

Using VirtualGL on ICTLab Servers

SonTG - ICTLab, USTH

November 2017

Introduction

Most ICTLab servers are equipped with NVIDIA GPUs for GPU-accelerated computations. These GPUs provide a foundation for local-scale High Performance Computing platform. Most of current researches at ICTLab focus on the computing capabilities of these GPUs in different areas, including (but not limited to) model training and inference in machine learning, real time computer vision and image processing.

The above research axes of ICTLab have yet to utilize the rendering capabilities of these GPUs using OpenGL. This is a common limitation as all accesses to the servers are provided using SSH with key pair authentication. Users are not provided with direct access to the physical servers for security purposes.

In order to explore possibilities of rendering on these GPUs (which is, of course, their main strength), we start installing and utilizing “VirtualGL”, an OpenGL *proxy* for OpenGL-based applications. Additionally, second week of the master course MI 3.22 “Advanced Programming in HPC” for M2 students at USTH also uses OpenGL (GLU/GLUT) for rendering output of the images.

This small tutorial aims at providing a small how-to to ICTLab researchers and students in order to use hardware-accelerated OpenGL applications on ICTLab servers.

VirtualGL

OpenGL (Open Graphics Library) is a cross-platform API allowing applications to render 2D and 3D vector graphics. It utilizes hardware acceleration (on GPUs) if possible and can fallback to software rendering. However, the latter is quite limited in performance and features and should be avoided for real world applications.

On the other hand, since SSH with keypair authentication is a secured mechanism to provide access to the servers, we enforce this way of connection to all ICTLab

researchers and students. Nevertheless, an SSH-based user session is not allowed to access the hardware-accelerated OpenGL. We rely on an OpenGL *proxy* application called “VirtualGL” to provide hardware acceleration to remote users.

VirtualGL can be operated in two different ways: X11 forwarding and VNC. The former requires low latency, high bandwidth network (Ethernet), and therefore does not satisfy the requirement of remotely work on the servers (at home, for example). Finally, we decide to follow the VNC mode.

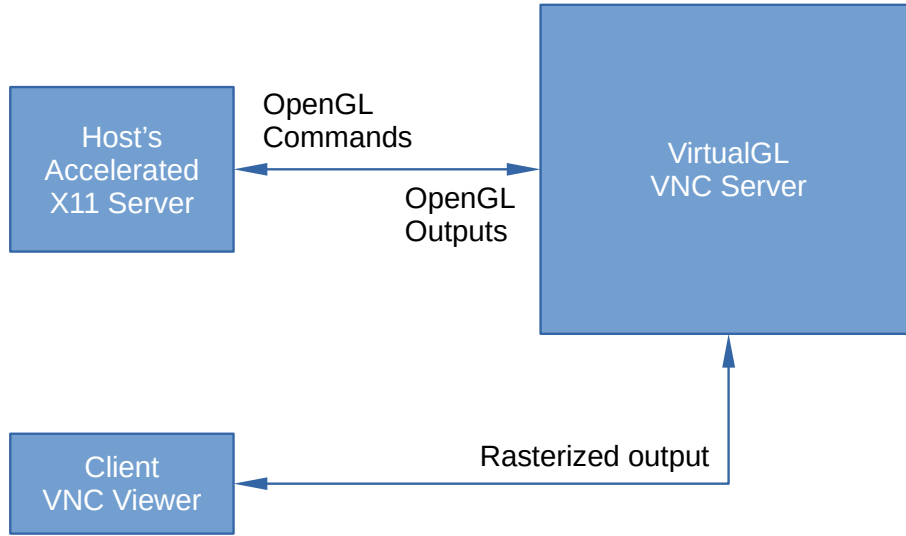


Figure 1: VirtualGL

In this mode, VirtualGL provides a **vncserver** that provides *fake* OpenGL capabilities. The normal unaccelerated rendering commands are handled by itself, while OpenGL rendering requests to this VNC session is then redirected to a physical, hardware-accelerated X11 session (usually on :0.0). The rendering results are then rasterized to the VNC session and forwarded back to the VNC viewer. The VNC viewer can be of any implementation. The requirements, however, is to have an active physical X11 session (even on login screen).

The authors of VirtualGL is kind enough to provide us a **TurboVNC** package that furtherly speedups the encoding and transporting process between VirtualGL **vncserver** and our laptop/desktop.

Enough. How to use it?

Starting VirtualGL vncserver

VirtualGL has already been installed on all ICTLab servers at `/opt/VirtualGL`.

- You need to start a VirtualGL's VNC server display before using it. Simply launch

```
you@ictserverX ~$ /opt/VirtualGL/bin/vncserver :10
```

Replace 10 with a number of your choice, between 1 and 20. Let's call it *YY*. This will be the display number that you will forward port and then connect to. If it complains about port being used, choose another value for *YY*.

- You will be asked for **VNC password** in the first time launching. Type a password of your choice. Don't share it, or anyone will be able to see what you're doing on your VNC screen!

Forward the VNC port

Now that VirtualGL's vncserver is running, you have a virtual VNC session. However, you will not be able to connect to it directly, since it's in a private network behind the **frontend** machine. We need a SSH port forward to reach this screen.

- Disconnect from the server **ictserverX** that you're currently SSHing to
- Disconnect from the **frontend**
- Reconnect to the **frontend** with port forwarding:

```
you@yourlaptop ~$ ssh -p 22222 you@ictlab.usth.edu.vn -L 59YY:localhost:59YY
```

YY is the display number that you have already chosen before (check last section).

- Reconnect again from the **frontend** to the server **ictserverX** (where you started your VirtualGL vncserver), with port forwarding:

```
you@frontend ~$ ssh ictX -L 59YY:localhost:59YY
```

Keep the SSH session open. Without it you will not be able to connect to your VNC session.

Connect to the VNC session

Next, since you have already forwarded the port 59YY from **ictserverX** to your local laptop/desktop, you can now make a VNC connection to

127.0.0.1:YY

This connection will be forwarded to port 59YY to the **frontend**, and in turns, to the port 59YY on **ictserverX**

You will be asked for password. Type the VNC password that you have chosen before.

A successful connection will show you a full featured LXDE desktop with icons and panels. You can start a terminal on this session with **xterm** or **lxterminal**.

Run your OpenGL application

- Any OpenGL application cannot start normally at the moment, you can check with

you@ictserverX ~\$ glxgears

- However, you can run it with a **vglrun** prefix, like so

you@ictserverX ~\$ vglrun glxgears

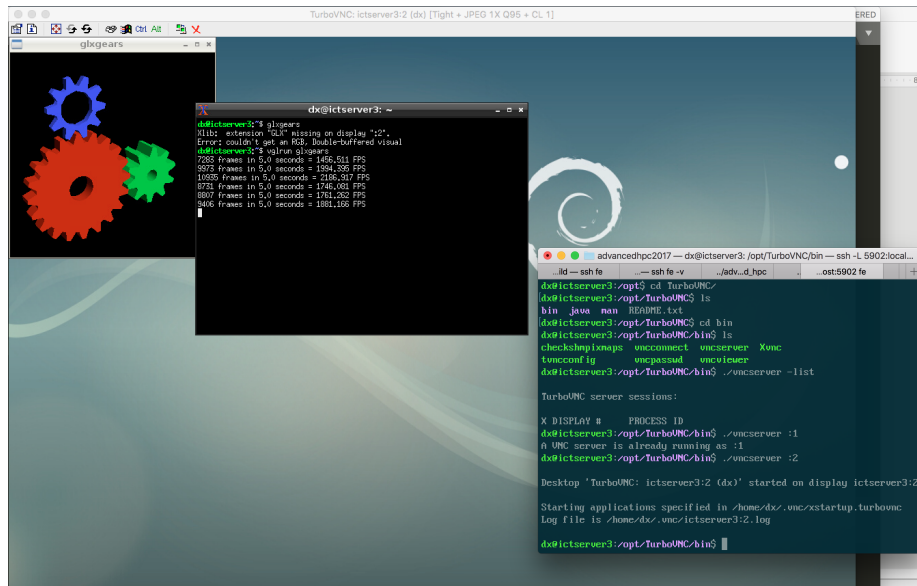


Figure 2:

Good Luck and Have Fun

SonTG @ICTLab

Nov 23rd, 2017.