Memory Access Techniques in Virtual Environment

Ekta, Banasthali University and Ms. Manisha Agarwal, Associate Professor, Banasthali University

Abstract—In cloud computing, Virtualization is used to make virtual image of all resources like server, network, storage etc. This helps in sharing a single resource among multiple users. The main concept is to create and handle virtual machine on existing and available operating system but the environment of new virtual machine does not have any relation with host operating system. The virtual environment is handled by an application called Hypervisor.

Keywords: *Hypervisor, Virtualization, Shadow page table, Ballooning, Smart paging, Hypervisor Swapping.*

*****

I. Introduction

A web space where data can be approached, administered and configured is Cloud Computing. In Virtualization, the resources that are idle can be combined into shared pool and allot dynamically to the users. No user is aware of that they are using the shared resource. The main advantage for using is reduction of cost of maintenance of server.[2]

Virtualization consists of 4 levels: Emulation, para virtualization, full virtualization and operating system level virtualization. [3]

1. *Emulation:* This technique helps in conversion of commands as well as instruction via software. And also kernel driver are not used by both guest as well as host. Therefore, it makes a software more secure, portable and using an emulator without its installation. [3]
2. *Para virtualization:* This method helps in execution of normal commands on CPU without any kind of translation. Commands that are risky not permitted to execute on the host system as they can down the system so, they are pre-translated to make an error free environment. [3]
3. *Full virtualization:* Hypervisor can run any no. of operating system on host machine. It provides good security to the virtual machines. [3]
4. *Operating system level virtualization:* It offers high coordination between host and guest by providing a common and highly altered kernel which works spontaneously for both i.e. host and guest at same time.

II. TYPES OF VIRTUALIZATION

1. *Hardware virtualization:* To use processors in effective manner homogeneous servers are combined to form a large server, is Hardware virtualization. [1]
2. *Network virtualization:* In network virtualization, connection between two or more virtual systems is also virtual and even physical cables are modified to virtual cables in which cabling time as well as its maintenance time has been fully deducted. [1]
3. *Storage virtualization:* Collection of physical storage from numerous networks converted to single storage. [1]
4. *Data virtualization:* User can easily modify the data and format the data without awaring of the technical knowledge, the data modification takes place to actual location. [1]

III. Advantages

1. Resources can be easily managed as it hides the hardware and other network resources in a single host environment. [10]
2. The cloud computing security is achieved by protecting the integrity on guest machine and components of cloud are available on demands of the guest, also provide reliability. [10]
3. The resources are shared among many guests, utilized pooled resources and balanced workload.
4. Operating System for Guest can be Ubuntu, Red Hat, Windows 7, Windows 8 etc.

IV. Hypervisor

A hypervisor is a process that behaves like a Virtual Machine Manager for guest operating system i.e. manages the virtual machine. Two types of hypervisor:

1. *Type 1*: This type of virtual machine contains no host operating system hence virtual machine is installed on bare system. It is known as Full virtualization as it contains only hardware on which it is running. Examples: Oracle Virtual machine, RTS Hypervisor, LynxSecure etc. [11]

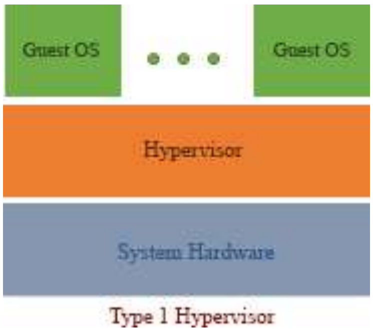


Figure: 1(a) Hypervisor Type 1 [11]

2. *Type2*: The hypervisor contains its installation on host operating system is known as hosted or type2 virtual machine. [11]

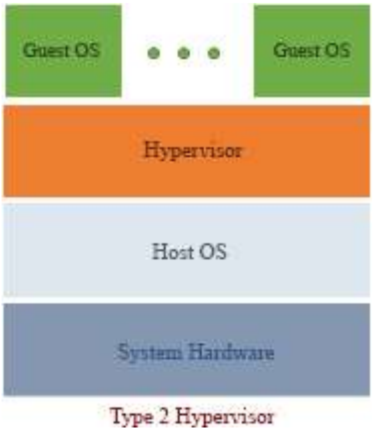


Figure: 1(a) Hypervisor Type 1 [11]

V. PAGE TABLE

Page Table in an operating system is like a data structure used for storing the mapping between physical and virtual addressing. Processes that are large in size are segmented to small units known as pages. Every process in the CPU has its own page table, therefore no need to provide process identifier to page table. A page table entry has 4 bytes of size which can store not only address but also information like valid/invalid bit, access type (read/write) etc. Data is in page frames and is easily scattered in overall primary memory. Pages are mapped in a manner that is easy to access. Demand paging is also featured. Swap the pages in an efficient manner and also swap the pages that are used very rare. To store the page table is the issue while designing page table. [13]

VI. WHY SHADOW PAGE TABLE

Shadow page table is the replica of page which contains mapping between virtual address and physical address. The concept of shadow page table is to keep an eye on state of guest page table. As guest operating system is different from host operating so the guest user is not permitted to contact host/hardware page table otherwise the guest will supervise the host. However, hypervisor has the actual mapping table, so hypervisor presents the shadow page table to the guest. Guest is not aware about the page table they are accessing, is shadow page table not actual. [4]

The mapping to guest page table with shadow page table is not immediately synced as guest page table cannot contact for update to hypervisor spontaneously. For these two approaches are followed: [5]

1. *Naïve Approach:* The operating system can modify its page table i.e. response to a page fault. The hypervisor accepts only the privileged instructions. Now, hypervisor can make relevant changes to the shadow page table, i.e. substitute the guest address with relevant host address. [5]
2. *Fallback Approach:* In this hypervisor has write protected to guest page tables. Operating system cannot modify the page table; if modification is done then page fault occurs. This page fault is called tracing fault. So, hypervisor can spontaneously modify its shadow page table. But this approach is used when having large amount of page fault. [5]

VII. PAGE TABLE ENTRY IN SHADOW PAGE TABLE

Shadow page table consists of 2 things: GVA (Guest Virtual Address) and HPA (Host Physical Address). Guest page table consists of GVA and GPA (Guest Physical Address). HVA refers to Host Virtual Address and KVM is Kernel Virtual Machine. [9]

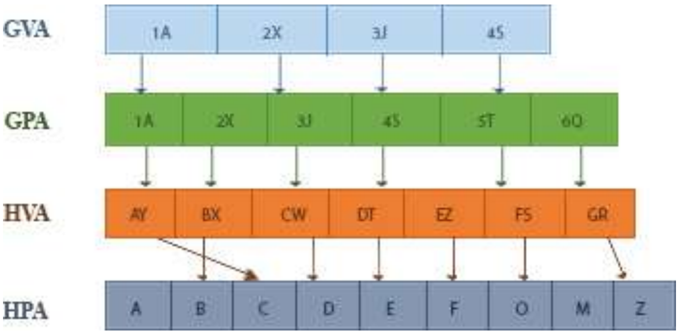


Figure: 2(a) Shadow page table mapping [9]

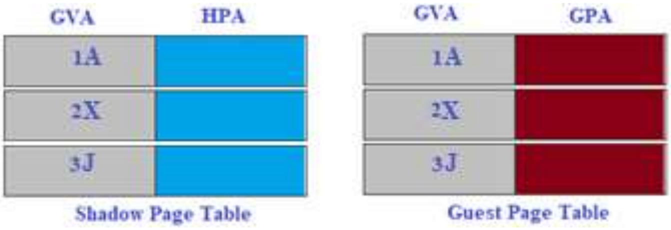


Figure: 2(b): Shadow page table [9]

Guest user first crested a linear mapping for the processes. Following are the steps:

Step 1: In Guest Operating System, Guest first maps the GVA with GPA, if any page fault found then Virtual Machine has to exit. [9]

GVA->GPA

Step 2: Later on, the mapping between GPA to HVA is done and it is possible because of the KVM (Kernel Virtual Machine) as KVM is already aware of mapping from GPA to HVA.[9]

GPA->HVA

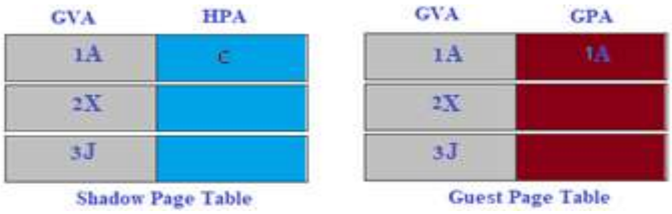
Step 3: KVM finds the HPA for every corresponding HVA.[9]

HVA->HPA

Step 4: Now KVM updates the Shadow page table and also updates the guest page table.[9]

After mapping of Shadow page table with Guest page table, it may result like this: [9]

GVA->GPA->HVA->HPA



GVA	HPA
1A	c
2X	
3J	

Shadow Page Table

GVA	GPA
1A	1A
2X	
3J	

Guest Page Table

Figure: 2(c) Shadow page table entry [9]

VIII. Advantages

1. Log record writing overhead is reduced. [14]
2. Fastest recovery. [14]

IX. Disadvantages

1. For every page table of guest, an updated Shadow page table is required.
2. A consistent shadow page table required for both host as well as guest operating system.

X. Page fault handling in shadow page table

Following is the flowchart to handle page fault:

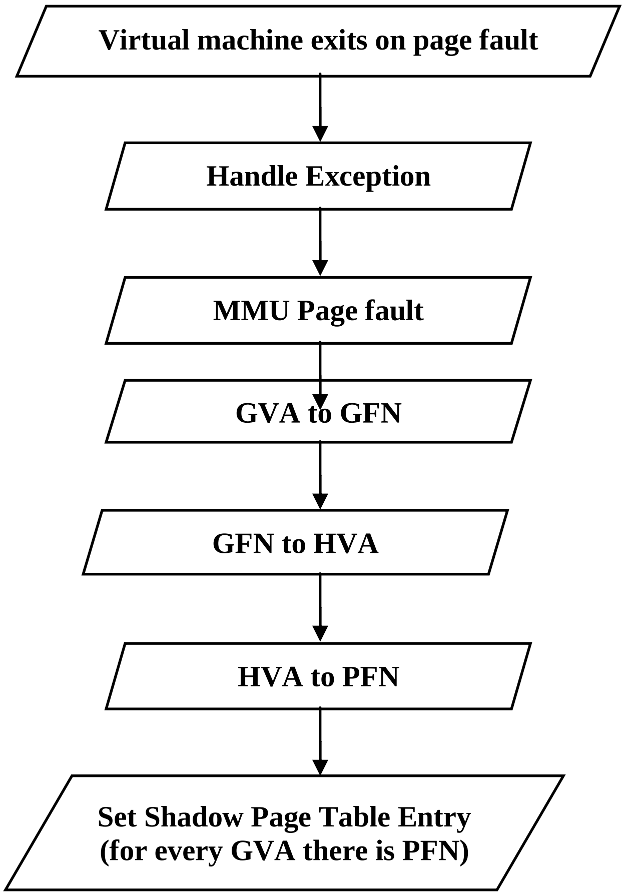

Virtual machine exits on page fault
Handle Exception
MMU Page fault
GVA to GFN
GFN to HVA
HVA to PFN
Set Shadow Page Table Entry
(for every GVA there is PFN)

Figure: 3 Flowchart to handle page fault [8]

Step 1: When any page fault occurs in the virtual machine, it exits. [8]
Step 2: On occurrence of page fault, it passes the exception to handle i.e. to Hypervisor. [8]
Step 3: Passes the error code of page fault in virtual machine i.e. GVA. [8]
Step 4: Conversion between GVA to GFN takes place as Host machine aware of only GFN (Guest Frame Number) not GVA. [8]

- Page Table Entry(PTE) contains code of CR3 (Control Register 3) [8]
- GFN will have only 1-20 bits of PTE i.e. Base
- PT_INDEX will have 1-10 bits of GVA [8]

NOTE: PT_INDEX is Page Table Index

Step 5: Now, conversion between GFN to HVA takes place [8]

- GFN value must be lies between memory_slot -> Base_GFN and memory_slot -> Base_GFN + memory_slot -> npages (total no. of pages) then HVA is memory_slot -> userspace_address + [(GFN-memory_slot-> Base_GFN) << 12] [8]

*Step 6:*Now, convert HVA with corresponding HFN (Host Frame Number). Here a new page is inserted in the memory if only required. [8]
Step 7: Setting the entry to Shadow Page Table for GFN/GVA to HFN. [8]

XI. MEMORY PAGING TECHNIQUES

Memory paging techniques are preferred by Hypervisor to improve the performance of virtual environment. The paging techniques that can be used are:

1. *Smart Paging:* Smart paging ensures that virtual machine will not burst due to insufficient memory, but this can reduce the overall performance of the virtual machine. There is no available memory at the time of restarting the virtual machine so; it is used when a virtual machine restarts. It is the concept which is used in very less cases. It automatically frees the memory after 10 minutes of restart of the virtual machine. Smart paging file has by default pre-defined area on the server disk executing Hyper-V. This causes the server suffer from issues like performance degraded. To avoid this problem, it is required to reconfigure the storage space of smart paging file on a non-system disk. [7]
2. *Ballooning:* Hypervisor can able to manage and execute multiple guest Operating Systems simultaneously. Due to which memory gives low performance and also no guest are ready to free the utilizing memory. This results in shortage of host memory. Guest machine are running independent of each other, so, they are unaware that they are working on virtual machines. Therefore, this technique realizes the guest operating system about the shortage of memory of host. The driver of balloon refers to hypervisor using private channel. By this contact, if guest requires memory, then hypervisor helps to assign memory pressure from host machine to guest machine. This method is intelligent; it decides which memory page is important or less important. The less important page is then swapped to the disk with no hypervisor. And the available memory contains only required memory page. This method depends on guest operating system to response for allocation of memory. Because of this it is very time consuming process for memory reclamation from host system. [6][12]
3. *Hypervisor Swapping:* When ballooning is not much efficient to reclaim the memory, Hypervisor reclaims it. Ballooning techniques require time to reclaim memory from host. It guarantees to reclaim specified amount of memory for specified time. A separated swap file is created for virtual machine of extension (*.vswp).* This file is created inside the virtual machine folder on host system by default. Hypervisor had no information regarding which virtual machine memory pages are used or unused, so it is difficult for hypervisor to swap out those unused pages. Therefore, this technique reclaims the specified amount of memory, so that the swapped out pages can be swapped in back to memory.[7]

XII. COMPARISON BETWEEN MEMORY PAGING TECHNIQUES

Smart Paging	**Ballooning**	**Hypervisor Swapping**
Memory is utilized when virtual machine restarts and stayed for 10 min. after restart.	Memory is utilized till VM is active. During restart of VM this technique is not used like smart paging.	Memory is utilized till VM is active. During restart of VM this technique is not used like smart paging.
Hard disk utilization is temporary. Done to enhance the shortage of memory.	Hard disk utilization is permanent.	Hard disk utilization is permanent.
Virtual machine performance degrades as hard disk speed is slower than memory access speed.	Performance is degraded if multiple guest machine uses the memory simultaneously which increases need for virtual memory then host memory.	Swap space is created on physical memory to store the memory pages. Due to which complexity arises. When guest OS access the memory then swapping is done between virtual to real physical pages. It leads to performance issues.
No memory is reclaimed neither other virtual machines nor from host machine.	Memory is reclaimed when virtual memory utilization is excessive.	The physical memory of guest is swapped by hypervisor to the swap space. And the host physical memory is available to other guests.

XIII. SUMMARY

Shadow page table entry must be consistent and updated according to the guest page table entry; this can lead to less page faults. The above memory paging techniques should be handled as well controlled very attentively so that the performance issues may not arise. The guest able to continues his work without interruption or exiting of virtual machine

REFERENCES

[1] https://www.javatpoint.com/virtualization-in-cloud-computing

[2] Santosh Kumar and R. H. Goudar "Cloud Computing – Research Issues, Challenges, Architecture, Platforms and Applications: A Survey" *International Journal of Future Computer and Communication, Vol. 1, No. 4, December 2012*

[3] Durairaj. M and Kannan.P "A Study on Virtualization Techniques and Challenges in Cloud Computing" *International Journal of Scientific& Technology Research Vol. 3, Issue 11, NOVEMBER 2014*

[4] https://recalll.co/app/?q=virtualization%20-%20What%20exactly%20do%20shadow%20page%20tables%20(for%20VMMs)%20do%3F%20-%20Stack%20Overflow

[5] https://corensic.wordpress.com/2011/12/05/virtual-machines-virtualizing-virtual-memory/

[6] http://deinoscloud.wordpress.com/2010/09/30/ballooning-and-hypervisor-swapping-common-misunderstandings/

[7] http://searchservervirtualization.techtarget.com/feature/memory-paging-techniques-hypervisors-use-to-improve-VM-performance

[8] https://www.cse.iitb.ac.in/synerg/lib/exe/fetch.php?media=public:students:shashank:kvm_shashank_kernelgiri.pdf

[9] https://www.cse.iitb.ac.in/~puru/courses/autumn12/cs695/classes/kvm-overview.pdf

[10] http://milner.com/company/blog/technology/2015/07/14/the-advantages-and-disadvantages-of-virtualization

[11] http://cloudsearch.blogspot.in/2016/01/understanding-abstraction-and.html

[12] https://turbonomic.com/blog/on-technology/memory-101-the-challenges-of-managing-memory-in-a-virtual-environment/

[13] https://www.slideshare.net/guestff64339/implementation-of-page-table

[14] https://dbms-ii.blogspot.in/2010/03/shadow-paging.html