An Approach To Graphics Passthrough In Cloud Virtual Machines

Sunit Parmar
Department of Information Technology
G H Patel College of Engineering
Gujarat Technological University
Vallabh Vidyanagar - 388 120
Gujarat, India.

Aniruddh Kurtkoti
Department of Computer Engineering
A D Patel Institute of Technology
Gujarat Technological University
Vitthal Udyognagar - 388 121
Gujarat, India.

Abstract

Growing need of cloud computing services makes it more challenging field for cloud providers and researchers. After a brief introduction of Xen hypervisor paper discusses the problem of inefficient access to graphics accelerators in cloud. The reason behind such problem is proprietary models and heterogeneity in architecture of graphics processors. This paper focuses on such problem of GPU (Graphical Processing Unit) acceleration by providing a sample GPU passthrough model. At the last we will discuss the advantages of such technique and its future.

1. Introduction

To understand graphics passthrough ones should understand the concept of GPU and its working. To start with GPU is a graphics processing unit a chip used to accelerate graphics in a computer system. GPU is generally available on a VGA (Video graphics adapter) in typical computer hardware. At present many high quality graphics cards available in market to provide customers or users advance graphics which they can use for entertainment and development purpose. Now if we want to provide same sort of quality graphics services, then there is requirement of virtualization of graphics devices. So the graphics passthrough is such a technique to achieve graphics virtualization for cloud users.

As mentioned earlier new generation clients are using cloud for various purposes. So it is always a challenge for cloud developers to fulfil the needs of users. Quality graphics for gamming and other entertainment purpose or even for development is such a novel requirement of users of cloud market.

The only way to fulfil such requirement is graphics virtualization. But there is a large list of problems in achieving such milestone for cloud business. Because GPU's architecture is complex. Another reason is GPU developers never reveal methodology of GPU, and they keep upgrading the architecture for better performance [2].

The problem of virtualizing graphics becomes worst when we think of recent virtualized cloud environment. Vendors of graphics adapters follow proprietary models. So virtualizing such adapters requires separate logic dedicated to various models specifically. It also complicates the task of sharing graphical resources between two or more VMs [3].

2. Xen Hypervisor

Xen was actually developed as an open source research project at Cambridge University. Now it is a proprietary of Citrix organization but continuous to be open source. Citrix now provide all the modification done to xen code by its users and also supports various researches on xen. So now xen is a good option to continue research in area of cloud and virtualization [1].

In xen as it is bare metal hypervisor the initial operating system will be booted as "Domain 0". Domain 0 operating system is a normal linux that communicates with host system hardware directly. While the guest systems which are running on xen can communicate to
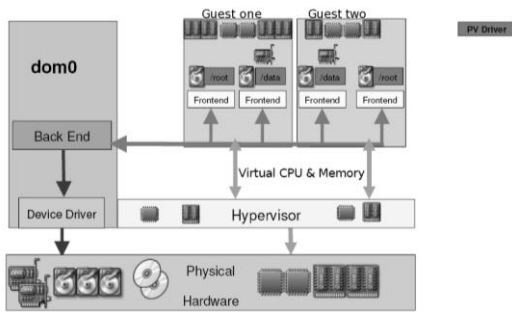
Figure 1. Xen Paravirtualization Architecture

physical hardware via xen domain 0.

2.1. Xen Memory Layout

It is basically required to understand Xen memory layout including its physical host and guest systems to gain knowledge of passthrough. Figure 2 shows a typical memory layout of machine running Xen. A communication area in terms memory address range is reserved along with BIOS, generally used by NICs and disk controllers.
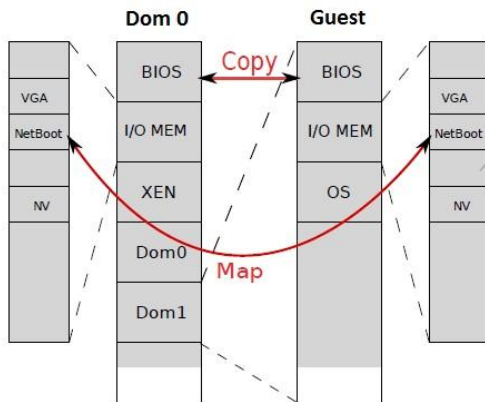

Figure 2. Xen memory usage layout

After that Xen and its guest domains are placed. Guest domains have a normal layout of memory same as a typical computer system. For a guest domain a memory layout means simply an address range starting with 0,whatever translation require is done by physical host and Xen. Same way BIOS of guest is replica of original BIOS.

If we think of PCI passthrough, some areas of physical host are mapped to guest as shown in figure. When the VMs guest operating system tries to access such areas by writing/reading them, Xen confirms the address

access by translating the address. Result is only single VM is a candidate for accessing such area, others will be isolated.

2.2. Graphics Vs PCI Passthrough

If we compare VGA adapters with other PCI devices graphics adapters are complicated ones. Reason behind such complexity is VGA adapters have to take care of many x86 features like I/O ports, text mode, memory ranges, VESA modes, VGA BIOS etc. It simply means more code, more logic and more complication than PCI passthrough in a system [4].
A GPU passthrough includes various complex steps based on adapter model like disabling internal adapter, mapping of VGA BIOS, emulation of the same. It is also required to passthrough all I/O ports connecting guest to original physical adapter of the host system [3].

3. Graphics Acceleration

Graphics processing unit is a special chip available on VGA cards which performs various graphics related operations as instructed by the operating system. Earlier we already discussed that graphics cards are extremely complicated devices due to their secretive vendor dependent architecture. This section describes a sincere approach to achieve graphics passthrough in specific VGA card.
In the approach we have used citrix xen server 5.6.1 version having kernel version xen- 4.2 or above. It is also required to have one or more guest domain configured with guest operating system windows XP service pack 3. As graphics virtualization is still far from generalized scheme. This scheme roughly dictates four generalized steps to achieve passthrough for any VGA card.
Figure 3 shows general steps required in achieving gpu virtualization.
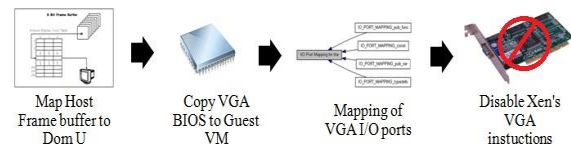

Figure 3. Graphics passthrough process

3.1. Map host frame buffer to DomU

Very first step in implementing passthrough for VGA is to map its frame buffer to guest domain running in

Xen. The addresses 0xA0000 to 0xC000 are standard range to which frame buffer is generally mapped. To get access to VGA card guest's memory should be mapped with such a range, which will be useful for guest operating system to detect the VGA. Normally such locations are hardcoded for various operating systems. Two files which are configured to achieve this task are /libxc/xc hvm build.c and /libxc/xc linux.c

3.2. Replicate VGA BIOS to VM

Same way like frame buffer VGA BIOS also have a fixed memory location and its address range in general case is 0xC0000 to 0xCBFFF.VGA adapter control is handled by a number of functions provided by BIOS, this BIOS is first copied to Dom U and then executed in this step. Configuration of tools/ioemu/hw/pc.c will results in initialization of graphics adapter with updated BIOS.

3.3. Mapping of VGA I/O ports

VGA adapters establish communication with operating systems using I/O ports. By writing the required instruction to appropriate address for VGA BIOS, one can communicate with VGA. Because typically VGA adapter is hard coded to read values from fix locations and perform operation based on the value. So such mapping important to allow communication of guest operating system request is done in this step.

3.4. Immobilize Xen's VGA instructions

Last but very crucial step is to stop xen's domain 0 from interrupting anymore in VGA adapter's communication when it is requested by a guest domain. The step allows fully functional video support to guest running in xen.

4. Results

This section includes results taken using two benchmark tools for graphics testing. Tools used for testing are TechPowerUp GPU-Z a video card information tool and Performance test provided by passmark softwares.
Snapshots in figure 4 and 5 shows poor graphics response by guests while running benchmark tests using testing tools.
In figure 4 we can see GPUz utility is not showing any graphics detection in a guest domain running in xen. Similarly figure 5 show output of performance test tool having a warning: "Nothing will be rendered".
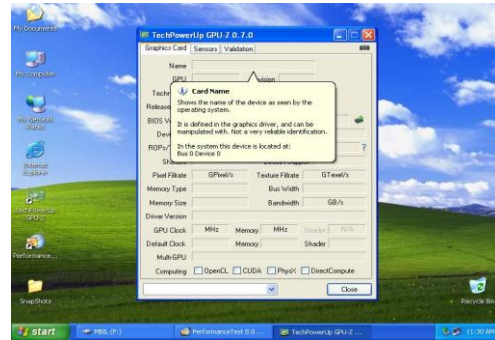
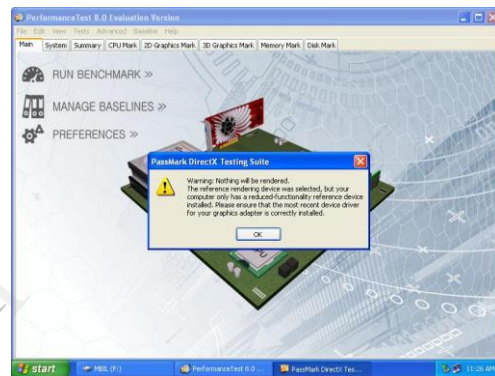

Figure 4. Gpu-z before passthrough
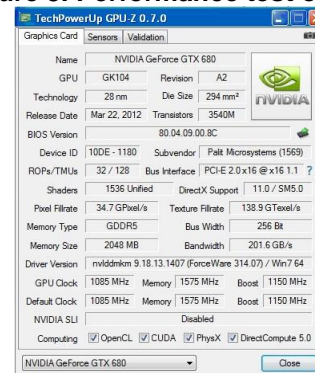

Figure 5. Performance test output

Figure 6. Gpu-z after passthrough

Snapshots in figure 6 and 7 show testing result of same tools after implementation of GPU passthrough. Test results of GPUz in figure 6 shows detection of graphics card including its name and BIOS version. Same way snapshot in figure 7 shows performance test running without any error.
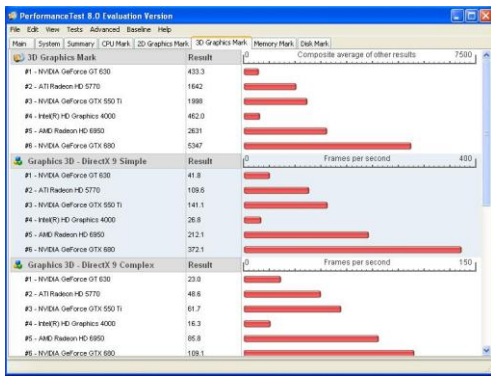
Figure 7. Performance test after passthrough

5. Benefits of Passthrough Model

Cloud environment can gain significant advantages by implementing graphics virtualization.GPU passthrough results in following plus points for cloud [2].
VMs can render high quality graphics. CAD/CAM applications can be available remotely for clients of cloud. Cloud user can gain access to high quality graphics at low cost. Games demanding 3D rendering can be available on cloud. Remotely accessible high range graphics can be available at lower rates. Cloud can be loaded with low cost graphics services.

6. Limitations

Multiple graphics card sharing and acceleration is still an issue, because the technique is extracting BIOS of host system such operation is complicated and hardware specific. Currently the patches and steps are working well with Intel chips, but they may not prove efficient with other hardware.

7. Future

Because HPC benefits attracting more and more people to use Cloud, and because of the need of high graphics applications like Maya, AutoCAD, High Graphics 3D Games, it make sense that acceleration is required at hardware level in Graphics adapters. The accomplished implementation work is a sincere effort to virtualize NVDIA graphics adapter which opens doors for enhancing the current work to achieve GPU acceleration for graphics adapters from other vendors. All the manufacturers of graphics adapters are possessive about their models, it is the right time to do research in area of virtualization enabled flexible graphics for cloud.

8. Conclusion

Graphics passthrough is found a delicate task due to secretive architecture of graphics adapters specifically NVDIA. Accomplishment of graphics virtualization via gpu passthrough enables cloud for providing quality graphics services at very low cost. As discussed earlier in the paper graphics card vendors are secretive in disclosing the architecture of graphics devices, this makes graphics acceleration specific to the graphics vendor.

9. References

[1] Sarfraz Nawaz Brohi,Mervat Adib Bamiah "Exploit of Open Source Hypervisors for Managing the Virtual Machines on Cloud" (IJAEST) INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES Vol No. 9, Issue No. 1, 055 – 060.
[2] Micah Dowty, Jeremy Sugerman," GPU Virtualization on VMware's Hosted I/O Architecture" USENIX, SIGOPS OPER. SYST. REV., VOL. 43, NO. 3. (JULY 2009), PP. 73-82.
[3] Vishakha Gupta, Ada Gavrilovska,Karsten Schwan, Harshvardhan Kharche, Niraj Tolia, Vanish Talwar,Parthasarathy Ranganathan "GViM: GPU-ACCELERATED VIRTUAL MACHINES," 3RD WORKSHOP ON SYSTEM-LEVEL VIRTUALIZATION FOR HIGH PERFORMANCE COMPUTING (HPCVIRT'09) 31 MARCH 2009, 17-24.
[4] S.-T. Shen, S.-Y. Lee, and C.-H. Chen,"FULL SYSTEM SIMULATION WITH QEMU: AN APPROACH TO MULTI-VIEW 3D GPU DESIGN", 2010 IEEE INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, MAY 2010, PP. 3877–3880.
[5] Jean Guyader \[PATCH] "PASS-THROUGH A GRAPHIC CARD" http://article.gmane.org/gmane.comp. emulators. xen.devel/51194, Xen-devel mailinglist.
[6] Todd Deshane,Muli Ben-Yehuda,Amit Shah,Balaji Rao"Quantitative Comparison of Xen and KVM"Xen Summit, June 23-24, 2008, Boston, MA, USA.
[7] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris,Alex Ho, Rolf Neugebauery, Ian Pratt, Andrew Wareld," XEN AND THE ART OF VIRTUALIZATION", SOSP'03, OCTOBER 19–22, 2003, BOLTON LANDING, ACM 1-58113-757 -5/03/0010.