Proxmox VE Nvidia-vGPU doc

发行版本 *6.4*

bingsin

2023 年 02 月 18 日

目录:

本文将简要介绍 Nvidia vGPU 原理，更加详细的资料，请访问 https://docs.nvidia.com/grid/

详细叙述 Nvidia vGPU 在 Proxmox VE 的部署过程。

CHAPTER 1

第一章认识 vGPU

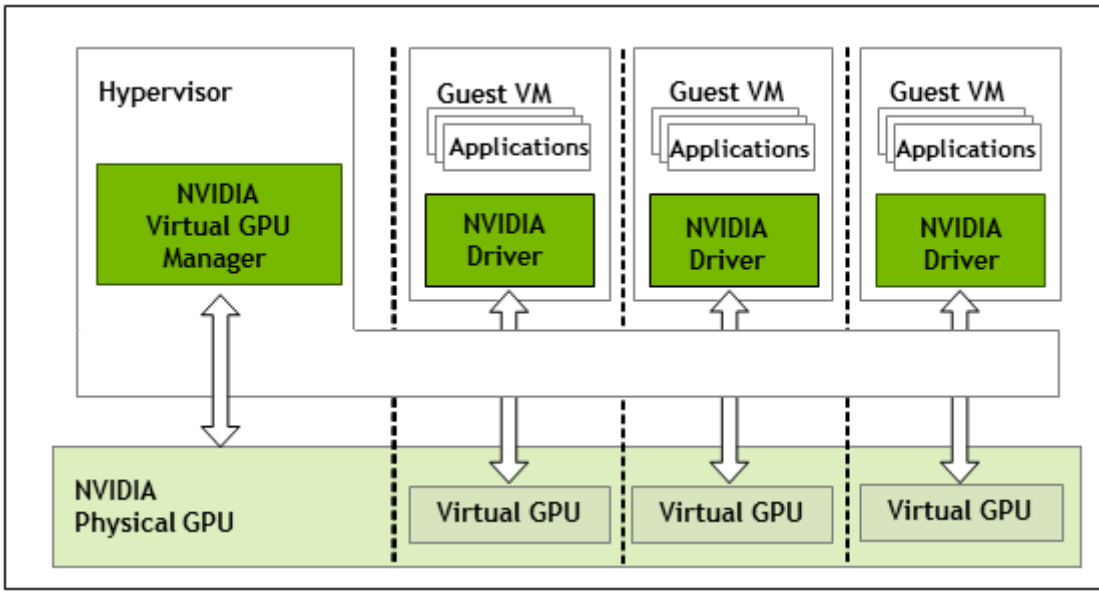
要为 VM 提供图形引擎，一般分为 3 种。

- 软件模拟图形-性能差
- 显卡直通-性能最好，一个虚拟机独享一个显卡
- vGPU-性能好，多个虚拟机共享一个显卡

在市面上主要有如下 3 种 vGPU 解决方案。

本文主要介绍 Nvdia vGPU

1.1 1.1. Nvidia vGPU 架构



下图为 Nvidia vGPU 系统架构

从上图中可以看到在 Hypervisor 的硬件上存在 Nvidia 物理上的 GPU，软件上存在 Nvidia vGPU 管理器。

那么不难看出，要实现 Nvidia vGPU 不仅要物理 GPU，还需要相应的管理程序。

1.2 1.2. Nvidia vGPU 支持的显卡

具体型号请参考下面链接 https://docs.nvidia.com/grid/13.0/product-support-matrix/index.html

大致如下

- A10
- A16
- A30
- A40, A10
- M6, M10, M60
- P4, P6, P40, P100, P100 12GB
- T4
- V100
- RTX A5000
- RTX A6000
- RTX 6000, RTX 6000 passive, RTX 8000, RTX 8000 passive

1.3 1.3. Nvidia vGPU 支持的系统

不同的 Hypervisor 和不同的驱动软件对系统的支持性不一样。完整的兼容性，请参考对应的 Grid 文档

那么笔者归纳大致有如下：

- ubuntu
- redhat
- Windows
- SUSE

1.4 1.4. 获取 Nvidia vGPU 软件和支持

请前往 Nvidia 官网获取软件和支持。

CHAPTER 2

第二章准备 Nvidia vGPU 驱动环境

2.1 2.1 安装必要的软件源

修改/etc/apt/sources.list 使其如下:

pve6

```
#使用清华源
deb https://mirrors.tuna.tsinghua.edu.cn/debian/ buster main contrib non-free
deb https://mirrors.tuna.tsinghua.edu.cn/debian/ buster-updates main contrib non-free
deb https://mirrors.tuna.tsinghua.edu.cn/debian/ buster-backports main contrib non-
↪free
deb https://mirrors.tuna.tsinghua.edu.cn/debian-security buster/updates main contrib␣
↪non-free
#清华pve源镜像
deb https://mirrors.tuna.tsinghua.edu.cn/proxmox/debian buster pve-no-subscription
```

pve7

```
#使用清华源
deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye main contrib non-free
deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-updates main contrib non-
↪free
deb https://mirrors.tuna.tsinghua.edu.cn/debian/ bullseye-backports main contrib non-
↪free
```

(续下页)

(接上页)

```
deb https://mirrors.tuna.tsinghua.edu.cn/debian-security bullseye/updates main␣
↪contrib non-free
#清华pve源镜像
deb https://mirrors.tuna.tsinghua.edu.cn/proxmox/debian bullseye pve-no-subscription
```

2.2 2.2 调整相关内核参数

禁用 nouveau 驱动, 使 Nvidia 显卡不被其占用，从而能够顺利安装 Nvidia vGPU 驱动。

```
echo  "blacklist nouveau" >>/etc/modprobe.d/disable-nouveau.conf
echo  "options nouveau modeset=0" >>/etc/modprobe.d/disable-nouveau.conf
```

提示：如果当前主机不方便重启，可以解绑 *nouveau* 模块。

```
cd /sys/bus/pci/drivers/nouveau
ls  #查看你的PCIe设备号
echo 0000:xxxxxxxxxxxx > unbind  #将000xx替换成PCIe设备号
```

允许不安全中断

```
echo "options vfio_iommu_type1 allow_unsafe_interrupts=1" >/etc/modprobe.d/iommu_
↪unsafe_interrupts.conf
echo "options kvm ignore_msrs=1" > /etc/modprobe.d/kvm.conf
```

配置好之后，需要更新内核以应用内核参数。update-initramfs -k all -u

2.3 2.3 开启 iommu

编辑/etc/default/grub，在 cmdline 中添加 iommu 参数如下

```
#intel_cpu
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on"

#amd_cpu
GRUB_CMDLINE_LINUX_DEFAULT="quiet amd_iommu=on"
```

最后使用 update-grub 更新系统引导。

注意:

如果使用 Systemd-boot 引导的系统，应该编辑 /etc/kernel/cmdline 文件。使用 proxmox-boot-tool refresh 更新系统引导。

2.4 2.4 安装依赖

安装 dkms 和 pve-headers，用于安装驱动。jq 和 uuid-runtime 用于配合 mdevctl 管理 Nvidia vGPU 设备。

```
apt install dkms build-essential pve-headers pve-headers-`uname -r` dkms jq uuid-
↪runtime -y
```

安装 mdevctl

```
curl http://ftp.br.debian.org/debian/pool/main/m/mdevctl/mdevctl_0.81-1_all.deb -o /
↪tmp/mdevctl.deb && dpkg -i /tmp/mdevctl.deb
```

CHAPTER 3

第三章安装 Nvidia vGPU 驱动

3.1 3.1. 下载 Nvidia vGPU 驱动

Proxmox VE 作为 KVM 平台。自然需要下载 KVM 版本的驱动。

Nvidia vGPU 驱动和 Proxmox VE 平台的兼容性（在未使用 patch 得情况下，直接能够安装好驱动，判定为兼容）

可以前往 Nvidia 官网下载驱动，也可以去下面下载。

https://foxi.buduanwang.vip/pan/foxi/Virtualization/vGPU/

3.2 3.2. 安装 Nvidia vGPU 驱动

这里以 460.73.01 为例

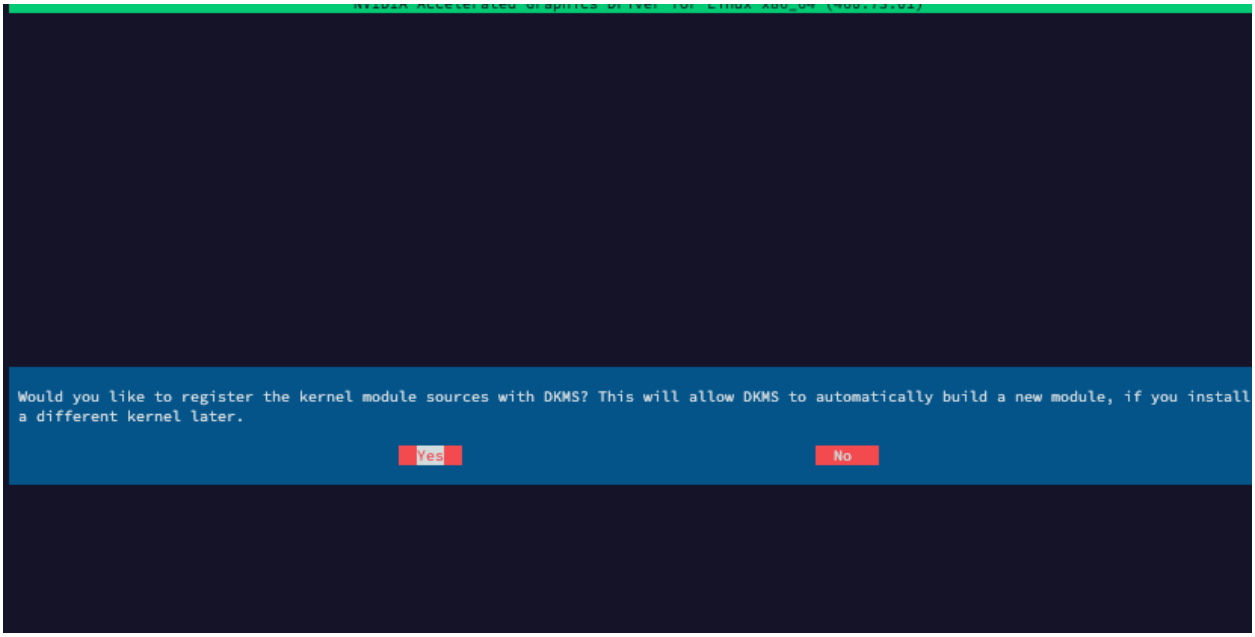
给驱动添加可执行权限

```
chmod +x /opt/NVIDIA-Linux-x86_64-460.73.01-grid-vgpu-kvm.run
```

以 dkms 方式安装驱动

```
sh -c /opt/NVIDIA-Linux-x86_64--grid-vgpu-kvm.run
```

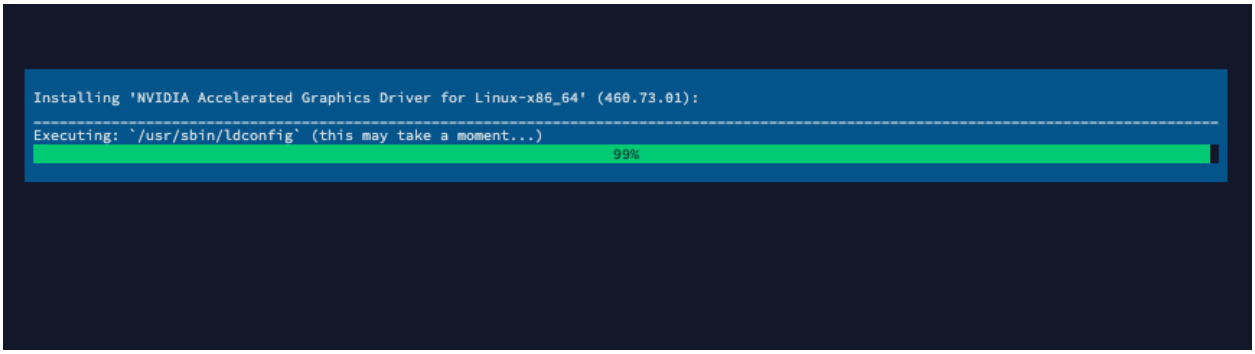
运行命令后，会提示是否用 dkms 方式安装，选择 yes，回车继续



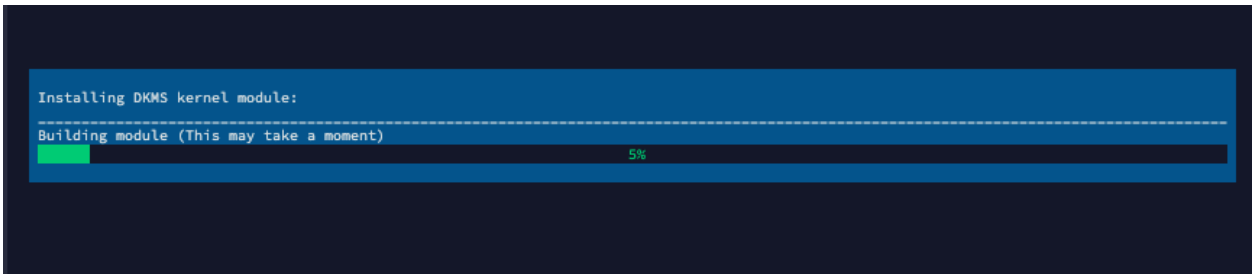
出现 xorg 告警，忽略

询问是否启用 32 位兼容库。这里可选可不选

开始安装驱动



进度条走完就 ok，可能会有点时间。



成功之后，可以执行 `dkms status` 查看

```
root@pve:~# dkms status
nvidia, 460.73.01, 5.4.203-1-pve, x86_64: installed
```

3.3 3.3. 升级或者降级内核

升级或者降级内核务必参考上表的兼容性，否则可能会导致驱动和内核不兼容，使设备无法驱动。

dkms 的优势就是动态安装内核模块，升降内核自动触发模块编译，十分方便。

注意的是，dkms 需要内核的 header，所以当你准备升级或者降级内核的时候，务必安装 header，否则 dkms 会无法编译模块。

例如安装 5.11 内核

```
apt install -y pve-kernel-5.11.22-5-pve pve-headers-5.11.22-5-pve
```

若内核升级降级之前，忘记安装 headers，可以在新内核启动之后，手动安装 header

```
apt install -y pve-headers-`uname -r`
```

再执行 dkms 安装

```
dkms install -m nvidia -v <YOUR_VERSION>
```

3.4 3.4. 开启 vGPU

使用时间分片的 GRID 卡，安装好驱动之后，自动会出现 mdev 设备。

然而，对于 MIG 的 GRID 卡，需要使用开启 SRIOV

参考:https://pve.proxmox.com/wiki/NVIDIA_vGPU_on_Proxmox_VE_7.x

3.5 3.5. 启用和禁用 ECC 内存

某些 GPU 带 ecc 内存，有些 GPU 不带 ecc 内存，有些驱动支持 ECC，有些驱动不支持 ECC，所以请参考下文，合理安排。

https://docs.nvidia.com/grid/latest/grid-software-quick-start-guide/index.html#disabling-enabling-ecc-memory

CHAPTER 4

第四章配置 vGPU

4.1 4.1. 认识 vGPU 的配置

截至目前 2022-9-27，Nvidia 共有 4 中 vGPU 配置。

下面是 Nvidia 官方的介绍：

- vCS：NVIDIA 虚拟计算服务器，加速基于 KVM 的基础架构上的虚拟化 AI 计算工作负载。
- vWS：NVIDIA RTX 虚拟工作站，适用于使用图形应用程序的创意和技术专业人士的虚拟工作站。
- vPC：NVIDIA 虚拟 PC，适用于使用办公效率应用程序和多媒体的知识工作者的虚拟桌面 (VDI)。
- vApp：NVIDIA 虚拟应用程序，采用远程桌面会话主机 (RDSH) 解决方案的应用程序流。

不同的 vGPU 配置使用不同的许可证进行授权。若无授权，则会阶梯降低性能。更多访问：关于 NVIDIA vGPU 软件许可 - 参考

4.2 4.2. 使用 mdevctl 查看 vGPU 配置

`mdevctl` 可以查看当前系统下的 mdev 设备。

下面是一个安装了 P40 的机器的 `mdevctl` 的输出:

```
root@pve:~# mdevctl types
0000:06:00.0           //此处是显卡设备号
  nvidia-156           //mdev设备显示名称
```

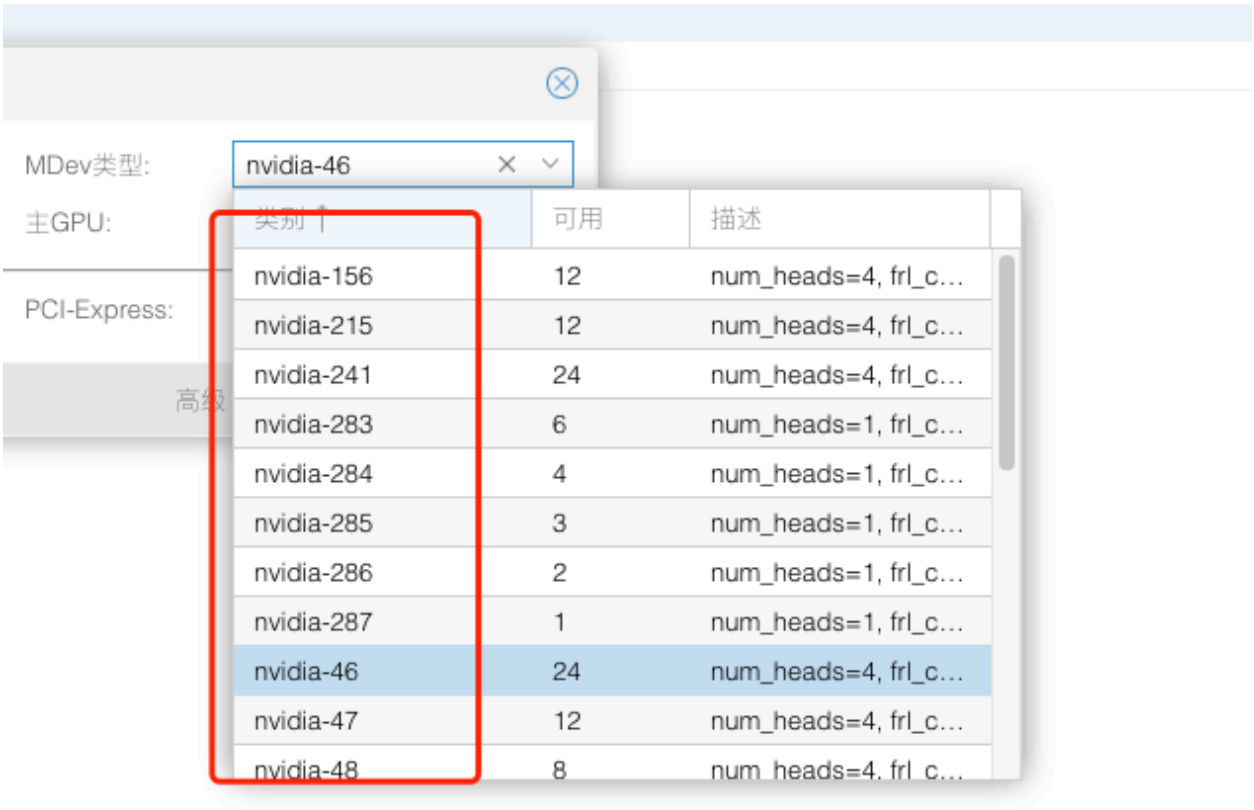
(续下页)

(接上页)

```
    Available instances: 0      //当前可用数量
    Device API: vfio-pci
    Name: GRID P40-2B      //mdev设备友好名称.
    Description: num_heads=4, frl_config=45, framebuffer=2048M, max_
↪resolution=5120x2880, max_instance=12       //mdev设备描述
  nvidia-283
    Available instances: 0
    Device API: vfio-pci
    Name: GRID P40-4C      //C是vGPU配置类型,Q=vWS,C=vCS,B=vPC,A=vApps
    Description: num_heads=1, frl_config=60, framebuffer=4096M, max_
↪resolution=4096x2160, max_instance=6
  nvidia-46
    Available instances: 20
    Device API: vfio-pci
    Name: GRID P40-1Q     //最后面的字母前代表的是显存，1=1G,4=4G
    Description: num_heads=4, frl_config=60, framebuffer=1024M, max_
↪resolution=5120x2880, max_instance=24
```

参考下面链接https://docs.nvidia.com/grid/14.0/grid-vgpu-user-guide/index.html#virtual-gpu-types-grid

在 Proxmox VE 的 web 面板上，我们只能看到 mdev 设备的显示名称，而不是友好名称，所以我们不好判断是否分配了我们想要分配的设备。如下：



所以，要分配正确的设备给虚拟机，建议先使用 `mdevctl` 先对应，否则分配到了错误的 vGPU，可能会导致无法安装驱动或者无法授权等等。

4.3 4.3. vGPU 分配规则

4.3.1 4.3.1 基于时间分片（Time-Sliced）

- 同一张卡上，同显存的 C 和 Q 可以同时分配。
- 同一张卡上，不能同时分配不同显存的 vGPU
- 不同卡上，不同显存可以同时分配，但必须满足上一原则。

4.3.2 4.3.2 基于 MIG（Sriov）

- 更加灵活

此部分参考：https://docs.nvidia.com/grid/14.0/grid-vgpu-user-guide/index.html#valid-vgpu-configurations-one-gpu

4.4 4.4. 将 vGPU 配置至 Proxmox VE 虚拟机

打开虚拟的硬件选项，点击添加，选择 PCI 设备。

在添加：PCI 设备弹窗中，在设备中选择物理显卡，随后会出现 MDev 类型选择框，请选择自己需要的 vGPU 类型。

4.4.1 4.4.1 将 vGPU 作为 PCIe 设备直通

当虚拟机机型为 I440FX 时，虚拟机没有 PCIe 通道，所以 vGPU 会成为一个 PCI 设备。

当虚拟机机型为 Q35 时，虚拟有 PCIe 通道，此时可以选择将 vGPU 作为 PCI 设备或者 PCIe 设备。

若要将 vGPU 作为 PCIe 设备，请点击高级，勾选 PCI-Express.

4.4.2 4.4.2 主 GPU 选项

勾选主 GPU 选项时，会将 x-vga=on 参数传递给 kvm，同时将取消虚拟的默认 vga 设备，所以虚拟机控制台此时不能使用。只能通过系统内部的 VNC 或者其他远程协议访问。

并不建议开启此选项。

4.4.3 4.4.3 vGPU 额外配置

在 PVE7.2, qemu-server 软件包低于 7.2-4 之前，需要手动给虚拟机添加 UUID，才能够启动分配了 vGPU 的虚拟机。

否则会出现如下错误

```
vfio 00000000-0000-0000-0000-000000000102: failed to get region 0 info: Input/output␣
↪error
TASK ERROR: start failed: QEMU exited with code 1
```

方法 1:

```
添加下面行到虚拟机conf中

args: -uuid 00000000-0000-0000-0000-000000000100
```

(续下页)

(接上页)

```
注意的是，uuid最后的值需要改成你的vmid。如果你的vmid为3333，那么你应该改成

args: -uuid 00000000-0000-0000-0000-000000003333

如果你的vmid是121，那么你应该改成

args: -uuid 00000000-0000-0000-0000-000000000121

注意，uuid的长度和格式是不能变的，根据自己的vmid，替换尾数。
```

方法 2：

升级到 qemu-server=7.2-4

方法 3：

参考 qemu-server=7.2-4 中对 nvidia vgpu 的支持，修改源文件

https://git.proxmox.com/?p=qemu-server.git;a=commitdiff;h=bbf96e0f1ea0977c1b37e1ae3bbd9a9aed900c26

CHAPTER 5

第五章在虚拟机中安装 Nvidia 驱动程序

5.1 5.1. GRID 虚拟机驱动程序版本限制与开放

Nvidia vGPU 管理程序允许虚拟机使用旧版本 GRID 驱动，但是有一个最小的版本。

Nvidia vGPU 管理程序不允许虚拟机使用比管理程序高的 GRID 驱动。

参考：https://docs.nvidia.com/grid/14.0/grid-vgpu-release-notes-generic-linux-kvm/index.html#vm-old-drivers-gpu-start-failure

5.2 5.2. 在 Windows 上安装 GRID 驱动程序

请先在 GuestOS 中安装好远程环境，例如 Vnc Server，RDP 等。双击 vGPU 驱动程序中的 Windows 程序，和常规的 Nvidia 驱动安装步骤相同。请参考 vGPU 软件附带的 `user-guide`FDF 中的 `Installing the NVIDIA vGPU Software Graphics Driver on Windows` 部分

5.3 5.3. 在 Linux 上安装 GRID 驱动程序

与 Linux 的 NVIDIA 驱动程序安装步骤类似，请参考 vGPU 软件附带的 user-guideFDF 中的 Installing the NVIDIA vGPU Software Graphics Driver on Linux 部分

CHAPTER 6

第六章使用 vGPU

6.1 Windows 系统

Windows 系统需要使用 OS 内部的远程显示协议，才能够使用 vGPU 硬件。

Proxmox VE 控制台只能输出集成的虚拟 vga 图像，并且不能做解码编码。

所以需要一些特定的远程显示协议：

- VNC
- Parsec
- RustDesk
- sunshine
- todesk
- 向日葵
- ……

6.2 Linux 系统

Proxmox VE 控制台只能输出集成的虚拟 vga 图像，并且不能做解码编码。

所以需要一些特定的远程显示协议：

- VNC
- RustDesk
- todesk

如果并不是作为桌面使用，那么无需远程显示协议。

CHAPTER 7

第七章授权

授权需要配置 License 服务器，和在虚拟机驱动程序中配置 License 服务器地址

7.1 7.1. 安装 License 服务器

在 Windows 中安装 Lic 服务器

https://docs.nvidia.com/grid/ls/2022.09/grid-license-server-user-guide/index.html#installing-nvidia-grid-license-server-windows

在 Linux 中安装 Lic 服务器

https://docs.nvidia.com/grid/ls/2022.09/grid-license-server-user-guide/index.html#installing-nvidia-grid-license-server-linux

7.2 7.2. 在 Lic 服务器中添加许可证

https://docs.nvidia.com/grid/ls/2022.09/grid-license-server-user-guide/index.html#managing-nvidia-grid-license-server

7.3 7.3. 在虚拟机中配置 Lic 服务器

https://docs.nvidia.com/grid/14.0/grid-licensing-user-guide/index.html#configuring-nls-licensed-client