Subject: Direct Access to GPU
Posted by Boisterous on Wed, 30 Jul 2014 16:36:43 GMT
View Forum Message <> Reply to Message

I'm looking to run a graphical application inside a container, but it doesn't use the GPU when it's launched, it uses software rendering which is painfully slow. When I try to install the GPU driver I get an error message saying that no suitable adapters were found.

I'm able to launch X server and connect through VNC. I added the pci rail with the GPU on it, so when I run 'lspci' inside the container, the card shows up. But if I try to run 'dmidecode' I get an error saying that the operation is not permitted.

I noticed on the wiki that it states, "direct access to hardware is not available by default." Does this mean that there's a way to grant full hardware access? Any help would be appreciated.

Just to clarify, I've been able to run the setup on the hardware environment already. So the driver and software that I'm using is all correct.

Subject: Re: Direct Access to GPU
Posted by devonblzx on Wed, 06 Aug 2014 01:36:00 GMT
View Forum Message <> Reply to Message

I'm not too knowledgeable on GPUs in Linux but I can try to help.

First, make sure the drivers and kernel modules are loaded on the host node. You won't be able to install a driver from inside the container but you can probably install a utility to manage the device once your container has access to it.

Is there a corresponding /dev device like /dev/nvidia0? If so, then grant the container access to that device. You may also need specific features granted as well, but I'm not sure on that.

Most users don't use OpenVZ for GUI. Here is the man page of vzctl if you need help with devices/features: http://openvz.org/Man/vzctl.8#Device_access_management

The one problem I see with this is OpenVZ is OS level virtualization, not hardware virtualization. Therefore you probably won't be able to have both the host and the container using the GPU. I'm not sure how that would work and I'm not sure even in hardware virtualization how the GPU is virtualized.