

KVM & VGA Passthrough

- Objet : Virtualisation de Windows via KVM avec utilisation d'une carte graphique dédiée
- Niveau requis :
[avisé](#)
- Commentaires : *Pour jouer...*

Avertissement

Tuto pour qemu 2.8 et ovmf 0~20170911.5dfba97c-1. Chez moi, mettre à jour ces paquets casse la virtualisation. Mais peut-être que si les paquets sont à jour avant installation de windows 7 (ou ultérieur), il n'y a pas de problème.

Pré-Installation

Identification de la CG

```
lspci
```

```
...
01:00.0 VGA compatible controller: NVIDIA Corporation GP106 [GeForce GTX
1060 6GB] (rev a1)
01:00.1 Audio device: NVIDIA Corporation GP106 High Definition Audio
Controller (rev a1)
...
```

On retient **01:00.0** et **01:00.1**.

Récupération des pilotes VFIO

```
wget
'https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/latest-vir
tio/virtio-win.iso'
```

Ces pilotes seront nécessaires à l'installateur Windows pour trouver le disque dur et la carte réseau, je crois.

Création des disques

Avant l'installation

```
qemu-img create -f qcow2 /srv/vms/win7-systeme.img 60G
```

```
qemu-img create -f qcow2 /srv/vms/win7-jeux.img 100G
```

Après l'installation et la configuration

Pour pas tout se retaper en cas de pépin.

```
mv /srv/vms/win7-systeme.img /srv/vms/win7-systeme_orig.img  
qemu-img create -f qcow2 -b /srv/vms/win7-systeme_orig.img /srv/vms/win7-systeme.img
```

(Facultatif) Configuration de pulseaudio

Cette config permet au pulseaudio de l'utilisateur d'accepter les connections sur l'interface loopback. On demandera à KVM (lancé en root) de s'y connecter, ce qui évitera les ennuis de sorties audio simultanées.

Ajout de `load-module module-native-protocol-tcp auth-ip-acl=127.0.0.1` dans le fichier `~/.config/pulse/default.pa`.

(Si ce fichier n'existe pas: `cp /etc/pulse/default.pa ~/.config/pulse/`.)

Installation

Kernel Line

[/etc/default/grub](#)

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet spalsh intel_iommu=on  
modprobe.blacklist=nouveau hugepagesz=1G hugepages=8"
```

- `intel_iommu=on`

J'sais plus.

- `modprobe.blacklist=nouveau`

Pour être sûr que le device de la carte graphique n'est pas acquis par le module nouveau.

- `hugepagesz=1G hugepages=8`

Pour que la mémoire de la VM soit gérée via le mécanisme de hugepages, ici, 8×1Go.

InitRd

```
lspci -nns 01:00.0
```

```
01:00.0 VGA compatible controller [0300]: NVIDIA Corporation GP106 [GeForce GTX 1060 6GB] [10de:1c03] (rev a1)
```

```
lspci -nns 01:00.1
```

```
01:00.1 Audio device [0403]: NVIDIA Corporation GP106 High Definition Audio Controller [10de:10f1] (rev a1)
```

[/etc/initramfs-tools/modules](#)

```
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
vhost-net
pci_stub ids=10de:1c03,10de:10f1
```

Script de lancement

[start_kvm.sh](#)

```
#!/bin/bash

vmname="windows7vm"

# Cette fonction sert à définir le module vfio-pci, le module utilisé
# pour la virtualisation pci, comme driver de la CG
vfiobind()
{
    dev="$1"
    vendor=$(cat /sys/bus/pci/devices/$dev/vendor)
    device=$(cat /sys/bus/pci/devices/$dev/device)
    if [ -e /sys/bus/pci/devices/$dev/driver ]
    then
        echo $dev > /sys/bus/pci/devices/$dev/driver/unbind
    fi
    echo $vendor $device > /sys/bus/pci/drivers/vfio-pci/new_id
}

# On charge le module et on l'attribue à la carte graphique et à la
# carte son HDMI associée
modprobe vfio-pci
vfiobind 0000:01:00.0
```

```
vfio-bind 0000:01:00.1

# Juste pour être sûr...
if ps -A | grep -q $vmname; then
    echo "$vmname is already running." &
    exit 1
fi

# Les options de sorties, on garde un -vga std, histoire de passer
facilement les contrôles à la VM
# Cependant, cette carte doit être désactivée dans le gestionnaire de
périphériques de Windows
OPT_OUTPUT="-vga std -device vfio-pci,host=01:00.0,multifunction=on -
device vfio-pci,host=01:00.1"

# Copie des variables de l'UEFI
cp /usr/share/OVMF/OVMF_VARS.fd /tmp/my_vars.fd

# Utilisation de alsa ou de pulseaudio, suivant config de l'ordi...
#export QEMU_AUDIO_DRV=alsa QEMU_AUDIO_TIMER_PERIOD=0
export QEMU_AUDIO_DRV=pa QEMU_PA_SAMPLES=4096 QEMU_PA_SERVER=127.0.0.1

qemu-system-x86_64 \
    -name $vmname,process=$vmname \
    -machine type=q35,accel=kvm \
    -cpu host,kvm=off \
    -smp 3,sockets=1,cores=3,threads=1 \
    -enable-kvm \
    -m 8G \
    -mem-path /dev/hugepages/kvm \
    -mem-prealloc \
    -balloon none \
    -rtc clock=host,base=localtime \
    -serial none \
    -parallel none \
    $OPT_OUTPUT \
    -drive
if=pflash,format=raw,readonly,file=/usr/share/OVMF/OVMF_CODE.fd \
    -drive if=pflash,format=raw,file=/tmp/my_vars.fd \
    -device virtio-scsi-pci,id=scsi \
    -drive id=disk0,if=virtio,cache=none,format=qcow2,file=/srv/vms/win7-
systeme.img \
    -drive id=disk1,if=virtio,cache=none,format=qcow2,file=/srv/vms/win7-
jeux.img \
    -netdev type=tap,id=net0,ifname=tap0,vhost=on \
    -device virtio-net-pci,netdev=net0,mac=00:16:3e:00:01:01 \
    -drive file=/srv/data/images_cd/virtio-
win.iso,id=virtio-cd,format=raw,if=none \
    -device ide-cd,bus=ide.1,drive=virtio-cd \
```

```
-soundhw ac97  
  
exit 0
```

Utilisation

```
./start_kvm.sh
```

Sources

- <http://vfio.blogspot.fr/2015/05/vfio-gpu-how-to-series-part-1-hardware.html>
- <http://vfio.blogspot.fr/2015/05/vfio-gpu-how-to-series-part-2.html>
- <http://vfio.blogspot.fr/2015/05/vfio-gpu-how-to-series-part-3-host.html>
- <http://vfio.blogspot.fr/2015/05/vfio-gpu-how-to-series-part-4-our-first.html>
- <http://vfio.blogspot.fr/2015/05/vfio-gpu-how-to-series-part-5-vga-mode.html>
- autres...

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/utilisateurs:captnfab:tutos:virtualisation-vga-passthrough>



Last update: **21/01/2018 17:15**