PCI Passthrough

Ensure IOMMU Is Activated

> First step of this process is to make sure that your hardware is even capable of this type of virtualization. You need to have a motherboard, CPU, and BIOS that has an IOMMU controller and supports Intel-VT-x and Intel-VT-d or AMD-v and AMD-vi. Some motherboards use different terminology for these, for example they may list AMD-v as SVM and AMD-vi as IOMMU controller.

Ensure Kernel Modules

Debian

```
# /etc/modules
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.
+ vfio_pci
+ vfio
+ vfio_iommu_type1
+ vfio_virqfd
```

Bind `vfio-pci` Driver to Devices

Now you can bind the `vfio-pci` driver to your devices at startup so they can be passed through to a VM. There are two ways of doing this, the first way is quick and easy but forces you to blacklist an entire driver which would stop you from being able to use that driver for another device that you aren't passing through. The second way is a little more complciated but allows you to target individual devices without blacklisting an entire driver.

1) Blacklist Drivers

By running `lspci -knn` you can easily find out which drivers are being used for a device so you know what driver to blacklist in addition to their *<vendor>:<device>* identifier. Armed with both of these we can blacklist the drivers we don't want being used and let the *vfio-pci* driver know which device(s) to bind to.

Below is an example of blacklisting the driver `i915` (Intel iGPU driver) so I can pass through my iGPU to a virtual machine. The driver is blacklisted so it won't load and the device identified by *<vendor>:<device>* is added as a parameter to the `vfio-pci` driver so it knows which device to bind with.

```
# /etc/modprobe.d/blacklist.conf
+ blacklist i915
```

```
# /etc/modprobe.d/vfio.conf
+ options vfio-pci ids=8086:3e92 disable_vga=1
```

2) Alias Devices

Using `lspci -knn` it is easy to find a devices B/D/F identifier and its *<vendor>:<device>* identifier. Then we can find its *modalias* by running `cat /sys/bus/pci/devices/<B/D/F>/modalias`. Armed with both of these we can let the *vfio-pci* module know which devices to bind to.

```
# /etc/modprobe.d/vfio.conf
+ # Intel UHD 630 (8086:3e92)
+ alias pci:v00008086d00003E92sv00001458sd0000D000bc03sc80i00 vfio-pci
+
+ options vfio-pci ids=8086:3e92 disable_vga=1
```

Rebuild `initramfs`

Debian

```
update-initramfs -u
```

Update Bootloader

Update Kernel Parameters

Grub2

```
# /etc/default/grub
- GRUB_CMDLINE_LINUX_DEFAULT="quiet"
+ GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=igfx_off iommu=pt video=efifb:off"
```

Systemd

```
# /etc/kernel/cmdline
- root=ZFS=rpool/ROOT/pve-1 boot=zfs
+ root=ZFS=rpool/ROOT/pve-1 boot=zfs intel_iommu=igfx_off iommu=pt video=efifb:off
```

Rebuild Bootloader Options

Grub

```
update-grub
```

systemd-boot

```
bootctl update
```

Proxmox

```
pve-efiboot-tool refresh
```

Revision #5
Created 22 April 2020 11:16:55 by Dustin Sweigart
Updated 2 April 2021 14:03:54 by dustin@swigg.net