Research on Availability of Virtual Machine Hot Standby based on Double Shadow Page Tables

Zhiyun Zheng, Huiling Wang , Zhengfei Wang , Lun Li
School of Information Engineering Zhengzhou University, HennanProvince,China
iezyzheng@zzu.edu.cn

Abstract - Double Shadow Page Tables method based on Most Recently Used algorithm is presented to solve the low availability of output delay in virtual machine hot standby. This method uses check-pointing mechanism, the workflow of primary virtual machine is divided into two stages: running and synchronization stage. One of the two Shadow Page Tables is as the main table and the other is as the alternate table. During running stage of primary virtual machine, the main table is used to save the operation of the virtual machine, and then primary virtual machine uses the Most Recently Used algorithm to check out some pages and copies them in the alternate table. During synchronization stage, the previous main table realizes the synchronization of the two virtual machine s, and the previous alternate table is as the main table. Experimental results show that this method compared with the original method Remus has obvious advantages.

Index Terms –virtual machine; Availability; hot standby; Shadow Page Table; Remus

1. Introduction

At present, virtualization technology[1] has been widely used in security, load balance, server management and integration of data center system, and so on. It can make the average utilization rate of per server to increase from the traditional 5%-15% to 60%- 80%, and make sure that the operating cost of server should be reduced by 70%-80%. In virtual application of the single calculation system resources, a physical server can generate many virtual machines to provide services for users through the virtual machine monitor (VMM). A practical problem how to ensure the availability of virtual machines based on the physical server when it is down is what we need to care.

Xen, VMware, and KVM are the common virtual machine techniques. Among them, Xen is an open source VMM which is developed by the University of Cambridge Computer Laboratory, operating systems must be explicitly modified to run on Xen, which makes Xen to achieve a high performance virtualization without special hardware and to obtain wide supporting of the virtualization field. It has been integrated into the operating systems like NetBSD, Plan9, GNU/ Linux and FreeBSD[2]. The research of this paper is based on Xen.

In order to reduce the losses caused by servers' failure or unplanned broken, we usually use "Primary-Backup" mechanism. "Primary- Backup" mechanism can reduce the probability of system's failure and improve the availability of virtual machine through creating a backup virtual machine. Virtual machine hot standby is a realization method of "Primary-Backup" mechanism. There are three ways of virtual machines hot standby based on Xen: (1) Continuous storage, shared hard disk, event driven, trigger synchronization of the two virtual machines when read/write files[3]. This method can complete the process of synchronization by suspending primary virtual machine (VM) for millisecond level time. The availability of primary VM is high, but the quantity of transmission data is large in this method. (2) "Pre-copy" mechanism based on network attached storage uses fixed time intervals and shares Page Table (SPT)[4,5,6]. This method needs to transmit smaller quantity of data when synchronizes primary VM with backup VM, but it causes frequent shadow page errors and more delay time to response the user. (3) Stop VM and copy snapshot image files to backup VM. This method is simple, but the size of snapshot image file is the same as the size of memory with time, it can cause primary VM to transfer more and more dirty pages to backup VM and reduce the availability of primary VM [7].

In order to improve the availability of virtual machines during its hot standby, this paper put forward a way to realize zero downtime based on double shadow page tables, one of the double shadow page tables is as main table and the other is as standby table, the workflow of primary virtual machine is divided into two stages: running and synchronization stage. During running stage of primary VM, main table is used to save the operation of primary VM, later primary VM uses the Most Recently Used algorithm to check out the suitable pages and save them in standby table; During the synchronization stage, the previous main table is used to realize the synchronization of the two VMs, and the previous standby table turns to be main table, which can avoid the pause during the synchronization of the two VMs. Compared with the original program Remus in Xen4.0, experimental results show that Zero Downtime based on Double Shadow Page Tables mechanism can reduce more response delay time for user's service, especially in the case of frequent changes in memory of primary VM.

In Section II, we will introduce the key technologies of virtual machine hot standby. In Section III, we will analysis and design the way that realize zero downtime based on double shadow page tables. In Section IV we will present a case study on the way. Section V concludes and presents directions for the future work.

2. The Technology of Virtual Machine Hot Standby

Virtual machine hot standby, as one of the common methods of fault-tolerant systems, uses the "Primary-Backup"

mechanism to guarantee the critical applications' to work normally which are running in the virtual machine when it is crash. The technology of virtual machine hot standby requires a backup VM which is running a backup server, and records dirty page list using shadow page in primary server, copies primary VM's change state to backup VM for synchronization according to the list, to keep the consistency of primary VM and its backup VM.

2.1. Remus

Versions before Xen4.0 had no scheme for high availability. Citrix released Xen4.0 on 2010.4.7, which achieved the aim of virtual machine hot standby. The project Remus[8] was issued by the Computer Technology Institute of British Columbia University, which was built on the "Live Migration" function of Xen, and was according to save the backup of real-time update to provide high availability for the virtual machines running on Xen. The architecture of Remus is shown as Figure 1.

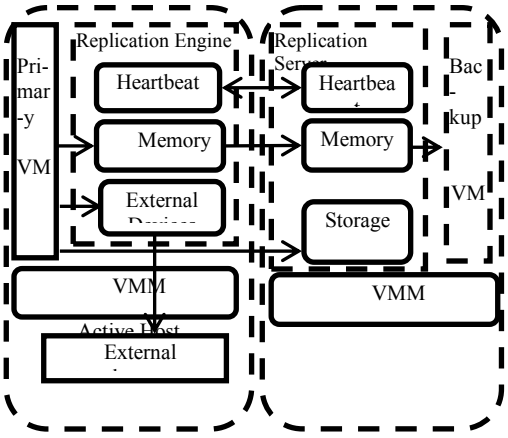

Fig.1 Architecture of Remus

As shown in Figure1, primary VM and backup VM have their own image files and connect through network, the primary VM provides external services, VMM transfers primary VM's status to backup VM in the rapid replication technology to synchronize them. When primary VM is failure, backup VM can immediately take over the master of external provision of services which is completely transparent to users.

The hot standby of Remus use shadow page table to record the status of primary VM's operation.

2.2. Shadow Page Table

Virtual machines based on Xen use Shadow Page Table(SPT) to translate the virtual addresses to the machine addresses, and mainly use for the real-time copy of the page table in the Guest OS. When VM is running, the hypervisor program puts the pointer which points to the most advanced of the shadow page tables (physical address) to the host's page table based on address register (CR3). If VM updates the pages of page table, the pages are called dirty pages. During synchronization stage, primary VM just transfers the dirty pages to backup VM, which can reduce the repetition rate and the number of transmission data. There are two methods on Xen to update the shadow page table: out-of-sync and emulated write. Emulated write is simpler than out-of-sync on the step of updating the shadow page table, and Remus uses emulated write to locate the dirty pages. The principle of location is as follows[9]:

1) Pages of Guest OS page table are set read-only and mapped to the shadow page table.
2) The operation of modifying Guest OS pages triggers page protection fault and passes them to Xen.
3) Xen directly parses the update of Guest OS, and completes the update for Guest OS, and then, synchronizes the shadow page table, finally records the pages in dirty page bitmap.

According the mechanism of shadow page table, dirty pages are tracked and recorded to the dirty page bitmap, which provides an important basis for the status synchronization between primary VM and backup VM.

2.3. Application of the Shadow Page Table in Remus

Remus is one of application which uses "Primary-Backup" mechanism to achieve virtual machine hot standby, and uses "stop-copy" method to synchronize the status between primary VM and backup VM. The workflow of Remus is shown as Figure 2.

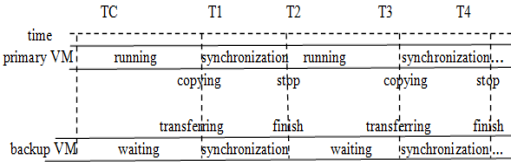

Fig.2 Workflow of Remus

T0-T2 is one work-cycle of primary VM, it includes two stages: running and synchronization stage. During running stage, as T0-T1 of the Figure 2, shadow page table records the running status of primary VM. T1 is the checkpoint, Remus begins to synchronize the primary VM and backup VM at this moment. T1-T2 is synchronization stage, during the stage primary VM suspends running and outputs to users is buffered, to transfer the dirty pages which are generated after the last checkpoint to backup VM. At the moment T2, the operation of backup is finished, backup VM sends received message to primary VM, and then primary VM empties the shadow page table and goes on running from T1 to provide services for users. The next running stage, primary VM rebuilds the shadow page table to save the status of it.

Remus had been achieved hot standby when VM is running, and improved the availability of VM, but there are still two drawbacks: (1)because of program locality principle ,the pages last time used may be used again in next time, while Remus empties shadow page table after checkpoint which will make frequently missing page error during the next running stage, which partly reduces the availability of VM.(2)During the frequently synchronization stages, primary VM delays for user's service, the delay time is unacceptable for the demand of real-time applications.

3. Zero Downtime based on Double Shadow Page Tables

To solve frequent downtime and the long time of shadow page table's rebuilding of Remus, the thinking of virtual machine dynamically migrates iterative transmission [10] is used, this paper draws Zero Downtime based on Double Shadow Page Tables (ZDDSPT)method. ZDDSPT defines double shadow page tables, one as main table which is used to record the status of primary VM during its running stage, and the other as standby table is used to save the MRU pages of main table to reduce the time of main table's rebuilding time. Define the double tables as A and B, and the initial main table is A, standby table is B .The workflow of ZDDSPT is shown in Figure 3.

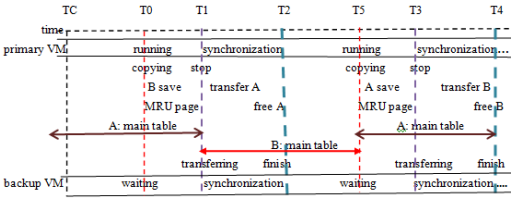

Fig.3 Route chart of ZDDSPT

In Figure 3,TC-T4 is one work-cycle of ZDDSPT including four stages:

TC-T1 is the first running stage, A as main table, B as standby table, and T1 is the checkpoint.

T1-T2 is the first synchronization stage, B changes to be main table, and A as standby table, T2 as the checkpoint.

T2-T3 is the second running stage, B still as main table, A as standby table, T3 as the checkpoint.

T3-T4 is the second synchronization stage, A changes to be main table, and B as standby table, T4 as the checkpoint.

During one work-cycle of ZDDSPT, the double tables alternate as main table and standby table. In running stage, main table records the operation status of primary VM, then chooses and transfers some MRU pages from it to standby table; In synchronization stage, primary VM doesn't pause running, standby table changes to be main table and main table is used to synchronize the primary VM with backup VM.

Section 3.1 and 3.2 will describe the running and synchronization stages of ZDDSPT in detail.

3.1. Running Stage

The first running stage, as TC-T1 in Figure 3, can be divided into two sub-periods: TC-T0 and T0-T1.The period TC-T0 is normal running stage, and the period T0-T1 is pre-copy stage.

During TC-T0, A as main table is used to save dirty pages of primary VM which are generated after the moment TC, and B as standby table is free. During the pre-copy stage T0-T1, because of program locality principle, the history pages may be reused in the next time, to cut down the time of the shadow page table's rebuilding, ZDDSPT uses the Most Recently Used algorithm to check out the suitable pages and saves them in standby table; During the whole running stage TC-T1, backup VM is kept waiting to receive primary VM to send dirty pages to keep its synchronization.

3.2. Synchronization Stage

During the first synchronization stage, primary VM sets pages of main table as "read-only" status, and transfers the pages to backup VM to synchronize primary with backup VM. Table B turned to be main table, meanwhile, pages of B are set "allowed to write" status. Primary VM doesn't pause running, and begin to access B, B instead of A service for it. Because B had saved MRU pages of A during the pre-copy stage, compared to Remus, primary VM will cut down the time of page table's rebuilding in synchronization stage, only if the pages which are accessed by applications are not in B, ZDDSPT will create a new page record in B.

At the moment T2, synchronization finished, pages of A has been copied in table A, primary VM gets message from backup VM and cleans up A, A is free, primary VM begin to the next "running-synchronization" stage.

The second running stage (T2-T3), just as the first running stage TC-T1, the difference is that B instead of A as main table to record operation status of primary VM, the MRU pages of the B are copied in A during pre-copying stage.

The second synchronization stage (T3-T4), just as the first synchronization stage T1-T2, the difference is that A as main table to record operation status of primary VM, B is used to synchronize the status of primary VM with backup VM, synchronization finished at T4, then primary VM empty B, and B is free.

In summary, shadow page table A and B alternate as main shadow page table to record the operating status of primary VM, ZDDSPT achieves zero downtime and low latency, makes sense to improve the availability of primary VM.

4. Experimental and Result Analysis

To verify superiority of ZDDSPT in memory changes frequently of VM, this paper has done two sets of experiments to compare performance of primary VM in hot standby between ZDDSPT and Remus, under different memory changes frequency and different checkpoint intervals.

4.1 The experimental environment and steps

Hardware environment: 2 HP servers, 100MB Ethernet, 1TB serial hot-swap hard disk.

Software environment: ubuntu8.04 OS, Xen4.0.1, import the same image file server in virtual machine.

The indicators of experiment is the response time of primary virtual machine to users' services, according the command remus-no-net [VM name] [backup server IP] to establish virtual machine hot standby, and test response time of primary VM for user services on memory changes in different frequency and different checkpoint intervals.

Experimental steps:

Run the command xm create [VM name] on primary server to set two VMs, and run different application programs on the two VMs.

Run the command remus-i [time interval][VM name][backup server IP] on primary server, Modify Remus checkpoint intervals as 200ms\100ms\50ms\25ms.

View Remus's log files and record the response time on different checkpoint intervals.

View Remus's log files and record the response time on memory changes in different frequency.

Repeat testing based on ZDDSPT on different checkpoint intervals and memory changes in different frequency.

4.2 Experimental Result

Run 1000 tests for different memory changes in different frequency and intervals, and get average response time as shown in Figure 4 and Table 1.

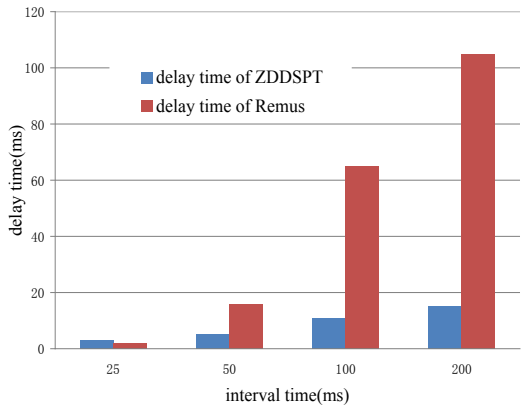

Fig.4 Output delay based on different intervals

Figure 4 shows that the comparison of different response time of primary VM service for users in different time intervals based on the two hot standby methods, Remus and ZDDSPT. Compared with Remus, the response delay time to user services based on ZDDSPT during the hot standby of VMs is reduced by 77.23% in average, which the highest and lowest is 85.71% and 68.75%. The checkpoint intervals are impacted largely to ZDDSPT, the reduced proportion of delay time increases with the interval.

Table 1 shows the selected ten comparisons of response time in different memory dirty pages based on ZDDSPT and Remus.

TABLE1 Experiment data that output delay based on different memory dirty pages.

the count of dirty pages	delay time of ZDDSPT	delay time of Remus
50	2	9
128	4	12
256	5	19
310	6	38
400	8	53
512	10	88
632	13	92
745	16	98
854	20	100
1024	25	108

From table 1, comparison of the response times in different memory dirty pages based on the two methods of hot standby is shown. Figure 5 shows that response delay time of user services based on ZDDSPT is reduced by 81.47% in average based on memory changes in different frequency, which the highest and lowest is 88.64% and 73.68%. It is shown that ZDDSPT improves the availability of primary VM larger than Remus on more memory dirty pages.

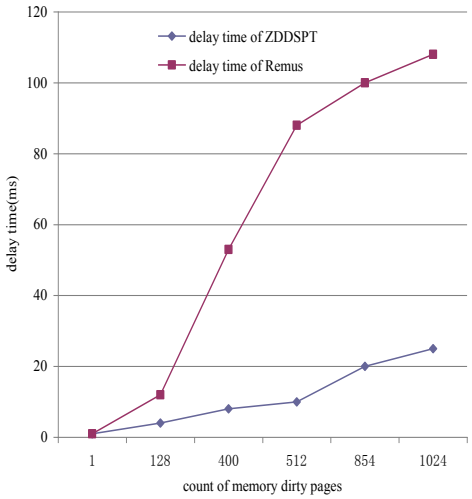

Fig.5 Output delay based on different memory dirty pages

5. Conclusions

Zero Downtime based on Double Shadow Page Table (ZDDSPT) method is presented to solve low availability caused by the copy technology in output delay during the virtual machine hot standby, it improves the availability of primary VM based on Remus. According the comparison experimental of response time in different intervals and memory dirty pages based on ZDDSPT and Remus, the result shows that the availability of primary VM improves to 81.47% in average based on ZDDSPT in the case of more memory dirty pages. It's a very meaningful work that improves the availability of virtual machine hot standby based on ZDDSPT, the next research is to study the impact of ZDDSPT on synchronization of primary VM and backup VM, and how to improve the performance of the primary VM based on this research.

References

[1] Dong Yao-zu, Zhou Zheng-wei, Chen Kang. System Virtual Machine Technology and Application based on X86 Framework [J] .Computer Engineering, 2006, 32(13): 71-73.

[2] Liu Qi-cheng, Zheng Wei-min. The Application of Virtualization Technology In Disaster Recovery System[J].Journal of Chinese Computer Systems, 2010, 10(5):1.

[3] Tamura Y, Sato K, Kihara S, et al. Kemari: Virtual Machine Synchronization for Fault Tolerance [C] // Proc. of USENIX Annual Technical Conference. Boston, USA: USENIX Press, 2008.

[4] Taiji. http:// net.pku.edu.cn/vc/files/ft/index.html,2011.

[5] Lu Mao-hua, Chiueh T.Fast Memory State Synchronization for Virtualization-Based Fault Tolerance[C] //Proc.39th Ann. IEEE/IFIP Int'l Conf. Dependable Systems and Networks (DSN '09), 2009.

[6] J. Zhu, W. Dong, Z. Jiang, et al. Improving the Performance of Hypervisor-Based Fault Tolerance [C]//Proc.24thIEEE Int'1 Parallel and Distributed Processing Symp.(IPDPS '10),2010.

[7] Liu Hai-kun, Jin Hai. Optimize Performance of Virtual Machine Check-pointing via Memory Exclusion[C] //Proc.of the 4th China Grid Annual Conference. Yantai, China: IEEE Press, 2009.

[8] B. Cully, G. Lefebvre. Remus: High Availability via Asynchronous Virtual Machine Replication[C]//Proc. Fifth USENIX Symp. Networked Systems Design and Implementation (NSDI'08), 2008.

[9] Zheng Zhi-yun, Ren Zhen-fang, Li Dun, et al. Research on Availability of Virtual Machine Hot Standby Based on Pre-transfer [J].Computer Engineering, 2012,38(15).

[10] Jun Zhu, Zhe Fu-jiang. Optimizing the Performance of Virtual Machine Synchronization for Fault Tolerance [C] //IEEE TRANSACT- IONS ON COMP- UTERS, VOL. 60, NO. 12, DECEMBER 2011.