PROGRESS REPORT

Performance Optimization on Display Wall

Student: Aaron Robinson

Advisor: Dr. Bruce Segee

Reporting Period: June 10 – July 8

Introduction

High-resolution data visualization is an import scientific tool which gives the viewer the ability to view the finer details while still having visual access to the big picture. In addition to the ability to see the finer details of a large data set, sometime researchers use the multiple displays to show different time steps of a smaller data set. This allows researchers the ability to see the transformation of a data set over some dimension while still allowing them the ability to study a single frame of that transformation ("LionEyes Display Wall"). Researchers at NASA use these displays to render data from complex problems like fluid dynamics and climate modeling ("NAS Technical Reports"). Similar problems are tackled at display walls at a number of other laboratories and universities. Beyond the scientific use of these displays, commercialization of such systems has already begun ("Video Wall Case Studies"). Some industries use complex models which are similar to those used by researchers but in addition to such applications, commercial display walls can be used for security, advertisement and entertainment. High-resolution data visualization is an emerging area of computer science which is quickly making its mark in academia and industry.

Background

There are two general frameworks for high-resolution data visualization. I prefer to call them direct rendering and indirect rendering. Here I define direct rendering to be the case where graphics are displayed on the card which renders them. In opposition, indirect rendering would be the case where graphics are rendered and then displayed on another system (presumably on another graphics card). Each framework has certain advantages. If direct rendering is used, a single machine does not dictate rendering capabilities of the cluster but if indirect rendering is used the back-end machine can be a thin client allowing for little to no configuration on that machine. In my research I have looked at implementing Xdmx and

Chromium to create a display wall using direct rendering. Xdmx is a proxy X server which connects to multiple back-end X servers ("DMX Homepage"). Unlike normal X servers which can only use multiple displays on a single machine, Xdmx can use multiple displays on multiple machines. Xinerama is also an important part to this structure because it allows the multiple displays to appear to the user as a single unified display in both multiple monitor configurations. However, Xdmx alone is not designed for 3D rendering. Chromium provides a structure for rendering on a graphics cluster. Chromium does not actually do the rendering but using its structure, it manipulates the OpenGL. In this setup Xdmx creates the unified display which when rendering is required is snatched by Chromium on the front-end. Chromium redirects it from the X proxy (front-end) to the individual X servers(back-end). Once there Chromium directs the rendering to OpenGL where it is rendered.

Progress Summary

I have successfully tested Xinerama by creating a dual0monitor configuration on a single machine. When working with Xdmx I found that the latest release (xdmx_1.6.1.901-3_i386.deb) has an issue where it causes a segmentation fault. To fix this issue I am using an older release (xdmx_1.1.1-21etch5_i386.deb). With the older version of Xdmx I have successfully created a Xdmx display on multiple machines. I have not attempted to combined these two separate components because this work seemed trivial in comparison to further research with Chromium. Chromium require XF4VNC for a few Shared Objects which I have successfully installed. I have been able to build Chromium from the latest CVS tree and from the 1.9 release. I was working with the CVS tree and ran into an issue because I was using Nvidia drivers. The particular issue was related to GLX and rendering to the graphics cluster using Chromium. However, I was able to test Chromium on a single machine without Xdmx and it worked properly. I am currently working with the 1.9 release without the Nvidia drivers. My first task will be to get the Chromium demo without Xdmx to run on a single machine.

References

Burris, Gavin. "LionEyes Display Wall." Penn State's Visualization Group . 8 Feb 2007 . Penn State. 7 Jul 2009 <http://viz.aset.psu.edu/ga5in/DisplayWall.html>.

"Chromium Render Server Setup." 25 May 2007. SourceForge. 7 Jul 2009 <http://vncproxy.sourceforge.net/setup.html>.

"DMX Homepage." 13 Jun 2004. SourceForge. 9 Jul 2009 <http://dmx.sourceforge.net/>.

Dunbar, Jill. "NAS Technical Reports." NASA Advanced Supercomputing (NAS) Division. 15 May 2009. NASA. 7 Jul 2009 <http://www.nas.nasa.gov/News/Techreports/techreports.html>.

"Video Wall Case Studies." Video Wall and Multi Monitor Solutions. CineMassive. 7 Jul 2009 <http://www.cinemassivedisplays.com/video-wall-case-studies.php>.