Fix page fault in post-copy live migration with RemotePF page table assistant

Zhongyuan Shan, Jianzhong Qiao, Shukuan Lin
Northeastern University
School of Computer Science and Engineering
Shenyang, China
Shan52@163.com, qiaojianzhong@mail.neu.edu.cn, linshukuan@mail.neu.edu.cn

***Abstract*—While using post-copy algorithm for virtual machine (VM) live migration, guest operating system access to a not-yet-transferred memory page will cause a page fault(PF). Destination host should send page request to source host to get the missing page. If the VM is not being migrated, Xen hypervisor will fix the PF by itself. But, the page fault information does not contain the VMs migration status. So, Xen has to deliver all the page faults, to an independent page fault handler unit, whether it is caused by post-copy live migration or not. Let the monitor do the judgment and make decision. This will reduce the VM performance especially while the VM at non-migration status. In this paper, we design and implement a new Xen page table assistant named as RemotePF(Remote Page Fault). RemotePF is used to identify the VM migration status. So, Xen could skip the page fault handle unit and get the VM migration status directly from RemotePF. We implemented the post-copy live migration algorithm and RemotePF page table assistant (PTA) for Xen. The experimental results show that RemotePF can accurately identify the VM migration status, reduce page faults fixing time, and has little influence on the normal running VM.**

***Keywords*-live migration; post-copy; page fault; Xen; page table assistant**

I. Introduction

While using post-copy live migration algorithm to relocate VM to another physical host, the VM on destination host will be resumed right after the entire VM core data are completely transferred. At this moment, the VM memory is incomplete and waiting to be transferred. The guest operation system (GuestOS) accesses to these not-yet-transferred memory pages will lead to page fault and make itself be hung up. The page faults occurred at this time can be divided into two type, those caused by accessing an already transferred page and those caused by accessing a not-yet-transferred page. For the first type, Xen needs to call its own page fault fix function. For the second type, it is necessary to send a page request to source host to acquire the missed page. But at this point, Xen cannot get enough information to identify the page fault type. The information that Xen could gather is from the structure cpu_user_regs and structure vcpu. Xen can get error_code from cpu_user_regs, and the CR2 linear address of missing page in structure vcpu. But there is no information on VM migration status. Error_code identifies the trigger factor of page fault. It seems to be a good way to identify the VM migration status. But actually, error_code is assigned automatically by GuestOS when page fault occurs. Because of the virtualization, GuestOS is neither to know whether it works under a virtual environment, nor to perceive whether it was being migrating. So GuestOS could not identify the migration status into error_code. So, Xen need to transfer all page fault to the page fault handler in post-copy driver to make the judgment. For the VM being migrated, it is a good solution to catch and fix the remote page faults. But for the VM in normal running, the behavior is obviously unnecessary and may lead to a reduction in GuestOS efficiency. VM is running in the normal state in almost 99.99% of the time. Even if each judgement takes only a few clock cycles, the addition of these times will add up to be a non-negligible number. So, It is necessary to find a way to solve the problem. Page table assistant is the best way to judge the type of page fault in Xen. The page table assistant determines the page table operation logic. Whenever an operation related to a page table is performed, Xen will call the corresponding manipulation function according to the settings in page table assistant. At present, Xen implements two kinds of page table assistants, shadow and HAP. The working status of page table assistant is stored in the domain structure which contents the basic information of VM. It could be found in the structure member domain.arch.paging.mode. When page fault occurs, Xen firstly gets the current working vcpu. Xen can locate the page fault domain and get domain basic information from structure domain, and the status of the page table assistant, according to which, Xen determines the correct processing flow for page fault fixing. Therefore, the page table assistant can be used to identify the status of the virtual machine migration, and Xen can get the current migration status directly from the page table assistant without additional operations when the page fault occurs. This will greatly simplify the page fault fixing process. In this paper, we designs and implements the RemotePF page table assistant. During the post-copy migration of guest VM, domain0 will enable RemotePF on guest VM when it is to be resumed on the destination host, and disable it when the migration is complete. Xen can get the migration status of the guest VM directly through RemotePF PTA. When RemotePF is enabled, Xen delivers the page fault to post-copy page fault handler. If the RemotePF is disabled, Xen will directly fix the page fault by itself. RemotePF on guest VM is controlled by domain0 through hypercall. We implement the RemotePF page table assistant in our post-copy migration mechanism for Xen. The experiment shows that RemotePF can accurately identify the page fault while VM is being migrated, and

978-1-5386-7445-1/18/$31.00 ©2018 IEEE
DOI 10.1109/DCABES.2018.00020

when the VM runs normally, RemotePF enabled Xen to quickly fix page faults, and has little effect on the page faults fix time.

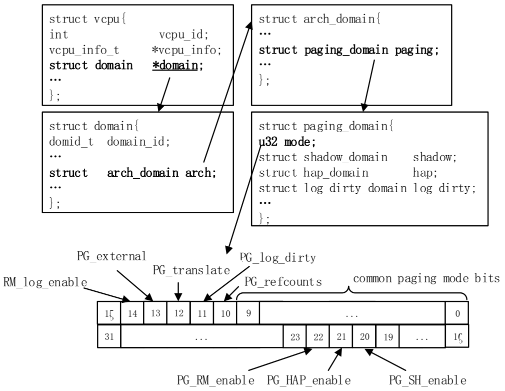
II. Related Work

VM live migration algorithm is developed from the process live migration. Currently, the most popular migration algorithm are pre-copy [1] and post-copy [2]. Post-copy has short downtime, low network overhead, and low residual dependency [3]. So it's attracted a lot of researchers' attention. Reference [4] implemented a remote page fault filter for Post-copy Live Migration. If the page fault is caused by guests overwriting operations, it would not cause remote page request. Reference [5] utilizes the Asynchronous Page Fault feature of KVM to let VM could keep running without temporary pause when page faults occur. Reference [6] proposed an agile live migration algorithm with a hybrid of pre/post-copy. It transferred only the working set pages at migration time.Reference [7] present several techniques to improve Migration efficiency of memory intensive applications on both post-copy and pre-copy.Reference [8] executes post-copy and pre-copy with High Speed Optical Network to experiment how network delay impact on VM live migration

III. The design of RemotePF PTA

A. *RemotePF status flag*

The status of the page table assistant is saved in the structure member vcpu-¿domain.arch.paging.mode. Mode is a 32 bit unsigned integer variable, where each bit is defined to represent the working status of page table assistant by Xen. A schematic diagram of the function of each mode bit is shown in Figure 1. The marked bits that have been used are as follows. The 20th bit of mode represents whether shadow page is enabled. The 21th bit represents that hardware assisted paging (HAP) is enabled. The 10th bit PG_refcounts identifies that Xen is counting the number of visits to the shadow page table. 11th bit PG_log_dirty identifies whether the shadow page table records dirty pages. The 12th bit PG_translate identifies that the transformation of P2M table will be finished by Xen instead of the guest domain. The 13th bit PG_external indicates whether the current guest domain works in external mode. There are also many flags still in the idle state that can be used by RemotePF PTA to indicate its working status. We use the 22th bit PG_RM_enable to identify the operative mode of RemotePF PTA, and use the 14th bit RM_log_enable identifies whether the page fault will be recorded while RemotePF PTA is enabled. The migrated VM will be restored as DomainU on the destination host by Domain0. So domain0 has a control interface for domainU to modify the working status of RemotePF PTA. After the VM migration started, Domain0 firstly receives the basic information of the migrated VM (vmconfig), then create a blank virtual machine DomainU based on the vmconfig settings. Next, the core data of VM will be transferred and restored into DomainU. So, DomainU has all the context of VM except memory pages and could be resumed. Before resuming DomainU, Domain0 will enable the RemotePF PTA in DomainU by hypercall. Xen can be informed by the working status of RemotePF PTA that DomainU is being migrated and the memory pages have not been transferred completely. All the page faults will be transferred to the page fault fix function of the migration algorithm to solve. When all the memory pages of DomainU are migrated, Domain0 uses hypercall to disable RemotePF PTA. Xen will then directly call its own page fault fix function to solve the page fault.The operation flow of Domain0 to the RemotePF PTA in DomainU is shown in Figure 2.



```
struct vcpu{
int                vcpu_id;
vcpu_info_t      *vcpu_info;
struct domain    *domain;
…
};
```

```
struct domain{
domid_t  domain_id;
…
struct   arch_domain arch;
…
};
```

```
struct arch_domain{
…
struct paging_domain paging;
…
};
```

```
struct paging_domain{
u32 mode;
struct shadow_domain    shadow;
struct hap_domain       hap;
struct log_dirty_domain log_dirty;
…
};
```

Figure 1. the function of each flag bit in domain.arch.paging.mode

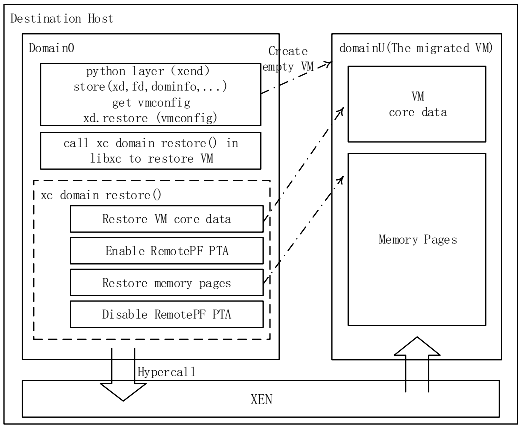


Figure 2. the function of each flag bit in domain.arch.paging.mode

B. *Hypercall control RemotePF*

Domain0 needs to use hypercall to notify Xen, and let Xen to modify the working status of RemotePF P-TA in the migrated VM DomainU. For the extensibility of the system, we do not take up a new hypercall number, but choose to expand the function of hypercall __HYPERVISOR_domctl with hypercall number 36 to complete the related operations of RemotePF PTA. First, add a new member XEN_DOMCTL_remote_op into Xen_domctl structure which is the parameter of hypercall __HYPERVISOR_domctl. and bind XEN_DOMCTL_remote_op with the operation number 65. Then we define the structure of Xen_domctl_ remote_op in the union Xen_domctl.u. The structure is shown in figure4. OP represents the operation of RemotePF PTA, including enable, disable and log operation. Log_address stands for the start address of the log save location, and MB represents the size of the log space. Currently, log is mainly used to record and analyze the working status of the RemotePF PTA in the experiment. The function can be extended to assist for function extension in future versions. The do_domctl() function of Xen is responsible for answering the hypercall __HYPERVISOR_domctl and calling the corresponding processing function according to the op value.

C. The working flow of fixing paging fault with RemotePF enabled

When a page fault occurs, Xen's own page fault fix unit is executed. The do_page_fault() function gets the address of missing memory page from vcpu CR2 register, then call the page fault fix function fixup_page_fault(). According to the current vcpu information, this function can obtain the structure domain of the page fault occurs VM, and further more obtain the domain.arch.paging.mode member variable that identifies the working status of page table assistant. Xen must identify the type of page faults to decide whether it is processed by itself or transferred to the page fault fix module of the migration algorithm. Therefore, Xen should first check the PG_RM_enable identifier in the mode variable member to determine whether the current VM is in the migration state. If the PG_RM_enable flag is true, it means the current VM is in the post-copy migration and the transfer of memory pages is not finished, yet. It is necessary to transfer the page fault to the page fault fix unit of the migration algorithm. If PG_RM_enable is false, It can go directly to the Xen normal page fault fixing process. Because Xen working in ring0 and cannot identify which memory pages have been migrated, It needs to be judged by the page fault fix unit of post-copy migration algorithm and decide whether to send a page request to the source host. The page fault fix unit reads the PFN number of the missing page from page table according to the address in CR2, and determine whether the missing page has been migrated according to the received page records which is maintained by Domain0. If the missing page has been migrated to the destination host, so this is a common page fault. Fix union in post-copy will return it back to Xen to finish the fix work. If the missing page has not yet been migrated, it needs to send a page request to source host and get the memory page with corresponding PFN number. After getting the missing page, fix unit converts the memory page PFN to MFN according to the P2M table, and calls function xc_map_foreign_page() to map the corresponding memory page from VM. The content of the received page will be written to the corresponding address. Finally, fix unit modifies the present flag in the page table entry, and also the PFN in address field of PTE will be converted to MFN. The fixing result is fed back to the Xen page fault fixing unit. If the page fault fixing unit of the migration algorithm successfully fixed the page fault, the entire process ended. If the missing page does not exist even on source host, the empty page is returned. The Xen page fault fixing unit will handle it as a normal page fault. If an exception occurs, the exception code is returned, and Xen exception handling unit will complete the rest of the work.

IV. Evaluation

In this paper, we implemente the post-copy migration algorithm and RemotePF page table assistant On Xen with version 4.1.2. In order to observe the impact of the RemotePF PTA on the efficiency of Xen fixing page fault, we respectively enabled RemotePF PTA on normal running VM and the post-copy migrating VM, recorded the time Xen takes to deal with a page fault, and made a comparation with the situation that VM without RemotePF PTA. The physical hosts used to build a post-copy migration environment have Intel core2 Duo E8500 CPU, 4GB memory. The operation system is CentOS5.6, with Linux core version 3.1.2. Two physical machine are used as source host and destination host in post-copy VM live migration respectively. And the third one is used as the NFS server to save physical disk of the migrated VM. The three hosts are connected by routers to form a local area network with a network bandwidth of 100M. Three hosts are connected via local area network using a router. The network bandwidth is 100M. Because Xen needs to use microsecond timer to calculate the time spent in handling page faults. Computation and write test results into logfile may influence the test result. In order to minimize the impact, the experiment will take 100 page fault as a statistical cycle. Every time a page fault occurs, it will just record the start and end time. When page fault number reaches 100, calculate and incorporate into log file. (1) The influence of RemotePF PTA on page fault fix time with VM under non migration state. Users can do all kinds of operations on the test host to give rise to page faults, such as checking system information, reading and writing files, running programs, etc. Record the time it takes to deal with page fault on original Xen VM, post-copy enabled (Remote off) VM and RemotePF PTA enable VM. The result is shown in Figure 3. The experiment results show that, after post-copy algorithm is implemented, the fixing time of page fault on Xen VM will increase, even if the VM is under a non-migration state. This is because Xen need to call an additional service function to determine the VM migration state. And when RemotePF PTA is enabled, Xen can directly learn the VM migration status, and omit the service function. This will greatly reduced the page fault fix time . Compared with the Xen VM without RemotePF PTA, the fix time

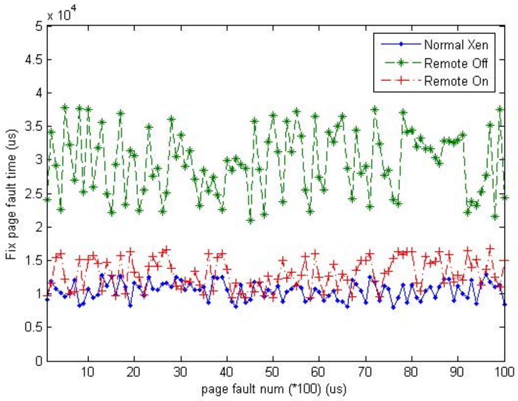


Figure 3. page fault fix time while domain running normally

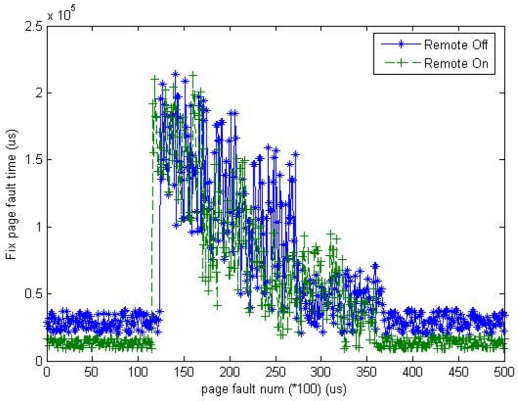


Figure 4. page fault fixing time while domain is being migrated

was reduced by nearly 60%, almost consistent with the original Xen VM. Just because Xen need to read the PTA status, the time is slightly increased. From the experiment, we can see that Implement post copy algorithm on Xen will increase the page fault fix time and RemotePF PTA can solve this problem effectively. (2) The influence of RemotePF PTA on page fault fix time with VM being migtated. In the experiment, three hosts are used to form the migration environment, and the post-copy algorithm is used to complete the VM migration. The physical memory test program MemTester was running on the migrated VM. MemTester is a program for memory reading and writing test for Linux system. it is mainly used to give rise to page fault in this experiment. Respectively migrate the VM to destination host with RemotePF PTA enabled and the PTA disabled. And record the time spent in fixing page fault during the migration. The results are shown in Figure 4. the experimental results show that Xen can identify the current VMs migration status through the status of the RemotePF PTA. And if there is a page fault occur while VM is in the process of pos-copy migration, Xen will send a page request to the source host to get the missing memory page. Whether or not enable RemotePF PTA has little effect on page fault fix time while VM is being migrated. This is because the remote page request takes up most of the page fault fix time while VM is being migrated, and the RemotePF PTA does not carry changes on the remote page request flow.

V. Conclusion

In this paper, we designs and implement the Remote page table assistant on Xen, with which Xen can be directly aware of whether the current VM is being migrated When page fault occurs. While the VM is normally running, Xen can omit the judgment process of monitoring program, and quickly call its own fix function to handle page fault. When the VM is being migrated, RemotePF PTA can delivery page fault to the fix function of post-copy algorithm, and send a page request to source host to acquire the missing page. We implemented the post-copy migration algorithm and RemotePF PTA on Xen. through the experiment method, It is proved that the RemotePF PTA can effectively reduce the time of fixing page faults while VM running normally.

References

[1] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, “Live migration of virtual machines,” in *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005, pp. 273–286.

[2] M. R. Hines, U. Deshpande, and K. Gopalan, “Post-copy live migration of virtual machines,” *ACM SIGOPS operating systems review*, vol. 43, no. 3, pp. 14–26, 2009.

[3] A. Zarrabi, “A generic process migration algorithm,” *International Journal of Distributed & Parallel Systems*, vol. 3, no. 5, pp. 29–37, 2012.

[4] K. Su, W. Chen, G. Li, and Z. Wang, “Rpff: A remote page-fault filter for post-copy live migration,” in *Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on*. IEEE, 2015, pp. 938–943.

[5] T. Hirofuchi, I. Yamahata, and S. Itoh, “Postcopy live migration with guest-cooperative page faults,” *Ieice Trans.inf. & Syst*, vol. 98, no. 12, pp. 2159–2167, 2015.

[6] U. Deshpande, D. Chan, T.-Y. Guh, J. Edouard, K. Gopalan, and N. Bila, “Agile live migration of virtual machines,” in *Parallel and Distributed Processing Symposium, 2016 IEEE International*. IEEE, 2016, pp. 1061–1070.

[7] A. Shribman and B. Hudzia, “Pre-copy and post-copy vm live migration for memory intensive applications,” in *European Conference on Parallel Processing*, 2012, pp. 539–547.

[8] M. I. Biswas, G. Parr, S. McClean, P. Morrow, and B. Scotney, “A practical evaluation in openstack live migration of vms using 10gb/s interfaces,” in *Service-Oriented System Engineering (SOSE), 2016 IEEE Symposium on*. IEEE, 2016, pp. 346–351.