

Using GPUs in KVM Virtual Machines

✍ PAVAN / 📅 JUNE 23, 2016 /

📁 [HARDWARE \(HTTPS://ARRAYFIRE.COM/CATEGORY/HARDWARE/\)](https://arrayfire.com/category/hardware/),

📁 [INFRASTRUCTURE \(HTTPS://ARRAYFIRE.COM/CATEGORY/INFRASTRUCTURE/\)](https://arrayfire.com/category/infrastructure/),

📁 [OPEN SOURCE \(HTTPS://ARRAYFIRE.COM/CATEGORY/OPEN-SOURCE-2/\) /](https://arrayfire.com/category/open-source-2/)

💬 [2 COMMENTS \(HTTPS://ARRAYFIRE.COM/USING-GPUS-KVM-VIRUTAL-MACHINES/#COMMENTS\)](https://arrayfire.com/using-gpus-kvm-virtual-machines/#comments)

Introduction

A couple of months ago, I began investigating GPU passthrough on my workstation to test ArrayFire on different operating systems. Around the same time, we at ArrayFire found ourselves with a few surplus GPUs.

Having had great success with my virtualization efforts, we decided to build a Virtualized GPU Server to utilize these GPUs. Building a Virtualized GPU Server alleviated one of the pain points at our company: We no longer need to swap GPUs or Hard Disks to test a new environment.

To maximize the number of GPUs we can put in a machine, we ended up getting a [Quantum TXR430-0768R](http://exxactcorp.com/index.php/solution/solu_detail/207) (http://exxactcorp.com/index.php/solution/solu_detail/207) from Exxact Computing which comes in a 4U form factor and supports upto 8x double width GPUs.



http://arrayfire.com/wp-content/uploads/2016/06/server_front.jpg



http://arrayfire.com/wp-content/uploads/2016/06/server_7_gpus_1.jpg

The GPUs used for this build include:

- [AMD Firepro S9150](http://www.amd.com/en-us/products/graphics/workstation/firepro-remote-graphics/s9150) (<http://www.amd.com/en-us/products/graphics/workstation/firepro-remote-graphics/s9150>) x3
- [AMD Firepro S7150](http://www.amd.com/en-us/products/graphics/server/s7150) (<http://www.amd.com/en-us/products/graphics/server/s7150>)
- [NVIDIA Tesla K80](http://www.nvidia.com/object/tesla-k80.html) (<http://www.nvidia.com/object/tesla-k80.html>)
- [NVIDIA Tesla K40](http://www.nvidia.com/object/tesla-workstations.html) (<http://www.nvidia.com/object/tesla-workstations.html>)
- [NVIDIA Tesla K10](http://www.nvidia.com/content/tesla/pdf/tesla-kseries-overview-lr.pdf) (<http://www.nvidia.com/content/tesla/pdf/tesla-kseries-overview-lr.pdf>) (PDF)

Special Thanks to AMD and NVIDIA for providing us with the GPUs used in this build.

Note: If you are planning to purchase hardware for PCI passthrough, ensure both your motherboard and the processor support IOMMU. A short (but by no means comprehensive) list [can be found here](https://en.wikipedia.org/wiki/List_of_IOMMU-supporting_hardware) (https://en.wikipedia.org/wiki/List_of_IOMMU-supporting_hardware).

Setting up PCI passthrough

This section describes how to setup PCI passthrough on Ubuntu 16.04. This is based on [this arch wiki article](https://wiki.archlinux.org/index.php/PCI_passthrough_via_OVMF) (https://wiki.archlinux.org/index.php/PCI_passthrough_via_OVMF) about the same topic.

Setup machine for virtualization



- Add `intel_iommu=on` to the kernel parameter in grub
- Edit `/etc/default/grub` and add `intel_iommu=on` (or `amd_iommu=on`, if you have AMD) to the `GRUB_CMDLINE_LINUX_DEFAULT` line.
- Make sure you create the new `grub.cfg` file by running the following command

```
1
2 sudo grub-mkconfig -o /boot/grub/grub.cfg
```

- Blacklist `nouveau`, `radeon`, `amdgpu`
- create a file called `/etc/modprobe.d/blacklist-gpu.conf` and add the following lines

```
1
2 blacklist nouveau
3 blacklist radeon
4 blacklist amdgpu
```

- This step may break GUI login on your server machine. Make sure you have an SSH or VNC login to your server.
- Reboot

Adding Devices for PCI passthrough

- Run the following command to get the list of IOMMU groups

```
1
2 for iommu_group in $(find /sys/kernel/iommu_groups/ -maxdepth 1 -mindepth 1 -type d); do echo "IOMMU group $(basename "$iommu_group")"; for device in $(ls -1 "$iommu_group"/devices/); do echo -n '$\t'; lspci -nns "$device"; done; done
```

- The PCI IDs are of the form `[abcd:xyzw]`
- Add the PCI ids you want to pass through to `/etc/modprobe.d/vfio.conf`. Create the file if it does not exist.
- A sample configuration is given below.

```
1
2 options vfio-pci ids=10de:1024,10de:102d
```

- Add the following vfio modules to `/etc/initramfs-tools/modules`

```
1
2 vfio
3 vfio_iommu_type1
4 vfio_pci
5 vfio_virqfd
```

- Run `sudo update-initramfs -u` to generate the new initrd image.

- Reboot



Setting up QEMU + KVM + libvirt for PCI passthrough

- Install the required virtualization packages

```
1
2 sudo apt-get install qemu-kvm ovmf bridge-utils uml-utilities libvirt-bin
   virt-manager
```

- Add the user to the `kvm` and `libvirtd` groups

```
1
2 sudo usermod -a -G kvm userid
3 sudo usermod -a -G libvirtd userid
```

- Edit `/etc/libvirt/qemu.conf` and add (or uncomment) the following line if not already present

```
1
2 nvram = [    "/usr/share/OVMF/OVMF_CODE.fd:/usr/share/OVMF/OVMF_VARS.fd"
3 ]
```

- restart libvirt-bin service

```
1
2 sudo systemctl restart libvirt-bin
```

- You are now ready to pass through the GPU!

Setting up the virtual machines

This section describes how to set up and maintain VMs using `libvirt` in the frontend and `qemu+kvm` in the backend. This can be achieved by either using the command line (`virsh`) and GUI (`virt-manager`).

Creating a storage pool

If you plan on using the default pool, skip this section.

- For creating a pool from command line using `virsh`, [follow these instructions](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Administration_Guide/ch13s03s0) (https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Administration_Guide/ch13s03s0).
- For creating a pool using `virt-manager`, [follow these instructions](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Administration_Guide/sect-Virtualization-Storage_Pools-Creating-Local_Directories.html#idp13442168) (https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Administration_Guide/sect-Virtualization-Storage_Pools-Creating-Local_Directories.html#idp13442168).

Creating a volume



- For creating a volume from the command line using `virsh`, use the following commands

```
1
2 virsh vol-create-as --pool my-storage-pool --name disk-01.qcow2 --capacity 80G
   --format qcow2
```

- For creating a volume using `virt-manager`, double click on your connection,

navigate to storage, choose your pool, press `+` and follow the instructions.

Networking

`libvirt` defaults to creating a NAT interface for networking. If you want to use networking through a NAT interface, you can skip this section.

The rest of the section deals with creating a bridge interface which can later be used by the VMs.

- Edit `/etc/network/interfaces`
- Add the following lines towards the end. Use the appropriate network interface instead of `eth1`

```
1
2 auto br0
3 iface br0 inet dhcp
4     bridge_ports eth1
```

- Restart the network

```
1
2 systemctl restart networking network-manager
```

Installing an OS

Installing an OS from command line

- Run a command similar to the following

```
1
2 virt-install --name test-vm-01 --memory 8192\
3 --vcpus 4 --disk /my/storage/location/disk-01.qcow2\
4 --graphics vnc,password=foobar,listen=0.0.0.0\
5 --cdrom /path/to/os-install.iso\
6 --network bridge=br0\
7 --host-device=pci_abcd_vw_xy_z
```



- Note the pci id passed to `host-device` is in a slightly different format than the output from `lspci`
- The `vw_xy_z` format is extracted from the output of `lspci` which returns the ids in `vw:xy.z` format.
- You can now run the following command to get the appropriate pci id for `virsh`

```
1  
2 virsh nodedev-list | grep vw_xy_z
```

- Note, `listen=0.0.0.0` makes the vm visible to everyone. Mask it to the required subnets appropriately.
- change the password to be something more appropriate.
- Once the command is run, you can query the `vncdisplay` using the following command:

```
1  
2 virsh vncdisplay test-vm-01
```

- You can then use this information to connect to the server using a vnc client of choice

Installing an OS using `virt-manager`

- Click on create new virtual machine (top left icon)
- `Local install media`. Browse and choose install media
- Optional: Set OS type and version appropriately
- Choose the desired amount of RAM and CPU cores
- Expand `Network selection` and make sure the Bridge `br0` is selected
- `Select or create custom storage`. Browse to the storage disk created earlier
- Check the box that says **Customize configuration before install**
- Click on `Add Hardware`
- Choose `PCI Host Device` and select the device you want to pass through
- Click `Begin Installation` on top left corner.

If the GPU being passed through has audio support, you need to add it to the VM as well.

Accessing the VMs

- If you setup the VM to use bridge networking, it can be accessed directly via the ip within the LAN.

- If you setup the VM to use VNC, you can access the VM using any VNC client software.
- If you are on Linux, you can install `virt-manager` and connect to the remote server and all of it's VMs.



Cloning the VMs

- From command line run the following command

```
1
2 virt-clone --original original-vm-name --name new-vm-name --file
/path/to/new/disk.qcow2
```

- Alternatively, from `virt-manager` simply right click on the vm and choose clone.

Conclusion

Although time consuming, this build has been very fruitful for us. It allows us to seamlessly snapshot, clone, restore, and switch between various configurations. We plan on doing interesting things with this setup including integrating it with our continuous integration (<http://ci.arrayfire.org/>) system.

We also have other interesting ideas brewing internally to leverage this infrastructure. Follow this blog for updates in this regard!



Related Posts:



[AMD \(HTTPS://ARRAYFIRE.COM/TAG/AMD/\)](https://arrayfire.com/tag/amd/)

[GPU \(HTTPS://ARRAYFIRE.COM/TAG/GPU/\)](https://arrayfire.com/tag/gpu/)

[KVM \(HTTPS://ARRAYFIRE.COM/TAG/KVM/\)](https://arrayfire.com/tag/kvm/)

[LIBVIRT \(HTTPS://ARRAYFIRE.COM/TAG/LIBVIRT/\)](https://arrayfire.com/tag/libvirt/)

[NVIDIA \(HTTPS://ARRAYFIRE.COM/TAG/NVIDIA/\)](https://arrayfire.com/tag/nvidia/)

[PASSTHROUGH \(HTTPS://ARRAYFIRE.COM/TAG/PASSTHROUGH/\)](https://arrayfire.com/tag/passthrough/)

[PCI \(HTTPS://ARRAYFIRE.COM/TAG/PCI/\)](https://arrayfire.com/tag/pci/)

[QEMU \(HTTPS://ARRAYFIRE.COM/TAG/QEMU/\)](https://arrayfire.com/tag/qemu/)

[VIRTMANAGER \(HTTPS://ARRAYFIRE.COM/TAG/VIRTMANAGER/\)](https://arrayfire.com/tag/virtmanager/)



LOG IN WITH

OR SIGN UP WITH DISQUS **Kendrick** • a year ago

Question with the K80 or any of those for that matter are you able to split that between both virtual desktop acceleration and opencl/cuda usage at the same time? the info on AMD's site is a bit lacking.

^ | ▾ • Reply • Share >

**pete** • 2 years ago

How do you get around nvidia's driver code 43 error in Windows using this method?

^ | ▾ • Reply • Share >

ALSO ON ARRAYFIRE | GPU AND ACCELERATOR BLOG

OpenCL on Intel HD / Iris graphics on Linux

15 comments • 3 years ago

Ryang Sohn — Well, the Beignet supports Haswell on linux kernel >=4.0, although v4.1 has to enable a boot parameter.

Visit ArrayFire at GTC 2015

1 comment • 3 years ago

ArrayFire — Don't forget to swing by our booth!

Building an ArrayFire program with CMake

3 comments • 3 years ago

bkloppenborg — Thank you very much. Somehow our blogging platform stripped out the text! I have restored it to its correct state.

ArrayFire is Now Open Source

2 comments • 3 years ago

Shehzan Mohammed — We look forward to your speed up stories! Feel free to ask for

Follow this blog: [Subscribe via RSS \(http://feeds.feedburner.com/arrayfire\)](http://feeds.feedburner.com/arrayfire)**Recent Posts**[ArrayFire v3.5.1 Release \(https://arrayfire.com/arrayfire-v3-5-1-release/\)](https://arrayfire.com/arrayfire-v3-5-1-release/)

[ArrayFire v3.5 Official Release](https://arrayfire.com/arrayfire-v3-5-official-release/) (<https://arrayfire.com/arrayfire-v3-5-official-release/>)



[Visit ArrayFire at GTC 2017](https://arrayfire.com/visit-arrayfire-at-gtc-2017/) (<https://arrayfire.com/visit-arrayfire-at-gtc-2017/>)

[Thrombotherm: Analyzing Blood Platelets with ArrayFire](https://arrayfire.com/thrombotherm-analyzing-blood-platelets-with-arrayfire/) (<https://arrayfire.com/thrombotherm-analyzing-blood-platelets-with-arrayfire/>)

[ArrayFire at SC16](https://arrayfire.com/arrayfire-at-sc16/) (<https://arrayfire.com/arrayfire-at-sc16/>)

Categories

[Android](https://arrayfire.com/category/android/) (<https://arrayfire.com/category/android/>)

[Announcements](https://arrayfire.com/category/announcements/) (<https://arrayfire.com/category/announcements/>)

[ArrayFire](https://arrayfire.com/category/arrayfire/) (<https://arrayfire.com/category/arrayfire/>)

[Benchmarks](https://arrayfire.com/category/benchmarks-2/) (<https://arrayfire.com/category/benchmarks-2/>)

[C/C++](https://arrayfire.com/category/cc/) (<https://arrayfire.com/category/cc/>)

[Careers](https://arrayfire.com/category/careers-2/) (<https://arrayfire.com/category/careers-2/>)

[Case Studies](https://arrayfire.com/category/case-studies/) (<https://arrayfire.com/category/case-studies/>)

[Computer Vision](https://arrayfire.com/category/computer-vision-2/) (<https://arrayfire.com/category/computer-vision-2/>)

[Computing Trends](https://arrayfire.com/category/computing-trends/) (<https://arrayfire.com/category/computing-trends/>)

[CUDA](https://arrayfire.com/category/cuda-2/) (<https://arrayfire.com/category/cuda-2/>)

[Events](https://arrayfire.com/category/events/) (<https://arrayfire.com/category/events/>)

[Fortran](https://arrayfire.com/category/fortran/) (<https://arrayfire.com/category/fortran/>)

[Hardware](https://arrayfire.com/category/hardware/) (<https://arrayfire.com/category/hardware/>)

[Image Processing](https://arrayfire.com/category/arrayfire/image-processing-arrayfire/) (<https://arrayfire.com/category/arrayfire/image-processing-arrayfire/>)

[Infrastructure](https://arrayfire.com/category/infrastructure/) (<https://arrayfire.com/category/infrastructure/>)

[Jacket](https://arrayfire.com/category/jacket/) (<https://arrayfire.com/category/jacket/>)

[JAVA \(https://arrayfire.com/category/java-2/\)](https://arrayfire.com/category/java-2/)

[Open Source \(https://arrayfire.com/category/open-source-2/\)](https://arrayfire.com/category/open-source-2/)

[OpenCL \(https://arrayfire.com/category/opencl/\)](https://arrayfire.com/category/opencl/)

[OpenGL \(https://arrayfire.com/category/opengl/\)](https://arrayfire.com/category/opengl/)

[R \(https://arrayfire.com/category/r/\)](https://arrayfire.com/category/r/)

[Webinar \(https://arrayfire.com/category/webinar/\)](https://arrayfire.com/category/webinar/)



Archives

Archives

Select Month



[ABOUT \(HTTPS://ARRAYFIRE.COM/COMPANY/\)](https://arrayfire.com/company/)

[PARTNERS \(HTTPS://ARRAYFIRE.COM/PARTNER-PROGRAM/\)](https://arrayfire.com/partner-program/)

[LEGAL \(HTTPS://ARRAYFIRE.COM/LEGAL/\)](https://arrayfire.com/legal/) [TWITTER \(HTTP://TWITTER.COM/ARRAYFIRE\)](http://twitter.com/arrayfire)

[CONTACT \(HTTP://ARRAYFIRE.COM/COMPANY/#CONTACT\)](http://arrayfire.com/company/#contact) [SHOP \(HTTPS://ARRAYFIRE.COM/SHOP/\)](https://arrayfire.com/shop/)

[LOG IN \(HTTPS://ARRAYFIRE.COM/LOGIN/\)](https://arrayfire.com/login/) [REGISTER \(HTTPS://ARRAYFIRE.COM/REGISTER/\)](https://arrayfire.com/register/)



[\(twitter.com/@arrayfire\)](https://twitter.com/arrayfire)



