

Remote Rendering on Visualization Cluster using VirtualGL and Chromium

Prepared by		Institution	
Srikanth Nagella		Science and Technology Facilities Council	
Lakshmi Sastry		Science and Technology Facilities Council	
Ronald Fowler		Science and Technology Facilities Council	
Verified by			
John O'Brien		Loughborough University	
Ref:	VIZNET-WP4-01-LU-Sample	Date:	12/10/2008

Remote Rendering on Visualization Cluster using VirtualGL and Chromium

Srikanth Nagella, Lakshmi Sastry and Ronald Fowler

Content:

1. Introduction
2. Chromium
3. VirtualGL
4. Visualization Cluster
5. Usage

1. Introduction

Data deluge from experimental facilities such as the LHC¹ (Linear Hadron Collider) instruments at CERN that generates data at the rate of 2GB/s to complex simulations and observational data pose visualization challenge to support real-time high resolution data analysis. The National Grid Service (NGS) and EGEE provide distributed, high performance storage and compute resources to support research communities for data management, simulation runs and data processing but not visualization and interactive data analysis. In general, until recently proprietary high performance graphics computers were used to analyse very large datasets in a post processing stage. Increasingly, high performance graphics cards, processors and high speed interconnect are used together to build graphics clusters similar to compute clusters. These generic systems are affordable and support open source software stack.

The vizNET STFC partners have installed a 17 node, 64-bit commodity cluster with dual processors on each node with Linux operating system. We installed a variety of visualization and other applications and test suites, met conformance testing and have gained NGS affiliation as an associate resource. This system is made available to the academic community under a joint STFC/vizNET/NGS banner.

Chromium and VirtualGL are the two software packages that are used for this purpose. Understanding the architecture these enhances the ability to use the system effectively. This document sets out the details of these open source applications that can be used to access our parallel visualisation services on the cluster from remote desktops, describes the hardware and how it can be used..

2 Open source software stack

2.1 Chromium

Chromium is an open source library for interactive parallel rendering of a single image using several rendering machines in parallel. There are several techniques in parallel rendering that will enable rendering large amount of data that cannot be rendered in real time using a single machine. Chromium is a completely extensible architecture and enables user to implement parallel rendering algorithms with ease. Most of the OpenGL applications will run under chromium without any modification to the application or recompiling it. To enable some of the features in chromium user needs to modify the application using Chromium API and recompile the application. The API is simple to use and minimal changes are required in the application code.

Chromium supports parallel rendering in Distributed Multi-Threaded X Server mode [Figure 1]. This enables rendering in high resolution tiled displays to visualise extremely large datasets in high resolution which is very useful for medical and engineering applications.

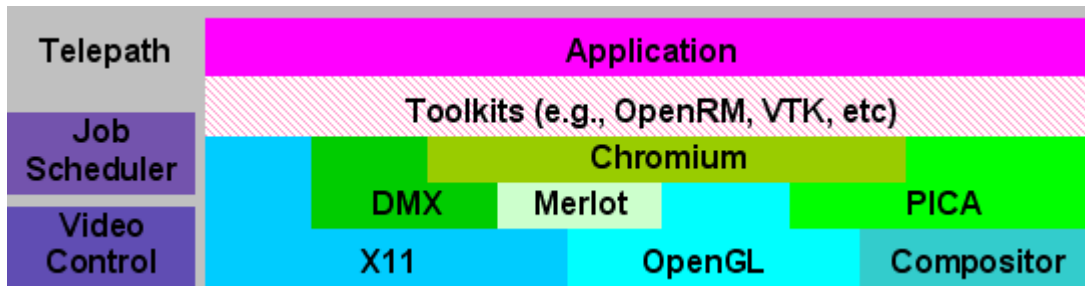


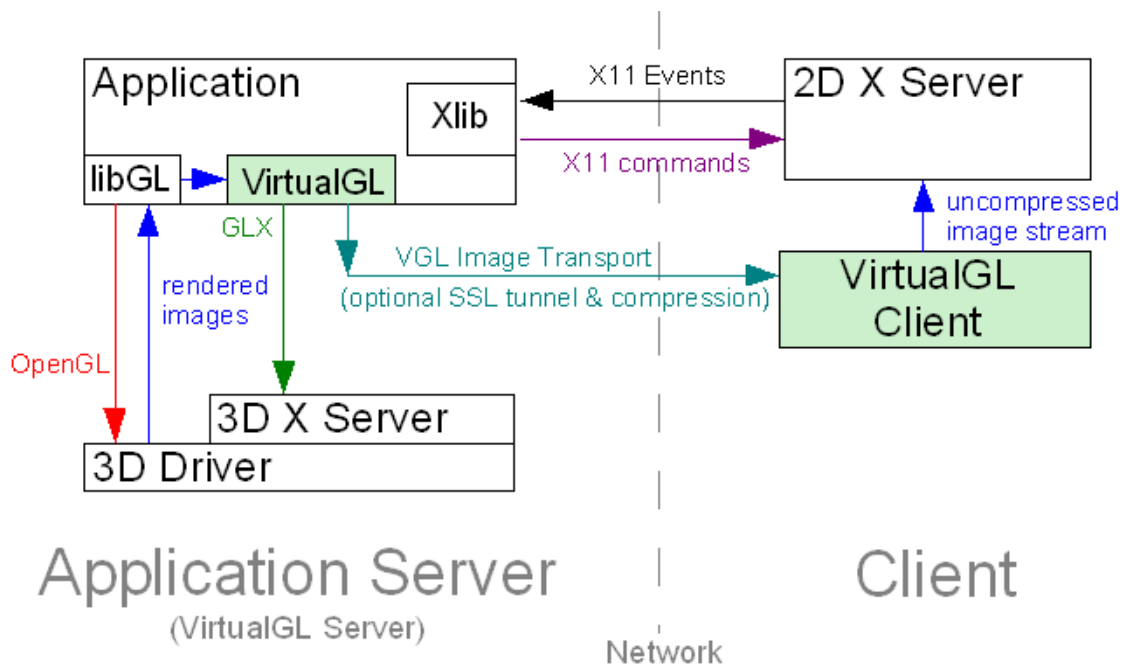
Figure 1 Architectural overview of Chromium with DMX.

2.2 VirtualGL³

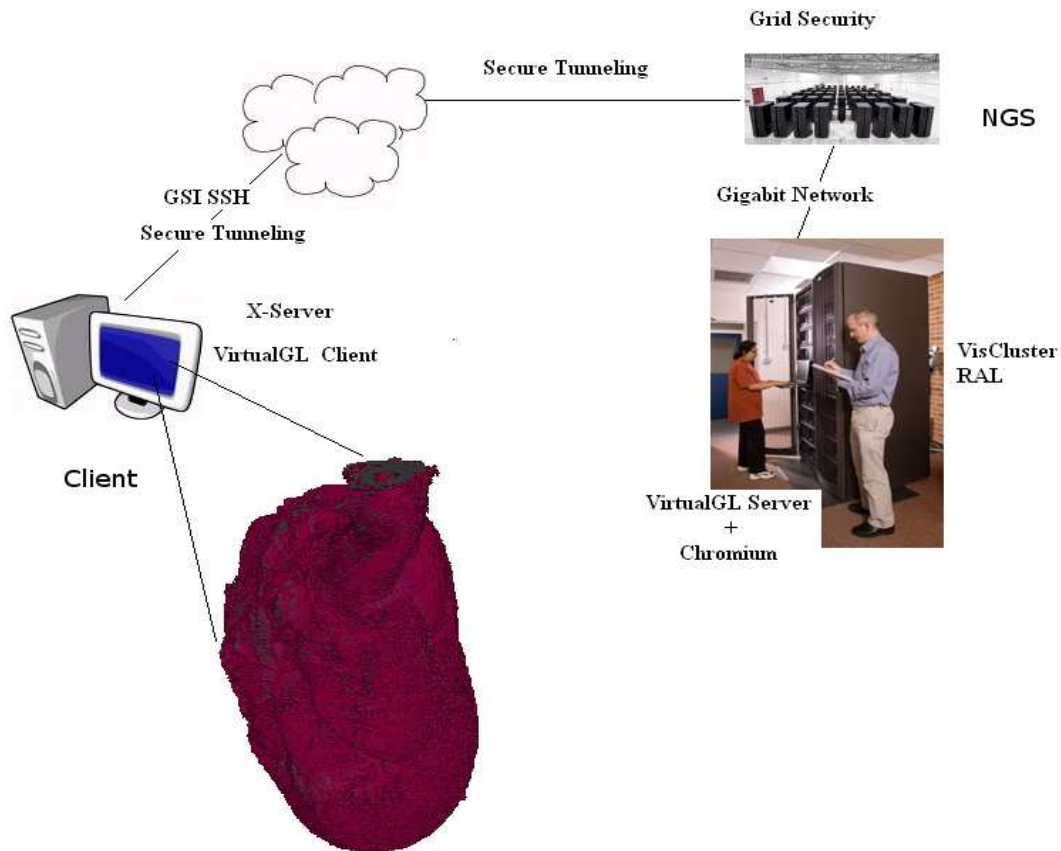
VirtualGL is an open source package which gives any UNIX or Linux remote display software the ability to run 3D applications with full hardware accelerations. VirtualGL can also be used in conjunction with remote display software such as VNC to provide 3D hardware accelerated rendering for OpenGL applications. VirtualGL is very useful in providing remote display to thin clients which lack the 3D hardware acceleration.

VirtualGL can be used in two modes, namely, VGL Image Transport and X11 Image Transport. In the VGL Image Transport mode, the user desktop X Server sends the user inputs to the X Server at the remote server for interactive rendering of the image and the rendered image is transported to the user desktop via a network channel other than X Server link. In X11 Image Transport mode the VirtualGL talks to a VNC like remote display software and renders the 3D applications and sends the entire remote desktop to the user desktop.

VGL Image Transport mode is the chosen, default mode with the visualization cluster as it gives the flexibility of multiple users sharing the cluster at the same time.



3. Visualisation Cluster



The visualization cluster is physically located at the Harwell Science and Innovation Campus in Didcot, Oxfordshire. The system is administered as part of the high performance compute clusters of NGS node and the Scientific Applications clusters which are NFS mounted to the system. It is linked with a 1Gbit to the NGS data sources and the compute resources. It is customised to allow either a number of simultaneous users or a single user at any given time.

The hardware of Vis Cluster consist of 17 HP workstations (xw9300) with Dual AMD Opteron 2GHz Processors, 12GB of RAM, 1 GByte Ethernet, 2 InfiniBand interconnects and high end graphics card NVIDIA FX4500 512MB Video RAM. The cluster is connected to network filesystem of 2.3TByte. The local display visualisation hardwares include a passive stereo screen, a high resolution 6 (3x2) panel display wall (1980x1280) and a desktop auto stereo monitor.

The applications installed on the cluster are NAG IRIS Explorer, VTK, Paraview, Chromium and several math libraries such as NAG numerical and data mining libraries, BLAS libraries and FFTW libraries. Software packages are installed as and when users request for specific packages.

The Operating system installed on all nodes is RHEL 4(Redhat Enterprise Linux). It uses NGS-2 grid resource. A user needs to login to the NGS-2 grid resource to access Visualization Cluster interactively. To enable multi user access, the cluster has LSF Interactive and batch scheduling software installed.

4. Step by step user guide to using the visualization cluster

4.1 Prerequisites:

- 4.1.1 Register with NGS at <http://www.grid-support.ac.uk/content/view/221/171/>
- 4.1.2 Register to use visualization cluster at <http://hpcsg.esc.rl.ac.uk/viscluster/user.html>
- 4.1.3 Install VirtualGL Client on user machine from <http://www.virtualgl.org/Downloads/VirtualGL>.
- 4.1.4 Start X Server installation on desktop. Linux PCs have built in X Server, Windows users need to install either HummingBird or equivalent X Server.
- 4.1.5 Install GSISSH Term on desktop from <http://www.grid-support.ac.uk/content/view/81/195/>.

4.2 Login:

- 4.2.1 Run GSISSH Term (Step 4.1.5).
- 4.2.2 Click on File->New Connection. A dialog box will appear.
- 4.2.3 Enter the Host to Connect to value as "ngs.rl.ac.uk" without quotes and click on OK.
- 4.2.4 Follow the instructions as in the link <http://www.grid-support.ac.uk/content/view/134/195/>

4.3 Simple OpenGL Application(glxgears):

- 4.3.1 Follow the login procedure as in 4.2
- 4.3.2 Start VirtualGL Client.(In windows at default location, C:\ProgramFiles\VirtualGL-2xxx\vglclient.exe this will automatically start hummingbird if not started)
- 4.3.3 In GSISSH Terminal Window, type: `visbsub -vglexec -n 1 -- /usr/X11R6/bin/glXgears`

This will print a message on screen such as

Using port number **58596** on ngs - please forward this port

If using gsissh, type:

~C

then: -R 58596:localhost:4242<return>

If using gsissh-term, set up an Incoming tunnel of
port 58596 to 4242

- 4.3.4 Click on menu Tools->Secure Tunneling. A window will appear, Click on "New Tunnel". Another window will appear prompting to enter

Listening Port(remote server): This will be the number as highlighted in the red in 4.3.3.

Destination Host: localhost

Destination Port: 4242

Then Click on Ok button and close the Secure Tunneling window.

- 4.3.5 Press enter key to continue.

- 4.3.6 A window will appear with three gears rotating. If you don't get this window then there is some problem with the configuration on desktop. One of the issue is with firewalls.

4.4 Starting Paraview:

- 4.4.1 Follow the login procedure as 4.2

- 4.4.2 Start VirtualGL Client.(In windows at default location, C:\ProgramFiles\VirtualGL-2xxx\vglclient.exe this will automatically start hummingbird if not started)

- 4.4.3 In GSISSH Terminal Window, type: visbsub -n 5 -vglscript -- pv261

You can change the number 5 to the required number of nodes to be used for paraview.

The above commad will print a message

This will print a message on screen such as

Using port number 58596 on ngs - please forward this port

If using gsissh, type:

~C

then: -R 58596:localhost:4242<return>

If using gsissh-term, set up an Incoming tunnel of
port 58596 to 4242

- 4.4.4 Click on menu Tools->Secure Tunneling. A window will appear, Click on “New Tunnel”. Another window will appear prompting to enter

Listening Port(remote server): This will be the number as highlighted in the red in 4.3.3.

Destination Host: localhost

Destination Port: 4242

Then Click on Ok button and close the Secure Tunneling window.

- 4.4.5 Press enter key to continue.
- 4.4.6 Paraview Client will appear in user desktop.

5. Conclusion

As part of qualifying for the Affiliate status, we have set up a formal HelpDesk and ticketing system which can be used for trouble shooting while getting started. The Scientific Applications Group also offer a limited hand-holding software and visualization expertise support to encourage take up by application developers. vizNET, NGS and STFC have been actively encouraging take up of the system. Initial indications are that the users require quite a lot of support to get using remote resources and then also help with adopting their data to the tools and visualization modules. The resource level required for such support is far beyond current provision. Given the user interest even at this beginning stage, it is obvious that advanced visualization for scientific productivity can be achieved with allocation of more resources to employ support staff.

References

1. **Deploying the LHC computing grid - the LCG service challenges**
Bird, I.; Robertson, L.; Shiers, J.
Local to Global Data Interoperability - Challenges and Technologies, 2005
Volume , Issue , 20-24 June 2005 Page(s): 160 – 165
2. **Chromium** <http://chromium.sourceforge.net/>
3. **VirtualGL** <http://www.virtualgl.org>
4. **RAL VisCluster** <http://hpcsg.esc.rl.ac.uk/viscluster/index.html>