

# PCI Passthrough

Tomáš Tichý

InstallFest 2019

3. března 2019



Uvedené dílo podléhá licenci Creative Commons Uveďte autora 3.0 Česko.

Prezentace je k dispozici ke stažení z:



<https://tichytom.cz/IF19.pdf>

# Co to je PCI Passthrough?

PCI passthrough allows you to give control of physical devices to guests: that is, you can use PCI passthrough to assign a PCI device (NIC, disk controller, HBA, USB controller, firewire controller, soundcard, etc) to a virtual machine guest, giving it full and direct access to the PCI device.

# Potřebný HW

Na zprovoznění je potřeba mít potřebné HW komponenty.

- Procesor podporující technologii IOMMU (*Input Output Memory Management Unit*)
  - Procesory Intel jsou označeny jako VT-d (*Virtualization Technology for Directed I/O.*)
  - U procesorů AMD je označení AMD-Vi.
- Základní desku podporující tyto technologie.
- Dvě grafické karty (může být i jedna integrovaná na desce.)
- Ideálně větší než malé množství paměti RAM.

Dále pro snadnější běh je vhodné mít:

- Dva monitory
- Dvě klávesnice a dvě myši.

# Potřebný HW

Na zprovoznění je potřeba mít potřebné HW komponenty.

- Procesor podporující technologii IOMMU (*Input Output Memory Management Unit*)
  - Procesory Intel jsou označeny jako VT-d (*Virtualization Technology for Directed I/O.*)
  - U procesorů AMD je označení AMD-Vi.
- Základní desku podporující tyto technologie.
- Dvě grafické karty (může být i jedna integrovaná na desce.)
- Ideálně větší než malé množství paměti RAM.

Dále pro snadnější běh je vhodné mít:

- Dva monitory
- Dvě klávesnice a dvě myši.

# Potřebný HW

Na zprovoznění je potřeba mít potřebné HW komponenty.

- Procesor podporující technologii IOMMU (*Input Output Memory Management Unit*)
  - Procesory Intel jsou označeny jako VT-d (*Virtualization Technology for Directed I/O.*)
  - U procesorů AMD je označení AMD-Vi.
- Základní desku podporující tyto technologie.
- Dvě grafické karty (může být i jedna integrovaná na desce.)
- Ideálně větší než malé množství paměti RAM.

Dále pro snadnější běh je vhodné mít:

- Dva monitory
- Dvě klávesnice a dvě myši.

# Potřebný HW

Na zprovoznění je potřeba mít potřebné HW komponenty.

- Procesor podporující technologii IOMMU (*Input Output Memory Management Unit*)
  - Procesory Intel jsou označeny jako VT-d (*Virtualization Technology for Directed I/O.*)
  - U procesorů AMD je označení AMD-Vi.
- Základní desku podporující tyto technologie.
- Dvě grafické karty (může být i jedna integrovaná na desce.)
- Ideálně větší než malé množství paměti RAM.

Dále pro snadnější běh je vhodné mít:

- Dva monitory
- Dvě klávesnice a dvě myši.

# Potřebný HW

Na zprovoznění je potřeba mít potřebné HW komponenty.

- Procesor podporující technologii IOMMU (*Input Output Memory Management Unit*)
  - Procesory Intel jsou označeny jako VT-d (*Virtualization Technology for Directed I/O.*)
  - U procesorů AMD je označení AMD-Vi.
- Základní desku podporující tyto technologie.
- Dvě grafické karty (může být i jedna integrovaná na desce.)
- Ideálně větší než malé množství paměti RAM.

Dále pro snadnější běh je vhodné mít:

- Dva monitory
- Dvě klávesnice a dvě myši.



# Potřebný HW

Na zprovoznění je potřeba mít potřebné HW komponenty.

- Procesor podporující technologii IOMMU (*Input Output Memory Management Unit*)
  - Procesory Intel jsou označeny jako VT-d (*Virtualization Technology for Directed I/O.*)
  - U procesorů AMD je označení AMD-Vi.
- Základní desku podporující tyto technologie.
- Dvě grafické karty (může být i jedna integrovaná na desce.)
- Ideálně větší než malé množství paměti RAM.

Dále pro snadnější běh je vhodné mít:

- Dva monitory
- Dvě klávesnice a dvě myši.

# Potřebný HW

Na zprovoznění je potřeba mít potřebné HW komponenty.

- Procesor podporující technologii IOMMU (*Input Output Memory Management Unit*)
  - Procesory Intel jsou označeny jako VT-d (*Virtualization Technology for Directed I/O.*)
  - U procesorů AMD je označení AMD-Vi.
- Základní desku podporující tyto technologie.
- Dvě grafické karty (může být i jedna integrovaná na desce.)
- Ideálně větší než malé množství paměti RAM.

Dále pro snadnější běh je vhodné mít:

- Dva monitory
- Dvě klávesnice a dvě myši.

# Potřebný HW

Na zprovoznění je potřeba mít potřebné HW komponenty.

- Procesor podporující technologii IOMMU (*Input Output Memory Management Unit*)
  - Procesory Intel jsou označeny jako VT-d (*Virtualization Technology for Directed I/O.*)
  - U procesorů AMD je označení AMD-Vi.
- Základní desku podporující tyto technologie.
- Dvě grafické karty (může být i jedna integrovaná na desce.)
- Ideálně větší než malé množství paměti RAM.

Dále pro snadnější běh je vhodné mít:

- Dva monitory
- Dvě klávesnice a dvě myši.

# Potřebný HW

Na zprovoznění je potřeba mít potřebné HW komponenty.

- Procesor podporující technologii IOMMU (*Input Output Memory Management Unit*)
  - Procesory Intel jsou označeny jako VT-d (*Virtualization Technology for Directed I/O.*)
  - U procesorů AMD je označení AMD-Vi.
- Základní desku podporující tyto technologie.
- Dvě grafické karty (může být i jedna integrovaná na desce.)
- Ideálně větší než malé množství paměti RAM.

Dále pro snadnější běh je vhodné mít:

- Dva monitory
- Dvě klávesnice a dvě myši.

Pokud máme potřebný HW a nainstalovaný systém, můžeme přejít k zprovoznění.

### Potřebné balíčky

```
zypper in libvirt libvirt-client libvirt-daemon virt-manager  
virt-install virt-viewer qemu qemu-kvm qemu-ovmf-x86_64 qemu-tools
```

Mohou se i hodit ovladače na VirtIO, ty lze nalézt ke stažení na [Fedorapeople.org](https://fedorapeople.org):

<https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso>

Pokud máme potřebný HW a nainstalovaný systém, můžeme přejít k zprovoznění.

### Potřebné balíčky

```
zypper in libvirt libvirt-client libvirt-daemon virt-manager  
virt-install virt-viewer qemu qemu-kvm qemu-ovmf-x86_64 qemu-tools
```

Mohou se i hodit ovladače na VirtIO, ty lze nalézt ke stažení na [Fedorapeople.org](https://fedorapeople.org):

```
https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso
```

Nyní máme vše připraveno k samotnému zprovoznění.

Zapneme si podporu IOMMU a to přidáním **intel\_iommu=on**, nebo **amd\_iommu=on** dle výrobce procesoru, na řádek `GRUB_CMDLINE_LINUX_DEFAULT` do souboru `/etc/default/grub`.

Bude to vypadat nějak takto:

```
GRUB_CMDLINE_LINUX_DEFAULT="video ... quiet showopts intel_iommu=on"
```

Poté je třeba inicializovat GRUB příkazem

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Nyní máme vše připraveno k samotnému zprovoznění.

Zapneme si podporu IOMMU a to přidáním **intel\_iommu=on**, nebo **amd\_iommu=on** dle výrobce procesoru, na řádek `GRUB_CMDLINE_LINUX_DEFAULT` do souboru `/etc/default/grub`.

Bude to vypadat nějak takto:

```
GRUB_CMDLINE_LINUX_DEFAULT="video ... quiet showopts intel_iommu=on"
```

Poté je třeba inicializovat GRUB příkazem

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```



Nyní máme vše připraveno k samotnému zprovoznění.

Zapneme si podporu IOMMU a to přidáním **intel\_iommu=on**, nebo **amd\_iommu=on** dle výrobce procesoru, na řádek `GRUB_CMDLINE_LINUX_DEFAULT` do souboru `/etc/default/grub`.

Bude to vypadat nějak takto:

```
GRUB_CMDLINE_LINUX_DEFAULT="video ... quiet showopts intel_iommu=on"
```

Poté je třeba inicializovat GRUB příkazem

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Po restartu systému zkontrolujeme jestli se IOMMU aktivovalo příkazem

```
dmesg|grep -e DMAR -e IOMMU
```

výstup z dmesg:

```
[ 0.000000] ACPI: DMAR 0x00000000C89E9868 0000B8 (v01 INTEL SNB 00000001 INTL 00000001)
[ 0.000000] DMAR: IOMMU enabled
[ 0.027037] DMAR: Host address width 36
[ 0.027039] DMAR: DRHD base: 0x000000fed90000 flags: 0x0
[ 0.027046] DMAR: dmar0: reg_base_addr fed90000 ver 1:0 cap c0000020e60262 ecap f0101a
[ 0.027047] DMAR: DRHD base: 0x000000fed91000 flags: 0x1
[ 0.027050] DMAR: dmar1: reg_base_addr fed91000 ver 1:0 cap c9008020660262 ecap f0105a
[ 0.027050] DMAR: RMRR base: 0x000000c89ea000 end: 0x000000c8a06fff
[ 0.027051] DMAR: RMRR base: 0x000000cb000000 end: 0x000000cflffffff
[ 0.027053] DMAR-IR: IOAPIC id 2 under DRHD base 0xfed91000 IOMMU 1
:
```

Po restartu systému zkontrolujeme jestli se IOMMU aktivovalo příkazem

```
dmesg|grep -e DMAR -e IOMMU
```

výstup z dmesg:

```
[ 0.000000] ACPI: DMAR 0x00000000C89E9868 0000B8 (v01 INTEL SNB 00000001 INTL 00000001)
[ 0.000000] DMAR: IOMMU enabled
[ 0.027037] DMAR: Host address width 36
[ 0.027039] DMAR: DRHD base: 0x000000fed90000 flags: 0x0
[ 0.027046] DMAR: dmar0: reg_base_addr fed90000 ver 1:0 cap c0000020e60262 ecap f0101a
[ 0.027047] DMAR: DRHD base: 0x000000fed91000 flags: 0x1
[ 0.027050] DMAR: dmar1: reg_base_addr fed91000 ver 1:0 cap c9008020660262 ecap f0105a
[ 0.027050] DMAR: RMRR base: 0x000000c89ea000 end: 0x000000c8a06fff
[ 0.027051] DMAR: RMRR base: 0x000000cb000000 end: 0x000000cf1fffff
[ 0.027053] DMAR-IR: IOAPIC id 2 under DRHD base 0xfed91000 IOMMU 1
:
```

Nyní musíme zjistit do které skupiny je přiřazena grafická karta

```
lsiommu.sh
```

```
#!/bin/bash
shopt -s nullglob
for d in /sys/kernel/iommu_groups/*/devices/*; do
  n=${d#/iommu_groups/*}; n=${n%%/*}
  printf 'IOMMU Group %s ' "$n"
  lspci -nns "${d##*/}"
done;
```

Nyní musíme zjistit do které skupiny je přiřazena grafická karta

```
lsiommu.sh
```

```
#!/bin/bash
shopt -s nullglob
for d in /sys/kernel/iommu_groups/*/devices/*; do
    n=${d#/iommu_groups/*}; n=${n%%/*}
    printf 'IOMMU Group %s ' "$n"
    lspci -nns "${d##*/}"
done;
```

## výstup iommu.sh

```
IOMMU Group 0 00:00.0 Host bridge [0600]: Intel Corporation Xeon E3-1200...[8086:0150]...
IOMMU Group 1 00:01.0 PCI bridge [0604]: Intel Corporation Xeon E3-1200...[8086:0151]...
IOMMU Group 1 01:00.0 VGA comp...NVIDIA Corporation GP107 [GeForce GTX 1050] [10de:1c81]...
IOMMU Group 1 01:00.1 Audio dev...GP107GL High Definition Audio Controller [10de:0fb9]...
IOMMU Group 10 00:1f.0 ISA bridge [0601]: Intel Corporation Q77 ...[8086:1e47] (rev 04)
IOMMU Group 10 00:1f.2 SATA controller [0106]: Intel...[AHCI mode] [8086:1e02] (rev 04)
IOMMU Group 10 00:1f.3 SMBus [0c05]: Intel Corporation 7 Series/...[8086:1e22] (rev 04)
IOMMU Group 2 00:02.0 VGA compatible controller [0300]: Intel Co...[8086:0152] (rev 09)
IOMMU Group 3 00:14.0 USB controller [0c03]: Intel Corporation 7...[8086:1e31] (rev 04)
IOMMU Group 4 00:16.0 Communication controller [0780]: Intel Cor...[8086:1e3a] (rev 04)
IOMMU Group 4 00:16.3 Serial controller [0700]: Intel Corporatio...[8086:1e3d] (rev 04)
:
:
```

Opíšeme si hodnoty z IOMMU skupiny

```
options vfio-pci ids=id_GK,id_GK_audio
```

```
options vfio-pci ids=10de:1c81,10de:0fb9
```

A tento řetězec vložíme do souboru `/etc/modprobe.d/gpu-passthrough.conf`

Opíšeme si hodnoty z IOMMU skupiny

```
options vfio-pci ids=id_GK,id_GK_audio
```

```
options vfio-pci ids=10de:1c81,10de:0fb9
```

A tento řetězec vložíme do souboru `/etc/modprobe.d/gpu-passthrough.conf`



Opíšeme si hodnoty z IOMMU skupiny

```
options vfio-pci ids=id_GK,id_GK_audio
```

```
options vfio-pci ids=10de:1c81,10de:0fb9
```

A tento řetězec vložíme do souboru `/etc/modprobe.d/gpu-passthrough.conf`

Opět zeditujeme soubor `/etc/default/grub` a opět na konec řádku `GRUB_CMDLINE_LINUX_DEFAULT` vložíme

```
rd.driver.pre=vfio-pci
```

Ve finále to bude vypadat takto:

```
GRUB_CMDLINE_LINUX_DEFAULT="vide...intel_iommu=on rd.driver.pre=vfio-pci"
```

a opět reinitializujeme GRUB

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Opět zeditujeme soubor `/etc/default/grub` a opět na konec řádku `GRUB_CMDLINE_LINUX_DEFAULT` vložíme

```
rd.driver.pre=vfio-pci
```

Ve finále to bude vypadat takto:

```
GRUB_CMDLINE_LINUX_DEFAULT="vide...intel_iommu=on rd.driver.pre=vfio-pci"
```

a opět reinitializujeme GRUB

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Opět zeditujeme soubor `/etc/default/grub` a opět na konec řádku `GRUB_CMDLINE_LINUX_DEFAULT` vložíme

```
rd.driver.pre=vfio-pci
```

Ve finále to bude vypadat takto:

```
GRUB_CMDLINE_LINUX_DEFAULT="vide...intel_iommu=on rd.driver.pre=vfio-pci"
```

a opět reinitializujeme GRUB

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

# Sestavení initrd

Je potřeba ještě vložit:

```
add_drivers+="pci_stub vfio vfio_iommu_type1 vfio_pci vfio_virqfd kvm kvm_intel"
```

do souboru `/etc/dracut.conf.d/gpu-passthrough.conf`

a reinitializujeme initrd příkazem:

```
dracut --force /boot/initrd $(uname -r)
```

**POZOR!**

Pokud uděláte chybu systém Vám již nenaběhne.

# Sestavení initrd

Je potřeba ještě vložit:

```
add_drivers+="pci_stub vfio vfio_iommu_type1 vfio_pci vfio_virqfd kvm kvm_intel"
```

do souboru `/etc/dracut.conf.d/gpu-passthrough.conf`

a reinitializujeme initrd příkazem:

```
dracut --force /boot/initrd $(uname -r)
```

**POZOR!**

Pokud uděláte chybu systém Vám již nenaběhne.

# Sestavení initrd

Je potřeba ještě vložit:

```
add_drivers+="pci_stub vfio vfio_iommu_type1 vfio_pci vfio_virqfd kvm kvm_intel"
```

do souboru `/etc/dracut.conf.d/gpu-passthrough.conf`

a reinitializujeme initrd příkazem:

```
dracut --force /boot/initrd $(uname -r)
```

**POZOR!**

Pokud uděláte chybu systém Vám již nenaběhne.

Po rebootu si zkontrolujeme, jestli je grafická karta izolovaná od systému.

Spustíme příkaz:

```
lspci -k
```

a výpis by měl vypadat přibližně takto:

```
⋮
```

```
01:00.0 VGA compatible controller: NVIDIA Corporation GP107 [...
```

```
Subsystem: Gigabyte Technology Co., Ltd Device 3765
```

```
Kernel driver in use: vfio-pci
```

```
Kernel modules: nouveau, nvidia_drm, nvidia
```

```
01:00.1 Audio device: NVIDIA Corporation GP107GL High Definiti...
```

```
Subsystem: Gigabyte Technology Co., Ltd Device 3765
```

```
Kernel driver in use: vfio-pci
```

```
Kernel modules: snd_hda_intel
```



Po rebootu si zkontrolujeme, jestli je grafická karta izolovaná od systému.  
Spustíme příkaz:

```
lspci -k
```

a výpis by měl vypadat přibližně takto:

```
⋮
```

```
01:00.0 VGA compatible controller: NVIDIA Corporation GP107 [...  
  Subsystem: Gigabyte Technology Co., Ltd Device 3765  
  Kernel driver in use: vfio-pci  
  Kernel modules: nouveau, nvidia_drm, nvidia  
01:00.1 Audio device: NVIDIA Corporation GP107GL High Definiti...  
  Subsystem: Gigabyte Technology Co., Ltd Device 3765  
  Kernel driver in use: vfio-pci  
  Kernel modules: snd_hda_intel
```

Po rebootu si zkontrolujeme, jestli je grafická karta izolovaná od systému.  
Spustíme příkaz:

```
lspci -k
```

a výpis by měl vypadat přibližně takto:

```
:
```

```
01:00.0 VGA compatible controller: NVIDIA Corporation GP107 [...  
  Subsystem: Gigabyte Technology Co., Ltd Device 3765  
  Kernel driver in use: vfio-pci  
  Kernel modules: nouveau, nvidia_drm, nvidia  
01:00.1 Audio device: NVIDIA Corporation GP107GL High Definiti...  
  Subsystem: Gigabyte Technology Co., Ltd Device 3765  
  Kernel driver in use: vfio-pci  
  Kernel modules: snd_hda_intel
```

# Vytváření VM

Nyní můžeme přejít k vytvoření virtuálního stroje (VM).

Ještě předtím si do `/etc/libvirt/qemu.conf` doplníme následující řádky:

```
nvram = [  
    "/usr/share/qemu/ovmf-x86_64.bin:/usr/share/qemu/ovmf-x86_64-code.bin"  
]
```

Nové VM

## Vytvořit nový virtuální stroj

Krok 1 z 5

Připojení: QEMU/KVM

Vyberte, jak chcete nainstalovat operační systém

- Instalace z lokálního média (ISO nebo CDROM)
- Instalace ze sítě (HTTP, FTP nebo NFS)
- Boot ze sítě (PXE)
- Importovat již existující obraz disku

► Architecture options

Zrušit Zpět Vpřed

Nové VM

## Vytvořit nový virtuální stroj

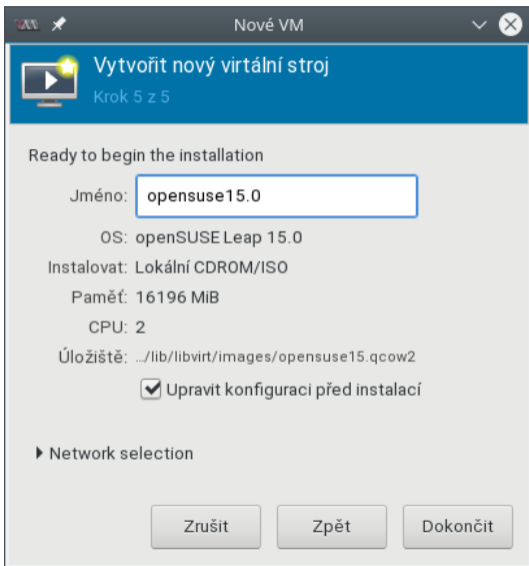
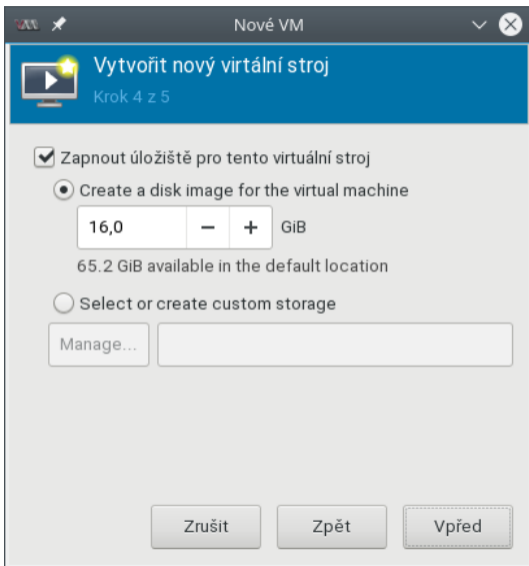
Krok 3 z 5

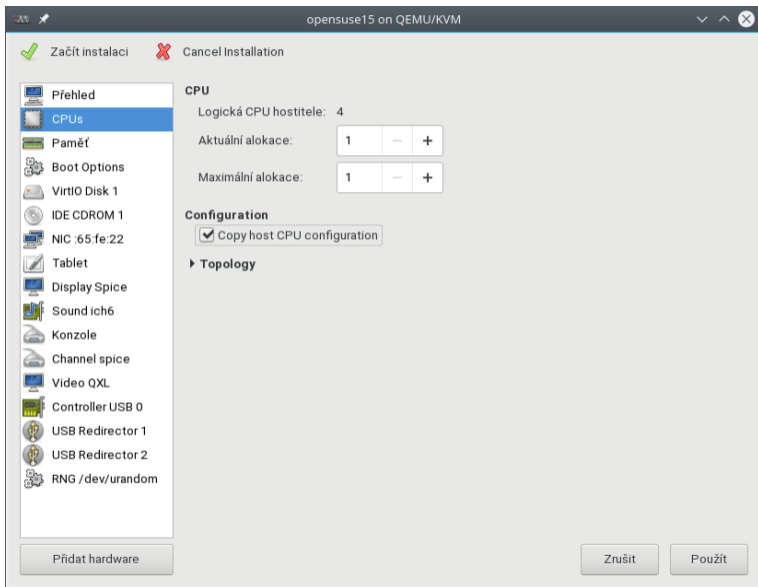
Nastavte paměť a CPU

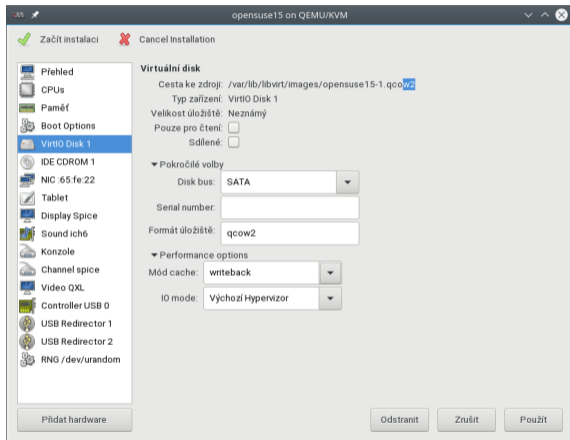
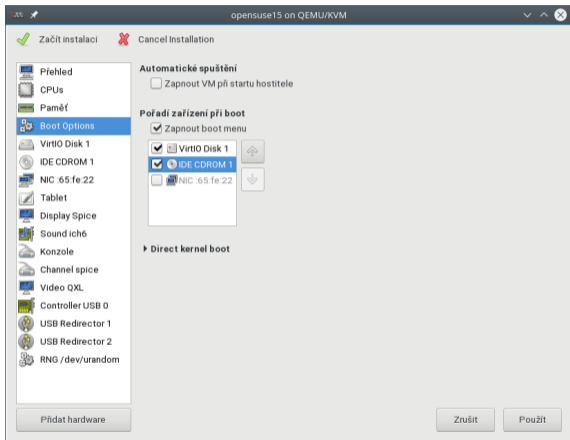
Paměť (RAM): 16196 - +  
Up to 32100 MiB available on the host

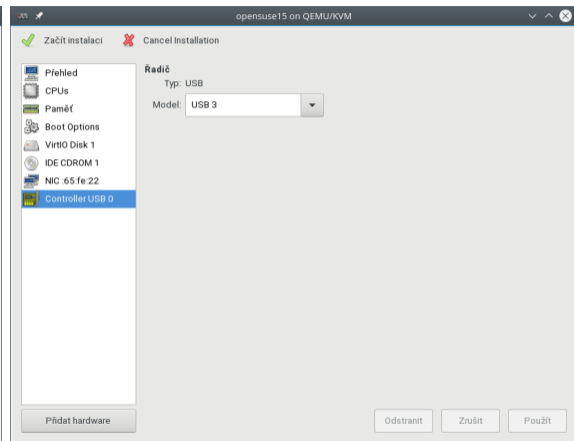
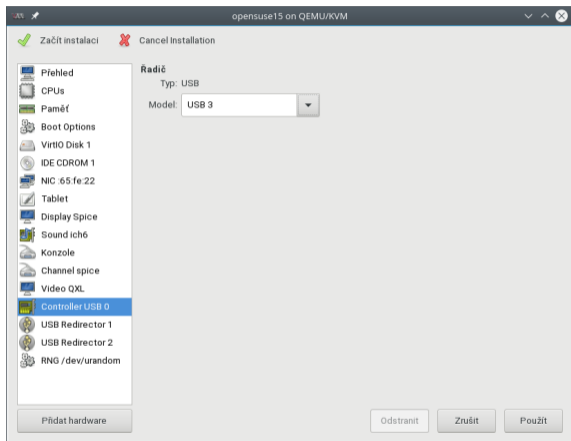
CPU: 2 - +  
Up to 4 available

Zrušit Zpět Vpřed

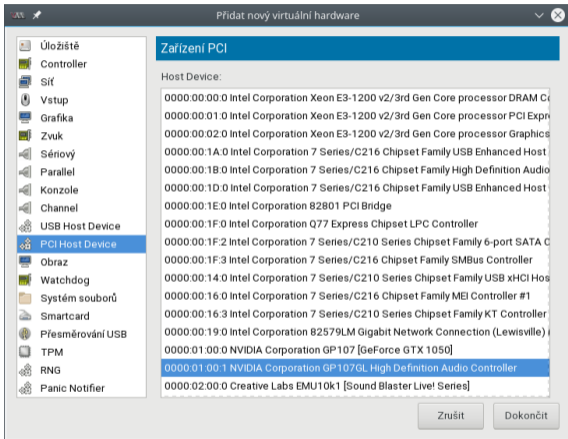
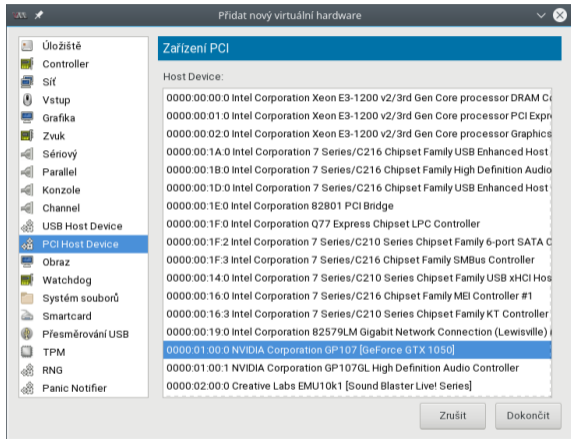


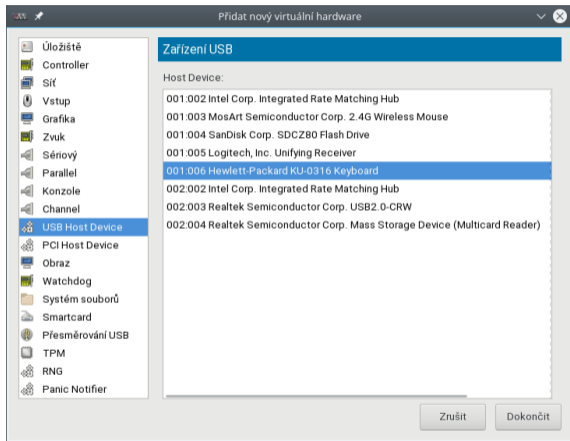
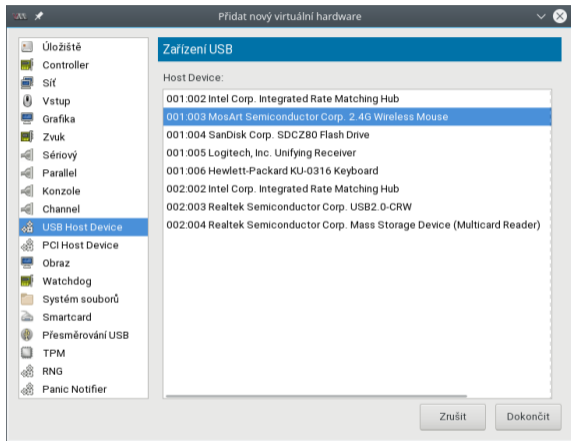












## Potíže s Pulseaudio



Červený trpaslík (RED DWARF®) je ochrannou známkou společnosti Grant Naylor®.

## Potíže s Pulseaudio



Červený trpaslík (RED DWARF®) je ochrannou známkou společnosti Grant Naylor®.

# Potíže s Pulseaudio

Oprava je naštěstí snadná.

Stačí jen trochu pozměnit konfiguraci virtuálního stroje pomocí příkazu:

```
virsh edit nazev_vm
```

Otevře se textový editor, kde je potřeba na prvním řádku změnit:

```
<domain type='kvm'>
```

na:

```
<domain type='kvm' xmlns:gemu='http://libvirt.org/schemas/domain/gemu/1.0'>
```

# Potíže s Pulseaudio

Oprava je naštěstí snadná.

Stačí jen trochu pozměnit konfiguraci virtuálního stroje pomocí příkazu:

```
virsh edit nazev_vm
```

Otevře se textový editor, kde je potřeba na prvním řádku změnit:

```
<domain type='kvm'>
```

na:

```
<domain type='kvm' xmlns:gemu='http://libvirt.org/schemas/domain/gemu/1.0'>
```

# Potíže s Pulseaudio

Oprava je naštěstí snadná.

Stačí jen trochu pozměnit konfiguraci virtuálního stroje pomocí příkazu:

```
virsh edit nazev_vm
```

Otevře se textový editor, kde je potřeba na prvním řádku změnit:

```
<domain type='kvm'>
```

na:

```
<domain type='kvm' xmlns:gemu='http://libvirt.org/schemas/domain/gemu/1.0'>
```

# Potíže s Pulseaudio

Oprava je naštěstí snadná.

Stačí jen trochu pozměnit konfiguraci virtuálního stroje pomocí příkazu:

```
virsh edit nazev_vm
```

Otevře se textový editor, kde je potřeba na prvním řádku změnit:

```
<domain type='kvm'>
```

na:

```
<domain type='kvm' xmlns:qemu='http://libvirt.org/schemas/domain/qemu/1.0'>
```



# Potíže s Pulseaudio

a na konec souboru mezi `</devices>` a `</domain>` připsat:

```
<qemu:commandline>  
  <qemu:env name='QEMU_AUDIO_DRV' value='pa' />  
  <qemu:env name='QEMU_PA_SAMPLES' value='8192' />  
  <qemu:env name='QEMU_AUDIO_TIMER_PERIOD' value='99' />  
  <qemu:env name='QEMU_PA_SERVER' value='/run/user/1000/pulse/native' />  
</qemu:commandline>
```

# Potíže s ovladači

Tím potíže ovšem nekončí.

- Po nainstalování *správných* ovladačů od Nvidie dojde k rozbití X serveru.
- Karty od AMD jsou *zatím* v pohodě.

Oprava se provádí opět úpravou konfiguračního souboru virtuálního stroje.  
Do sekce `<features>` před část, která začíná `<kvm>` stačí vložit:

```
<hyperv>  
  <vendor_id state='on' value='1234567890ab' />  
</hyperv>
```

# Potíže s ovladači

Tím potíže ovšem nekončí.

- Po nainstalování *správných* ovladačů od Nvidie dojde k rozbití X serveru.
- Karty od AMD jsou *zatím* v pohodě.

Oprava se provádí opět úpravou konfiguračního souboru virtuálního stroje.  
Do sekce `<features>` před část, která začíná `<kvm>` stačí vložit:

```
<hyperv>  
  <vendor_id state='on' value='1234567890ab' />  
</hyperv>
```

# Potíže s ovladači

Tím potíže ovšem nekončí.

- Po nainstalování *správných* ovladačů od Nvidie dojde k rozbití X serveru.
- Karty od AMD jsou *zatím* v pohodě.

Oprava se provádí opět úpravou konfiguračního souboru virtuálního stroje.  
Do sekce `<features>` před část, která začíná `<kvm>` stačí vložit:

```
<hyperv>  
  <vendor_id state='on' value='1234567890ab' />  
</hyperv>
```

# Potíže s ovladači

Tím potíže ovšem nekončí.

- Po nainstalování *správných* ovladačů od Nvidie dojde k rozbití X serveru.
- Karty od AMD jsou *zatím* v pohodě.

Oprava se provádí opět úpravou konfiguračního souboru virtuálního stroje.

Do sekce `<features>` před část, která začíná `<kvm>` stačí vložit:

```
<hyperv>
  <vendor_id state='on' value='1234567890ab' />
</hyperv>
```

# Potíže s ovladači

Tím potíže ovšem nekončí.

- Po nainstalování *správných* ovladačů od Nvidie dojde k rozbití X serveru.
- Karty od AMD jsou *zatím* v pohodě.

Oprava se provádí opět úpravou konfiguračního souboru virtuálního stroje.  
Do sekce `<features>` před část, která začíná `<kvm>` stačí vložit:

```
<hyperv>  
  <vendor_id state='on' value='1234567890ab' />  
</hyperv>
```

# Málo místa na desce

- Pokud počítač nemá integrovanou grafickou kartu, lze přidat druhou.
- Dnešní grafické karty jsou výkonné a dvouslotové, tudíž obsadí i vedlejší PCIe slot.
- Pokud na desce již další vhodný není, lze využít pomocí vhodné redukce volný PCIe 1x slot.
- Takto připojená karta ovšem není vhodná pro aplikace vyžadující vysoký výkon.

# Málo místa na desce

- Pokud počítač nemá integrovanou grafickou kartu, lze přidat druhou.
- Dnešní grafické karty jsou výkonné a dvouslotové, tudíž obsadí i vedlejší PCIe slot.
- Pokud na desce již další vhodný není, lze využít pomocí vhodné redukce volný PCIe 1x slot.
- Takto připojená karta ovšem není vhodná pro aplikace vyžadující vysoký výkon.



# Málo místa na desce

- Pokud počítač nemá integrovanou grafickou kartu, lze přidat druhou.
- Dnešní grafické karty jsou výkonné a dvouslotové, tudíž obsadí i vedlejší PCIe slot.
- Pokud na desce již další vhodný není, lze využít pomocí vhodné redukce volný PCIe 1x slot.
- Takto připojená karta ovšem není vhodná pro aplikace vyžadující vysoký výkon.

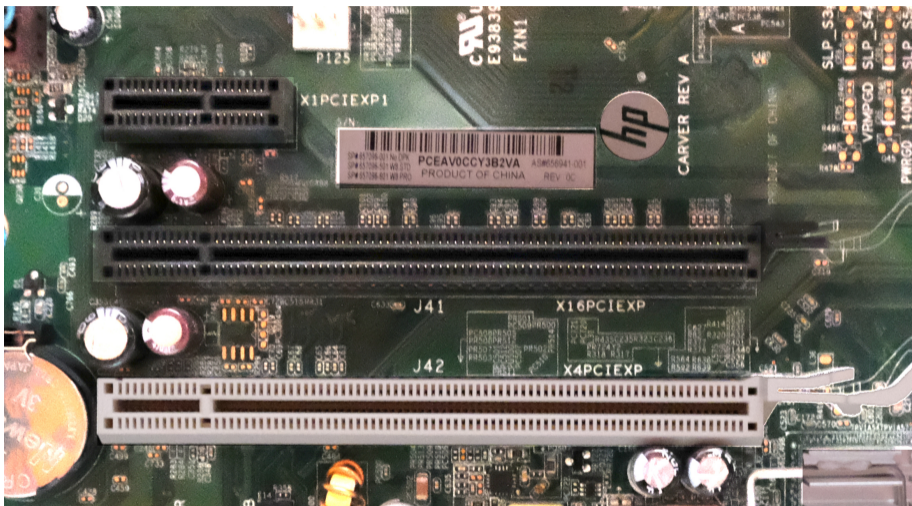
# Málo místa na desce

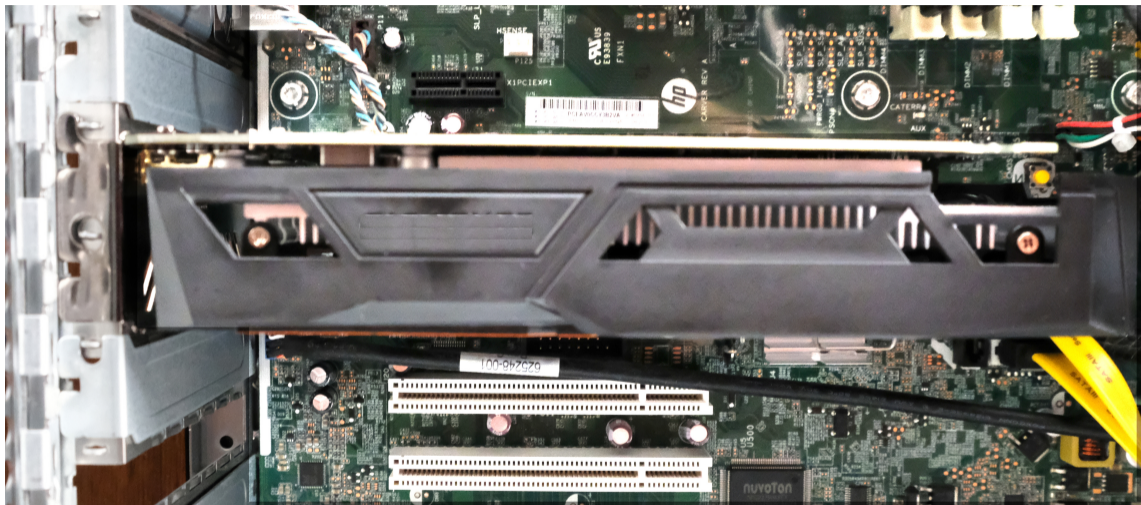
- Pokud počítač nemá integrovanou grafickou kartu, lze přidat druhou.
- Dnešní grafické karty jsou výkonné a dvouslotové, tudíž obsadí i vedlejší PCIe slot.
- Pokud na desce již další vhodný není, lze využít pomocí vhodné redukce volný PCIe 1x slot.
- Takto připojená karta ovšem není vhodná pro aplikace vyžadující vysoký výkon.

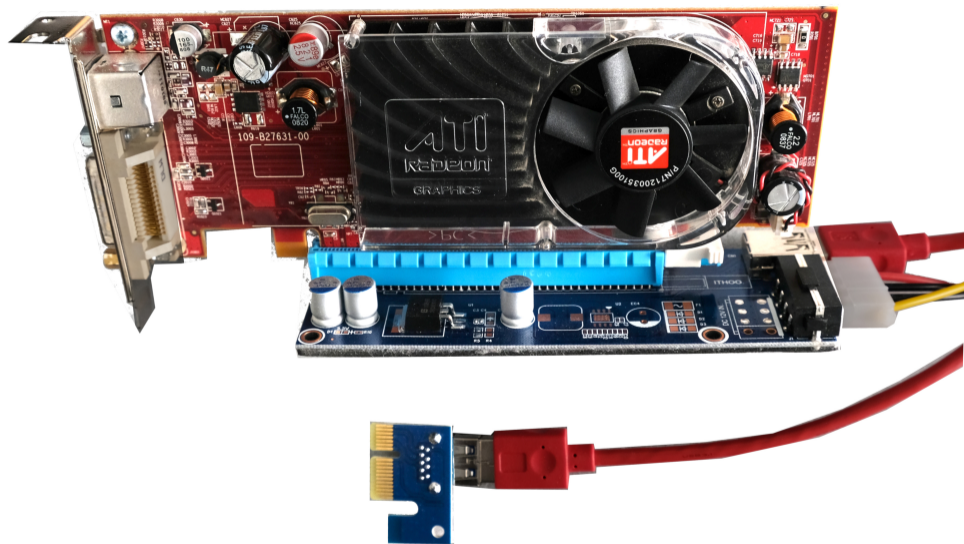
# Porovnání rychlosti PCIe

Typ	Jednosměrně	Obousměrně
x1	250 MB/s	500 MB/s
x4	1 GB/s	2 GB/s
x8	2 GB/s	4 GB/s
x16	4 GB/s	8 GB/s

*Porovnání rychlostí PCI-Express 1.0a*







# Porovnání výkonu

Hra	Linux			Native			Virtual		
	min	avg	max	min	avg	max	min	avg	max
CS:GO	25	190	318	22	182	297	24	43.9	60
Far Cry: New Dawn	-	-	-	5	51	61	7	17	21
	-			3306			1111		
Metro Exodus	-	-	-	6.59	27.69	91.98	1.78	16.56	28.76
	-			2833			1651		
Unigine Heaven	10.8	57.5	112.7	9.1	61.6	124.9	8.4	59.5	128.6
	1449			1551			1498		
Unigine Valley	30.2	58.6	93.9	24.3	64.7	119.0	20.8	57.6	91.8
	2450			2708			2412		

## Reference

-  [https://wiki.archlinux.org/index.php/PCI\\_passthrough\\_via\\_OVMF](https://wiki.archlinux.org/index.php/PCI_passthrough_via_OVMF)
-  <https://en.wikipedia.org/wiki/IOMMU>
-  [https://www.reddit.com/r/VFIO/comments/542bw1/ha\\_got\\_rid\\_of\\_the\\_pulse\\_audio\\_crackling/](https://www.reddit.com/r/VFIO/comments/542bw1/ha_got_rid_of_the_pulse_audio_crackling/)
-  <https://forums.opensuse.org/showthread.php/522015-VGA-PCI-Passthrough-guide-on-openSuSE-Leap-42-2>



# Děkuji za pozornost.

Tomáš Tichý

Tato prezentace je k dispozici ke stažení: <https://tichytom.cz/IF19.pdf>