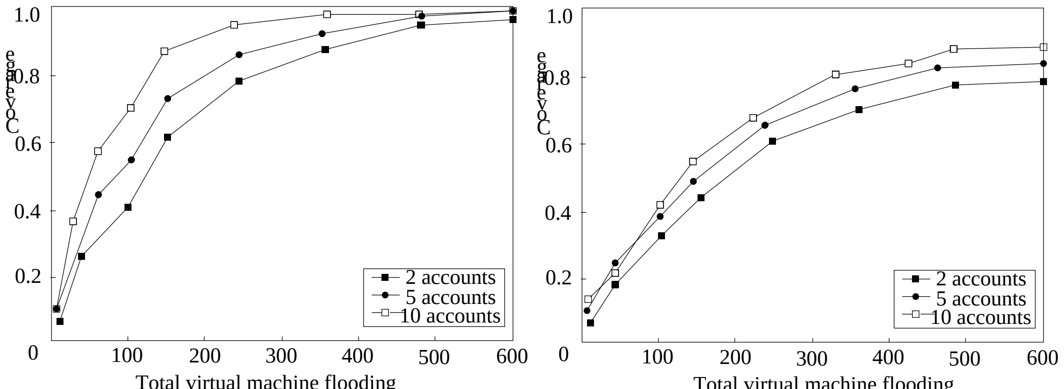
The comparison results of the co-standing accuracy of the two groups of virtual machines are shown in the table below, and the data in the table are compared to compare the effectiveness of the co-standing method of virtual machines.

Table 1. Comparison of the congruent rates between the virtual machine and the stationary method

Serial number	This paper's method		Traditional method	
	Accuracy rate	False detection rate	Accuracy rate	False detection rate
1	87.5	2.89	73.3	8.2
2	85.3	3.15	68.1	8.8
3	82.4	2.59	74.5	8.3
4	86.7	3.06	69.4	8.7
5	84.2	3.28	75.2	8.5
6	82.6	2.78	77.1	7.9
7	88.4	2.66	72.9	8.2
8	84.6	2.83	68.4	7.8
9	86.2	3.25	72.6	8.6
10	87.1	2.74	68.9	8.7

By analyzing the data in the above table, it can be seen that in 10 groups of experiments, the co-stationing accuracy of the virtual machine co-stationing method studied in this paper is higher than that of the traditional co-stationing method, and the co-stationing error detection rate of the method studied in this paper is lower than 3.5%, which is far lower than that of the traditional method. The sum of the co-stationarity and mis-detection rate of the present method is equal to 1, while the sum of the traditional method's co-stationarity and mis-detection rate is less than 1, which indicates that the traditional method still has virtual machines that have not been co-stationarity. After further processing of the values in the above table, the co-standing positive rate of the proposed method is about 13.46% higher than that of the traditional method, and the co-standing effect is better.

When targeting virtual machines with different number of accounts, the virtual machine coverage pairs using the same resident method are shown in the following Figure.



(a) This paper's method

(b)Traditional method

Figure 2. Compares the coverage of the resident method

By analyzing the curve in the figure above, it can be seen that the co-resident coverage of the method in this paper is higher than that of the traditional method by comparing the two figures A and B. When changing the number of user accounts on the server, with the increase of the number of virtual machine flooding, the resident coverage of the method in this paper gradually increases, and the more the number of virtual machine users, the higher the coverage. Although the same resident coverage rate of the traditional method also meets the above law, the rate of increase of coverage rate is lower than that of the method in this paper. To sum up, the virtual machine co-stationing method considering cloud computing platform studied in this paper has the advantages of high co-stationing accuracy and high coverage rate, which can reduce the protection cost and has advantages.

4. Conclusion

Cloud service providers provide firewall, IP protection and other services, also add more insurance for users' data. However, the characteristics of multi-tenant dynamic aggregation and boundary generalization make it difficult for cloud computing platform to resist the security threats brought by the sharing of computing resources between virtual machines. The virtual machine homing method considering cloud computing platform studied in this paper can enable the attacker to realize the homing of malicious virtual machine and target virtual machine quickly and reduce the protection overhead as much as possible.

5. Acknowledgments

This paper is sponsored by the program "Specialty Tutorial System" in HuaShang College of Guangdong University of Finance & Ecnomics (No.2020HSDS04)

References

[1] Tian Junfeng, Qu Xueqing, He Xinfeng, et al. The method of virtual machine live migration based on hashgraph. Journal of Electronics & Information Technology. 2020, 42(03), 712-719.

[2] Sun Zhiyong, Ji Xinsheng, You Wei, et al. A virtual machine dynamic migration method based on redundant transition. Computer Engineering. 2020,46(02), 21-27, 34.

[3] Li Longfei, Wang Jianfeng, Liu Huan, et al. Hardware supported methods of multi-VM switching and dynamic bandwidth allocation. Microelectronics & Computer. 2020, 37(01), 93-98.

[4] Hou Jie, Xue Liang, Wang Yang. Trusted migration method for virtual machine based on trusted chain. Command Control & Simulation. 2019, 41(06), 120-124.

[5] Yan Jianen, Zhang Hongli, Xu Haiyan. The method of virtual machine placement based on workload optimization. Intelligent Computer and Applications. 2019, 9(06), 178-180, 183.

[6] Zhai Ling, Shen Si, Cheng Shi-=xing. Balanced distribution and optimization of electronic information resources under cloud computing platform. Computer Simulation. 2019, 36(07), 397-400, 440.

[7] Wang Haitao, Li Zhanhuai, Zhang Xiao, et al. Virtual machine resources allocation methods based on history data. Journal of Computer Research and Development. 2019, 56(04), 779-789.

[8] Liu Weijie, Wang Li'na, Wang Danlei, et al. Virtual machine co-residency method on cloud computing platform. Journal on Communications. 2018, 39(11), 116-128.

[9] Sha Edwin H.M, Wu Ting, Zhuge Qingfeng, et al. An efficient shared in-memory file system for co-resident virtual machines. Chinese Journal of Computers. 2019, 42(04), 800-819.

[10] Zhang Jianbiao, Yang Shisong, Tu Shanshan,et al. Research on vTPCM trust management technology for cloud computing environment. Netinfo Security. 2018, (04), 9-14.