Grafikkartenpassthrough Proxmox ab version 4.3

Du möchtest dich gerne für unsere Hilfe erkenntlich zeigen . Gerne. Wir bedanken uns bei dir für deine Spende!

Spenden

Zum frei verfügbaren Apt-Repository

GITLAB:

Einleitung

Zum einen ist es wichtig hier eine VM und Grafikkarte mit UEFI Support zu haben. Getestet wurde das ganze hier mit Windows Server 2012r2 und KDEneon. Wichtig ist vorab zu sagen das diese Technologie sehr Hardware abhängig ist, hat z.B. das Motherboard die Devices in falschen oder besser gesagt unbrauchbaren Konstellationen zusammengeschlossen, wie z.B. PCIe-Slot und Netzwerknic, ist es mehr als unbrauchbar. Auch noch zu erwähnen das GPUpassthrough von Proxmox selbst nicht supportet wird. Erfolgreich getestet wurde es hier mit Supermicro. Wir gehen hier im ganzen Artikel von INTELchips aus. Lässt man den Grafikteil weg, ist das ganze hier auch für normalen PCIe Passthrough gültig, dabei wird auch kein UEFI benötigt.

Vorbereitung PVEhost

Als erstes bereiten wir den Host auf die Grafikkarte vor. Hierfür tragen wir in Grub die Iommu Unterstützung ein.

```
nano /etc/default/grub
```

ändere

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
```

in

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on"
```

Danach Grub updaten:

```
update-grub
```

Module einragen:

```
nano  /etc/modules
```

```
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
```

it will not be possible to use PCI passthrough without interrupt remapping. Device assignment will fail with a 'Failed to assign device „[device name]“ : Operation not permitted' error for users of KVM, and a 'Interrupt Remapping hardware not found, passing devices to unprivileged domains is insecure. Systems which don't support interrupt remapping:

All systems using an Intel processor and chipset that have support for Intel Virtualization Technology for Directed I/O (VT-d), but do not have support for interrupt remapping. Interrupt remapping support is provided in newer processors and chipsets. To identify if your system has support for interrupt remapping:

```
dmesg | grep ecap
```

On the IOMMU lines, the hexadecimal value after „ecap“ indicates whether interrupt remapping is supported. If the last character of this value is an 8, 9, a, b, c, d, e, or an f, interrupt remapping is supported. For example, „ecap 1000“ indicates there is no interrupt remapping support. „ecap 10207f“ indicates interrupt remapping support, as the last character is an „f“. Interrupt remapping will only be enabled if every IOMMU supports it.

Wenn das System interrupt remapping nicht unterstützen sollte, kann man unsichere interrupts erlauben. Aber davon sollte man abraten.

```
echo "options vfio_iommu_type1 allow_unsafe_interrupts=1" >
/etc/modprobe.d/iommu_unsafe_interrupts.conf
```

Von wird aber im produktivem Betrieb abgeraten.

Zu übergebende Karte vom Host abkoppeln

Da wir eine Nvidiakarte benutzen (Quadro K4000) müssen wir auch die Module in die Blacklist des Hosts eintragen, damit diese nicht automatisch genutzt wird:

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

Nach dem Neustart kann man die Garte dem Gast übergeben. Vorher prüfen wir noch die Iommu Gruppen. Damit das ganze funktioniert benötigt man eine eigenen Iommu Gruppe.

```
find /sys/kernel/iommu_groups/ -type l
```

Mit etwas Glück hängt der Slot alleine auf unserem Board. Hier hängt wie schön sieht noch Device dabei, das ist ein kleiner PCIe-Slot, in dem steckte eine Netzwerkkarte, die musste in einem anderen Slot.

```
/sys/kernel/iommu_groups/0/devices/0000:00:00.0

/sys/kernel/iommu_groups/1/devices/0000:00:01.0
/sys/kernel/iommu_groups/1/devices/0000:01:00.0
/sys/kernel/iommu_groups/1/devices/0000:01:00.1

/sys/kernel/iommu_groups/2/devices/0000:00:02.0
/sys/kernel/iommu_groups/3/devices/0000:00:03.0
/sys/kernel/iommu_groups/4/devices/0000:00:14.0
/sys/kernel/iommu_groups/5/devices/0000:00:16.0
/sys/kernel/iommu_groups/5/devices/0000:00:16.3
/sys/kernel/iommu_groups/6/devices/0000:00:19.0
/sys/kernel/iommu_groups/7/devices/0000:00:1a.0
/sys/kernel/iommu_groups/8/devices/0000:00:1b.0
/sys/kernel/iommu_groups/9/devices/0000:00:1c.0
/sys/kernel/iommu_groups/10/devices/0000:00:1c.3
/sys/kernel/iommu_groups/11/devices/0000:00:1c.5
/sys/kernel/iommu_groups/12/devices/0000:00:1c.6
/sys/kernel/iommu_groups/13/devices/0000:00:1c.7
/sys/kernel/iommu_groups/14/devices/0000:00:1d.0
/sys/kernel/iommu_groups/15/devices/0000:00:1f.0
/sys/kernel/iommu_groups/15/devices/0000:00:1f.2
/sys/kernel/iommu_groups/15/devices/0000:00:1f.3
/sys/kernel/iommu_groups/15/devices/0000:00:1f.6
/sys/kernel/iommu_groups/16/devices/0000:02:00.0
/sys/kernel/iommu_groups/17/devices/0000:03:00.0
/sys/kernel/iommu_groups/18/devices/0000:04:00.0
/sys/kernel/iommu_groups/18/devices/0000:05:03.0
/sys/kernel/iommu_groups/19/devices/0000:06:00.0
/sys/kernel/iommu_groups/20/devices/0000:07:00.0
```

Grafikkarte der VM übergeben

First, find the device and vendor id of your vga card:

```
lspci -n -s 01:00
01:00.0 0300: 10de:1381 (rev a2)
01:00.1 0403: 10de:0fbc (rev a1)
```

Hier können auch mehrer ID's verfügbar sein, meist Karten mit Audio. Wir benötigen aber nur Grafik, also:

```
echo "options vfio-pci ids=10de:1381 disable_vga=1" >
/etc/modprobe.d/vfio.conf
```

Nun fügt mal folgende Einträge zur VM hinzu:

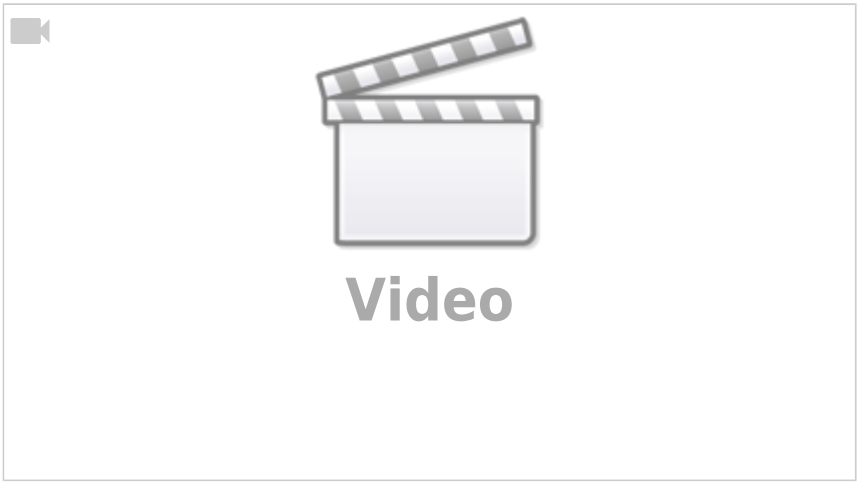
```
machine: q35
hostpci0: 01:00,pcie=1,x-vga=on
```

Bevor man das tut sollte aber sicher sein das man per Remote auf den Host zugreifen kann, also VNC und RDP, oder bei Linux Nomachine und VNC. Warum? Ganz einfach da eine Konsole in PVE dann nicht mehr verfügbar ist. Man kann auch einen Monitor direkt an der Grafikkarte anschließen, dann hat man die VM direkt auf dem Schirm. Könnte man so auch über einen HardwareKVM lösen. Hat man alle Einstellungen getroffen, Host nochmal durchstarten.

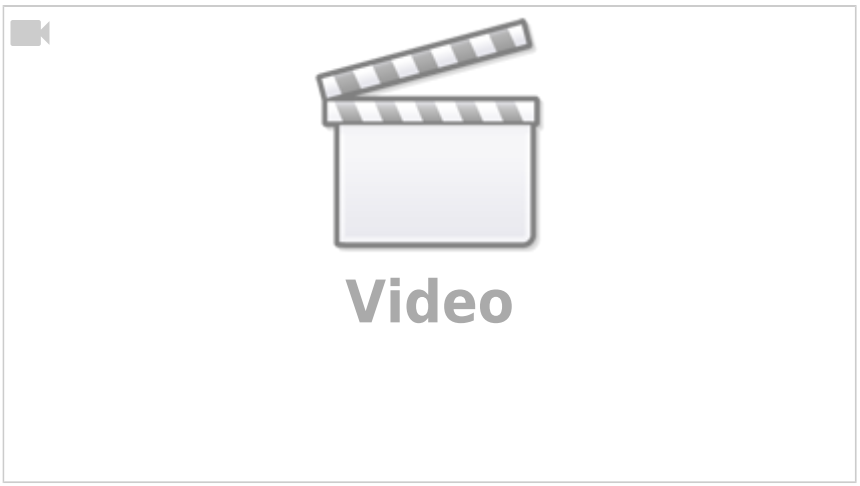
Benchmarks

Also gut wie sieht das ganze nun der Praxis aus? Ich hab natürlich einige Tests gemacht die ich hier natürlich auch gerne präsentiere. Bei allen Tests gabs die gleiche Auflösung, eine Nvidia Quadro K4000 und die gleichen Einstellungen.

Der Erste Test ist mit Windows Server 2012r2. Zugriff erfolge mit VNC. Als Servervariante verwende ich hier TightVNC. Als Test kam die Unigine Heaven zum Einsatz. VNC fühlt sich wie nativ auf der Maschine an, auch die Performance. Das einzige Manko: Gigabit Netzwerk bremst das ganze ein wenig. Hier sollte man ungedingt auf 10Gbit pro Client setzen.

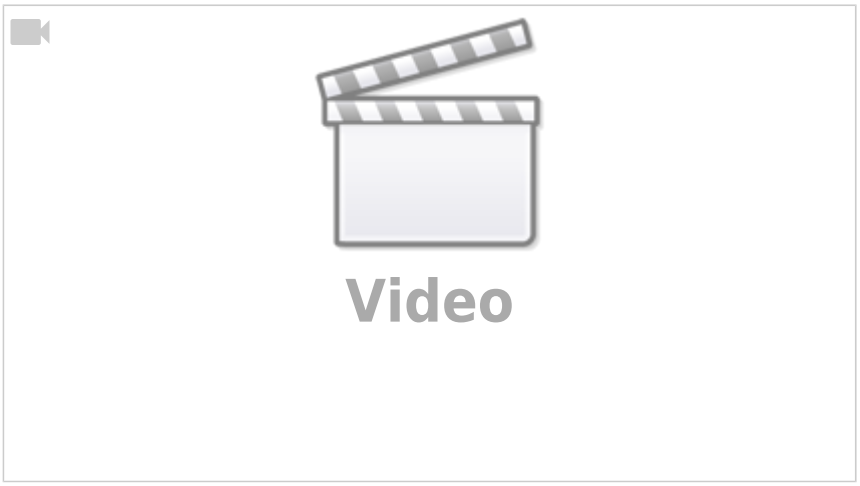


Als Zweites das gleiche Szenario aber mit der Valley Testumgebung.



Ich habe mir auch kurz die Verbindung über RDP angesehen. Vom Netzwerkdurchsatz wird (obwohl alle Einstellungen auf maximale Qualität gestellt waren) um gut die Hälfte weniger benötigt. Es leidet aber merklich am Client die Qualität der Applikation.

Zu guter letzt habe ich noch ein KDEneon 5.9 mit Nomachine als Clientzugriff getestet. Die Qualität ist relativ gleich dem VNC. VNC ist hier eine Spur schöner, ist ja auch völlig unkomprimiert. Aber jetzt zum Sehr postiven: Nomachine benötigt im Netzwerkdurchsatz gerade mal 1 - 2MB/s. Also nur ein Hundertstel gegenüber VNC. Leider ist nomachine, wenn man mehr als einen User auf dem Linux/Windows Terminalserver lassen möchte nicht ganz gratis. Da wäre es natürlich schön es hier mal eine vernüftige Lösung von x2go geben würde. Hier das Video dazu :)



Fazit

Um Grunde funktioniert das ganze sehr gut und die Einrichtungszeit ist gering. Zum Anderen sind 10Gbit am Client natürlich nicht ganz günstig. Geschweige dem was man am Server benötigt. Zum Anderen reichen Gigiabit Interfaces für normale Office Applicationen „doch“ völlig aus. Der Test sollte einfach mal zeigen was möglich ist. Eine Idee wäre natürlich hier auf den Steamlink zu setzen. Der nach meinen eigenen Tests auch schon ganz gut funktioniert. Leider noch lange nicht mit allen Spielen, aber brauchbar. Auch Supertuxkart lief bei meinen Tests auf Windowsserver und KDEneon einwandfrei.

Auch funktioniert GPU-Passthrough nicht mit jedem Board. HP ML350G6/G7… damit hat man keine Freude. Ich vermute mal das es mit G9 nicht anders ist. Hier scheint das ganze per Hardware deaktiviert zu sein. Trotz richtiger Flags im Bios lies sich hier die Grafikarte nicht durchreichen. So wie

es die Erfahrung zeigt und man es auch im Netz liest, ist man mit Supermicro gut beraten. Aber natürlich sollte man auch hier vor dem Kauf nochmal gründlich nachforschen.

Closed Lösung von Nvidia, event. auch interessant:
http://www.nvidia.com/object/grid-technology.html
Und etwas Ähnliches in der OpenSourcewelt: http://moonlight-stream.com/

Quellen

- https://pve.proxmox.com/wiki/Pci_passthrough
- https://forum.proxmox.com/threads/gpu-passthrough-possible-with-my-hardware.19078/
- https://forum.proxmox.com/threads/gpu-passthrough-tutorial-reference.34303/