Abstract

This report contains the results of research focused on COTS clusters for use in interactive computer visualization applications. An existing software solution named Chromium is briefly described from both hardware and software architecture point of view. It is also compared to a current SGI OpenGL VizServer installation at NTNU in Trondheim, both in terms of features and performance. The question to be answered is whether it is possible to install a cost efficient cluster using COTS hardware, without discarding any major functionality and performance compared to the existing SGI system.

Background

ODIN is an SGI Onyx2 server with 32 CPUs and 2 rendering pipelines. It is equipped with 16 GB of memory and 1 GB texture memory for each pipe. The SGI OpenGL VizServer enables a client located at a remote site to run OpenGL applications in a way that the VizServer takes care of all graphics processing and simply returns the result (the picture) back to the client. The current installation at NTNU enables either one client to use both rendering pipelines or two clients using one pipeline each, running simultaneously. Any application compiled to target OpenGL may exploit the VizServer's full rendering capacity simply by running in the VizServer shell.

Requirements

- **APIs and remote accessibility:** The "ODIN" SGI Onyx2 consists of 32 CPUs and 2 rendering pipelines. Client applications usually access the system from a remote location through software APIs such as OpenGL or OpenIventor, running on standard workstations. Support for OpenGL, OpenInventor and remote accessibility is fundamental for a possible substitute for the VizServer.
- **Simultaneous users:** An application utilizing the VizServer is assigned either one or both of the graphic pipelines, resulting in the possibility of several users/clients working simultaneously. Support for remote accessibility and several simultaneous users is the second requirement for possible alternatives to the VizServer.
- **COTS hardware:** The cluster should be assembled out of Commercial Of The Shelf (COTS) hardware, with a cost efficient high-performance system as a result.
- **Interactivity and performance:** The response time and performance of the system must provide true interactivity.

Chromium

Introduction

Chromium [1] is an open source system that enables clients to utilize clusters of workstations for interactive visualization purposes. It is derived from the WireGL project [2], but contains major extensions of both technology and features. Chromium is the result of a cooperative effort between, among others, IBM T.J. Watson Research Center [3], Tungsten Graphics [4], Lawrence Livermore National Laboratory [5] and Stanford University. The main author is Greg Humphrey, a former member of the computer graphics group at Stanford University. The first public release of Chromium was in September 2002, and several revisions and extensions have been released following the first one.

Features

To make a rather long story short, Chromium supports *all* of the requirements described in previous sections. The Chromium documentation lists the following key features:

- **Sort-first (tiled) rendering -** the frame buffer is subdivided into rectangular tiles, which may be rendered in parallel by the hosts of a rendering cluster.
- **Sort-last (Z-compositing) rendering** - the 3D dataset is broken into N parts, which are rendered in parallel by N processors. The resulting images are merged together according to their Z buffers to form the final image.
- **Hybrid parallel rendering** - sort-first and sort-last rendering may be combined into a hybrid configuration.
- **OpenGL command stream filtering** - OpenGL command streams may be intercepted and modified by a stream processing unit (SPU) to implement non-photorealistic rendering (NPR) effects, etc. Here's an interesting example.
- Many OpenGL programs can be used with Chromium without modification.
- One can write Chromium-specific applications that perform parallel rendering with the aid of special synchronization primitives.
- Chromium runs on Linux, IRIX, AIX, SunOS and Windows-based systems.
- Chromium is an open-source project.

Performance

According to Greg Humphrey, The Chromium installation at Stanford shows good scalability for several different datasets. I have not been able to compare, or find any comparisons, the performance of VizServer to Chromium based clusters. Still, the brief comments and discussions I've found on this matter, indicates that a COTS cluster can easily defend its performance/cost relation compared to a Onyx-server.

Conclusion

The most important conclusion about this brief research regarding Chromium and cluster-based interactive rendering is that there exist several implementations using COTS hardware with a decent cost to performance ratio. These implementations prove the usability of these clusters, and render them in to a real alternative to big-iron solutions such as SGI OpenGL Vizserver and Onyx servers.

A possible visualization cluster at NTNU will be used for educational purposes, and the fact that Chromium is open source should be appealing to the young and curious minds. Chromium is also highly extendable and SGI is planning to become one of the source code contributors.

Different commercial solutions and test beds using Chromium:

IBM: http://opendx.watson.ibm.com/cr/dv/index.htm
Graphstream: http://www.graphstream.com/
Tungsten Graphics: http://www.tungstengraphics.com/
VisBox: http://www.visbox.com/

References

[1] http://sourceforge.net/projects/chromium
[2] http://graphics.stanford.edu/software/wiregl
[3] http://www.watson.ibm.com
[4] http://www.tungstengraphics.com
{5] http://www.llnl.gov