

# Research and Application of GPU Pass-Through Based on KVM in Graphics Workstation

**Zhuangli Dong<sup>1, a</sup>, Zhengbin Wang<sup>1, 2, b</sup> and Jinkun Yuan<sup>3, c</sup>**

<sup>1</sup>College of Electronic and Optical Engineering & College of Microelectronics, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

<sup>2</sup>Nanjing University (Suzhou) High-tech Institute, Suzhou 215123, China

<sup>3</sup>Engineer, Nanjing Astute Software Technology Co. Ltd

<sup>a</sup>dong191217@163.com, <sup>b</sup>wangzb@njupt.edu.cn, <sup>c</sup>yuan.jinkun@astute-tec.com

**Keywords:** KVM/QEMU, vfio-pci, PCoIP, pass-through.

**Abstract:** Desktop cloud has become a research focus of cloud computing. As 3D rendering, scientific computing and large data fields on the requirements of high performance computing, end user take more and more high request to the cloud desktop. Yet in the desktop cloud, the simulation of virtual graphics can only be used to display 2D graphics, but not to accelerate graphics hardware. This paper mainly discusses KVM/QEMU PCI device passthrough so as to use GPU in the virtual machine, and use the PCoIP Zero Client at the terminal to meet desktop cloud latency and bandwidth requirements. In the case graphics workstation would be greatly enhance the virtual desktop performance, graphics designers can run graphics intensive applications from PCoIP zero Client. The pass-through technique involved in this article is vfio-pci.

## 1. Introduction

Desktop cloud is a typical cloud computing application which can provide remote computing services to users in the cloud. It has many advantages, such as providing a freely access to the desktop, reducing the management and maintenance costs of hardware and software. But as with other technologies, desktop cloud has its drawbacks. Traditional desktop virtualization is only suitable for the release of office automation, financial applications and 2D light-weight graphics applications while not widely applied in graphic intensive applications.

Aiming at the existing shortcoming of the desktop cloud claimed above, this paper introduces a method to implement GPU passthrough, as well as the experiment and summary.

## 2. Related Technology

### 2.1 VFIO-PCI

Currently there are two types of graphics processing in desktop cloud, one is using virtual graphics card, the other is using physical graphics card. In this paper, we take the second approach, which is also called GPU pass-through. Through the GPU pass-through technology to bind the physical display card to the virtual table and directly assign the physical display card to a virtual machine, which gives the virtual machine the same graphics performance as the real physical mainframe. And to achieve the best end-user experience with the PCoIP zero client at the terminal.

The pass-through technology used in this paper is vfio-pci, vfio is a Linux kernel infrastructure that allows to leverage the capabilities of modern IOMMUs to drive a device directly from user space without any additional specialized kernel driver being involved.

## 2.2 PCoIP

PCoIP, also known as “PC over IP”, is a high-performance display protocol purpose-built to deliver virtual desktops. PCoIP technology uses advanced display compression to provide end users with cloud-based virtual machines as a convenient alternative to local computers.

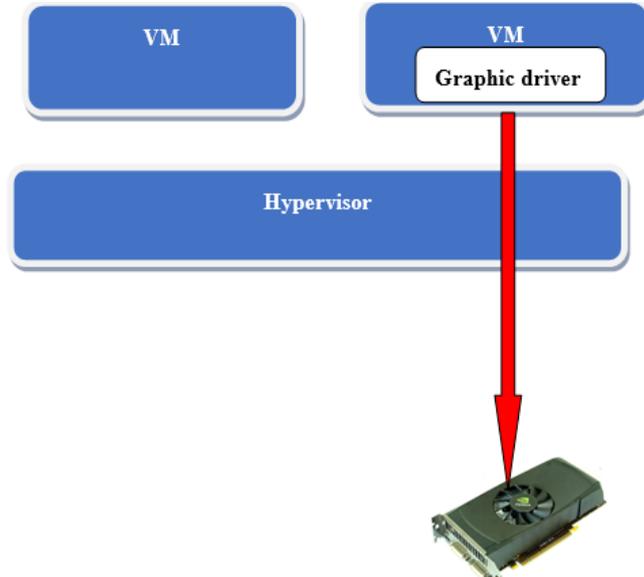


Fig. 1 GPU Pass-through Mode

## 2.3 Requirement

Hardware Information:

GPU: NVIDIA Quadro M4000

CPU: Intel(R) Xeon(R) CPU E5-2683 v3 @ 2.00GHz

MEM: 32G

SSD: 480G

OS: CentOS7.3

## 2.4 Implementation

- ① In order to assign devices in KVM, you'll need a system which supports Virtualization Technology for Direct I/O(VT-d).
- ② Recompile the kernel and add a VFIO\_VGA drive.  
CONFIG\_VFIO\_PCI\_VGA=y
- ③ Add follow parameters to grub. conf .  
intel\_iommu =on  
i915.enable\_hd\_vgaarb=1  
modprobe. Blacklist =nouveau

Notes:

Remember to update the gurb profile after edit. After restart, verify that the CentOS system's native Nouveau driver has been disabled.

```
$ lsmod | grep nouveau
```

Blacklist the opensource nouveau module

```
$ echo "blacklist nouveau" >> /etc/modprobe.d/blacklist.conf
```

If vfio-pci was built as a module,you need to add this to your bootloader

```
$ echo"optionsvfio_iommu_type1allow_unsafe_interrupts=1" >/etc/modprobe.d/vfio_iommu_type1.conf
```

```
$ echo "options kvm ignore_msrs=1" >> /etc/modprobe.d/kvm.conf
```

- ④ Check the VGA information you're going to pass-through, and unbind it from the default drive in

the host to make it available for virtual machine

```
$ lspci |grep NVIDIA
```

⑤ Bind vfio driver to PCI device

```
$ echo 0000:04:00.0 > /sys/bus/pci/devices/0000\:04\:00.0/driver/unbind
$ echo 0000:04:00.1 > /sys/bus/pci/devices/0000\:04\:00.1/driver/unbind
$ echo 10de:13f1 > /sys/bus/pci/drivers/vfio-pci/new_id
$ echo 10de:0fbb > /sys/bus/pci/drivers/vfio-pci/new_id
```

⑥ View the information of the unbind graphics card

```
$ lspci -s 04:00.0 -v
04:00.0 VGA compatible controller: NVIDIA Corporation GM204GL [Quadro M4000] (rev
a1) (prog-if 00 [VGA controller])
Kernel driver in use: vfio-pci
Kernel modules: nouveau
$ lspci -s 04:00.1 -v
04:00.1 Audio device: NVIDIA Corporation GM204 High Definition Audio Controller (rev
a1)
Subsystem: NVIDIA Corporation Device 1153
Kernel driver in use: vfio-pci
Kernel modules: snd_hda_intel
```

Notes:

If the Kernel driver in use: vfio-pci. Means VGA and vfio-pci are successfully binded.

⑦ Create bridge

⑧ Build start vm script

```
#!/bin/bash
modprobe vfio-pci
QEMU_AUDIO_DRV=alsa \
QEMU_AUDIO_TIMER_PERIOD=0 \
/usr/libexec/qemu-kvm -cpu host,kvm=off -no-user-config -nodefaults -m 4096M \
-smp 2, sockets=1, cores=2, threads=1 -boot menu=on \
-vnc 0.0.0.0:20 -k en-us \
-vga none \
-bios /usr/share/seabios/bios.bin \
-drive
file=/var/lib/nova/gpu/win10.img,if=none,id=drive-virtio-disk0,format=qcow2,cache=none,
werror=stop,rerror=stop,aio=native \
-device
virtio-blk-pci,scsi=off,bus=pcie.0,addr=0x7,drive=drive-virtio-disk0,id=virtio-disk0,bootind
ex=1 \
-M q35 \
-net nic,model=virtio,macaddr=00:55:aa:00:00:01 -net bridge,br=br-test \
-monitor unix:/tmp/qmp-sock,server,nowait \
-daemonize \
-device ioh3420, bus=pcie.0,addr=1c.0,multifunction=on,port=1,chassis=1,id=root.1 \
-device vfio-pci,host=01:00.0,multifunction=on,x-vga=on \
-device vfio-pci,host=01:00.1 \ - -device vfio-pci,host=05:00.0 \
- device vfio-pci,host=05:00.0 \
- device vfio-pci,host=05:00.1 \
- device vfio-pci,host=05:00.2 \
```

Notes:

kvm=off will hide the kvm hypervisor signature, this is required for NVIDIA cards  
x-vga=on is required for vga assignment

-vga none disables the default vga device on QEMU, it is also required for vga assignment.

⑨ Boot vm and start

### 3. Result

Download the GPU-Z software in the virtual machine tested, our results are displayed in Figure 2. At the same time, we also conduct design software tests, such as keyshot5.0, Auto CAD2016, Maya. We may safely draw the conclusion: The virtual machine will be able to meet various requirements of design, animation and video applications on the graphics workstation.

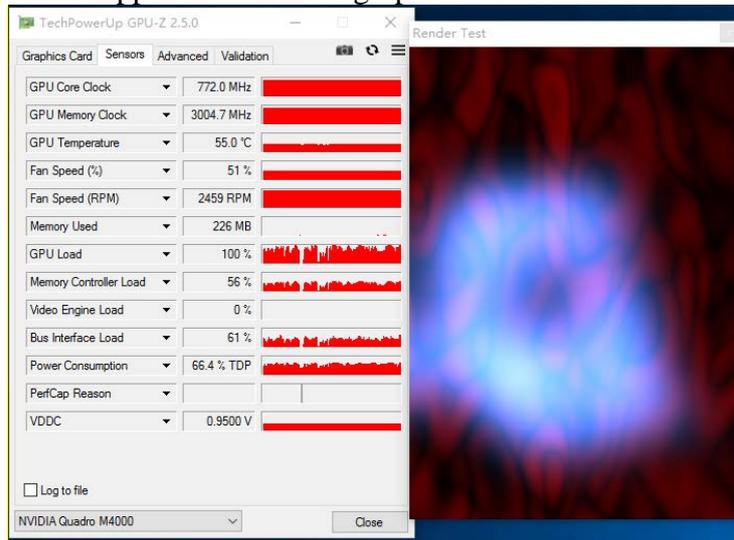


Fig. 2 GPU-z

### 4. Summary

At the moment, the virtualization is not adopt to professional design industry mainly in technology aspects because the virtual desktop cannot satisfy the requirements of the designer, especially on the performance of VGA. This research realizes the perfect combination of GPU's high performance and Cloud's efficiency and convenience, which enable the design staff to connect to the virtual desktop for graphics workstation both freely in time and space. But this article graphics pass-through is restricted by VGA type, which is mainly suitable for the current mainstream of independent VGA. The GPU virtualization solution will be a trend and all of these needs a further research.

### References

- [1] Lahabar S, Agrawal, Narayanan P J, High Permance patern recognition on GPU [C] //Proc of National conference on Computer Vision Pattern Recognition Image Processing and Graphics,2008, pp.154-159.
- [2] KVM. Kernel Based Virtual Machine [EB/ OL]. [2013-11-05]. [http:// www. linux-kvm. Org](http://www.linux-kvm.Org).
- [3] Managing KVM deployments with Virt-Manager Michael J. Hammel January 2011 Linux Journal: Volume 2011 Issue 201, January 2011.
- [4] Virtualization with KVM, Irfan Habib , February 2008 Linux Journal: Volume 2008 Issue 166, February 2008.
- [5] Keith Adams and Ole Agesen. A Comparison of Software and Hardware Techniques for x86 Virtualization. In Proceedings of ASPLOS '06, October 2006.
- [6] Gupta V, Gavrilovska A, Schwan K, et al. Gvim: Gpu-accelerated virtual machines. In: Proc of ACM Workshop on System-level Virtualization for High Performance Computing. New York, 2009.