

Scotti-BYTE Enterprise Consulting Services

Thinking Outside the Card: GPU's

Traditionally, video cards (aka. display card, display adapter or graphics adapter) is an expansion card which generates images that are passed to a display device such as a monitor. There have been a huge series of standards like CGA, VGA, EGA, XGA and many others which were introduced beginning in 1982 and supported by a variety of manufacturers.

As an alternative to the video card, the video card hardware can be combined into the motherboard, in which case the motherboard is referred to as having integrated graphics. Even when a motherboard provides integrated graphics, often the integrated graphics are disabled in favor of a higher performance external video card.



This tutorial is not about graphics display adapters or 3 dimensional rendering per se. Display adapters have evolved tremendously and are now touted as graphics processing units (GPU). A GPU is a specialized device designed to manipulate memory in images referred to as frame buffers and deliver these images rapidly to a display device.

Today, GPUs are primarily manufactured by Intel, Nvidia, and AMD/ATI. GPUs are like the CPU in a computer, except that GPUs were generally purposed towards calculations to display 3D graphics, 2D drawing acceleration, and framebuffer capabilities. Today most GPUs feature larger amounts of very fast memory with many cores and run with many parallel threads to provide extremely high performance. In addition, todays video display adapters are connected to the PCI Express bus which is a very high speed parallel bus architecture. Most video cards operate on the x16 bus which uses 16 lanes (x16) of simultaneous traffic from the main system bus to and from main memory.

The PCI Express bus was initially introduced in 2003 and has continued to evolve and improve ever since. As an example, the most prevalent PCI Express standard was introduced as PCI

Express 3.0 in 2010 and on an x16 bus slot it provides a throughput of 15.75GB/s. PCI-X 4.0 in 2017 increased that to 31.5GB/s and PCI-X 5.0 in 2019 increased to 63GB/s. Current display adapters regularly contain as much as 2TB of VRAM memory which is typically either DDR4 or GDDR5. High Bandwidth Memory (HBM) was first used by AMD GPUs in 2015. HBM achieves higher bandwidth while using less power in a substantially smaller form factor than DDR4 or GDDR5 because it stacks DRAM dies creating a kind of 3D circuit.

All the power in these GPUs is traditionally used to drive very high resolution displays depending very fast frame rates for 3D graphics, modelling, and gaming.

It wasn't long before it was noticed that these GPUs were a tremendous resource that could be used for other than graphics. Even though crypto-currency is getting more and more difficult to mine, CGMiner and BFGMiner both take advantage of the huge amount of threads that today's GPUs have to offer.

Sometimes display adapter performance can be used to render graphics and execute simulation applications without actually being physically connected to a monitor. One popular use of powerful GPUs is Folding@home which is a distributed computing project for simulating protein dynamics, including the process of protein folding and the movements of proteins implicated in a variety of diseases. It brings together citizen scientists who volunteer to run simulations of protein dynamics on their personal computers.

Plex Media Server often needs to convert the video to a different quality or a compatible format. Converting the video (transcoding) happens automatically, in real-time, while you're playing it. To convert videos faster and with less processing power, you can turn on Hardware-Accelerated Streaming in Plex Media Server. When hardware acceleration is turned on, Plex Media Server will use the video decoder and encoder hardware support on your GPU to convert videos, letting you stream HD or 4K video more smoothly and stream to more devices at once.

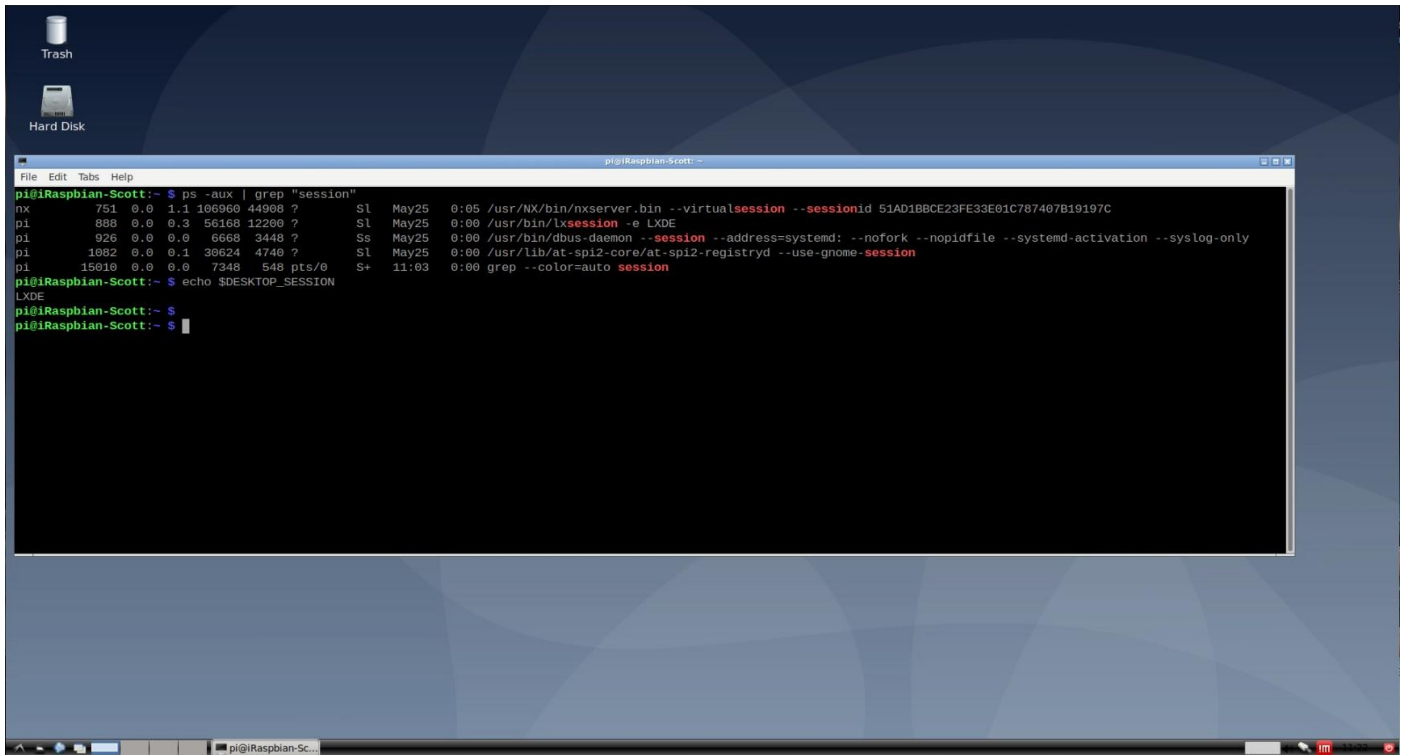
Many people use computers such as the Raspberry Pi, the Apple Mac Mini, and even desktop computers in a "headless" mode to host Network Attached Storage (NAS), music servers, and other applications. If you want/need to access the graphical user interface (GUI) of a modern desktop on a headless machine there are protocols such as Remote Desktop Protocol (RDP), Virtual Network Computing (VNC), Simple Protocol for Independent Computing Environments (SPICE) and NX Technology (NX).

The problem with remote desktop access and a headless computer is that in many cases, the actual generation of the screen display uses the GPU hardware even though the computer might not have a monitor connected to it. Most modern computers will not actually enable their GPU if a monitor is not physically attached to the computer.

I discovered this behavior recently while trying to use the NX protocol with the "Nomachine" software to remotely access a Raspberry Pi 4. I also used VNC to access the Pi 4 and as long as an HDMI monitor was connected, I was seeing a very customized display. If I booted with no monitor connected to the HDMI (headless), I was seeing a basic LXDE desktop without the "extras". I started to look at what was not displaying and I noticed that the custom top panel and the "Plank" dock were missing from my interface as well as several other elements that were being

generated with the use of the Raspberry Pi 4 Integrated GPU on the motherboard. Despite the fact that this is not a very powerful GPU, without a monitor plugged in, the GPU effectively went to sleep and could not provide the needed capabilities as long as the Pi 4 was running headless.

I kept thinking that perhaps that with the HDMI connected monitor that I was running a different display and/or window manager. As it turns out, both the HDMI connected monitor and the NX remote session running headless were both running an LXDE desktop. The difference was that the headless configuration could not engage the needed GPU to activate all of the screen elements.

A screenshot of a Raspberry Pi desktop environment. The desktop background is a dark blue geometric pattern. In the top-left corner, there are icons for 'Trash' and 'Hard Disk'. A terminal window titled 'pi@Raspbian-Scott: ~' is open in the center. The terminal shows the output of the command 'ps -aux | grep "session"', displaying several processes related to the NX session and LXDE. The terminal also shows the output of 'echo \$DESKTOP_SESSION', which is 'LXDE'. The terminal window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The bottom of the screen shows a taskbar with icons for the terminal, a file manager, and other applications.

```
pi@Raspbian-Scott:~$ ps -aux | grep "session"
nx      751  0.0  1.1 106960 44908 ?        Ssl  May25   0:05 /usr/NX/bin/nxserver.bin --virtualsession --sessionid 51AD1BBCE23FE33E01C787407B19197C
p1      888  0.0  0.3  56168 12280 ?        Sl   May25   0:00 /usr/bin/lxsession -e LXDE
p1      926  0.0  0.0   6668  3448 ?        Ss   May25   0:00 /usr/bin/dbus-daemon --session --address=systemd: --nofork --nopidfile --systemd-activation --syslog-only
p1     1082  0.0  0.1  38624  4740 ?        Sl   May25   0:00 /usr/lib/at-spi2-core/at-spi2-registryd --use-gnome-session
p1     15010 0.0  0.0   7348   548 pts/0    S+   11:03   0:00 grep --color=auto session

pi@Raspbian-Scott:~$ echo $DESKTOP_SESSION
LXDE
pi@Raspbian-Scott:~$
```

If I plugged in the HDMI monitor prior to boot, not only did the HDMI connected monitor display all the screen elements, but the Nomachine NX session also displayed properly. If I rebooted without the HDMI connected, the nomachine session appeared as above lacking the "features".

I've dealt with this behavior before and thought that perhaps certain X-Windows session managers just needed to run on a graphics card and not on a virtual display.

In industry, it is common to offer virtual desktop interfaces (VDI) through applications such as Citrix. The idea is to have a thin desktop client access a blade server or even dedicated blade to provide a powerful, but data center centralized desktop. This evolved to the point that these data center servers were even hosting powerful graphics adapters to support 3D modelling and simulation applications.

Another important use of graphics adapters is virtual machines. A virtual machine hypervisor can utilize an installed graphics adapter on the virtual machine host to accelerate the display on a VM. QNAP's Virtualization Station can use hardware acceleration on a video card installed in the NAS in this way.

Often, the graphics adapters on these blade servers would either "fall asleep" or be dormant at boot time because the servers on which they were installed were headless. There is a solution for this problem.

There is a device called an HDMI Dummy Plug which is a small device that fits on to the HDMI output of your display card for those systems that you wish to run headless. When you plug the dummy plug into your display adapter, it fools your computer into thinking that a high resolution display has been attached and as a result, your computer unleashes it's full graphical potential. So, when you plug the dummy plug in, you're actually simulating the presence of an attached display which in turn allows you to use all the processing power and resolution that your graphics hardware is capable of providing.



So, as I've mentioned above, many computers don't actually enable their GPU hardware until a monitor is attached. However, by plugging in a dummy plug into your HDMI port, it can activate your graphics card and it will simulate up to a 4K display for you.

I purchased a 3-pack of the HDMI dummy plugs pictured for \$10 on Amazon. This is a surprisingly simple solution to a often encountered problem. Since the Raspberry Pi 4 uses a Micro HDMI video connector, I used an Micro HDMI plug to a standard HDMI adapter to connect the HDMI Dummy plug as shown in the upper left of the picture below.



As soon as I plugged in the HDMI Dummy plug and rebooted the Raspberry Pi 4, I used the Nomachine client to connect to it. The Raspberry Pi came up with all the screen elements as expected. Not only that, but sound from the Raspberry Pi 4 was working again. I believe that the sound was working because the Raspberry Pi 4 motherboard integrated graphics is HDMI based and by default will route sound to the HDMI port. So, once the HDMI Dummy Plug was in place, sound was working as well. Here is a picture of the completed desktop connected via Nomachine.



The image above is actually Raspbian running on the Raspberry Pi 4 skinned to look like Mac OSX Catalina.

In summary, it appears that a video card installed and not connected to a monitor will, in many cases, go dormant and not allow its GPU to be used by ancillary applications as described in this tutorial. An HDMI Dummy Plug can allow the display adapter to remain active so that virtual displays, transcoding processing, and other external applications can make use of a display adapter on a computer that is configured to be headless.