Proceedings of Workshop on Commodity-Based Visualization Clusters,
IEEE Visualization 2002, October 27, Boston, Massachusetts.

Integration of a Commodity Cluster into an Existing 4-Wall Display System

Douglas B. Maxwell, Aaron Bryden, Greg S. Schmidt, Ian Roth, J. Edward Swan II
Virtual Reality Laboratory, Naval Research Laboratory

Abstract

Not all virtual reality applications today require the power or expense of single large visualization "super-computers". Factors such as frame rate and polygon count have a major impact upon the performance of a VR application. Increasingly, low cost commodity consumer electronics and computing technology are becoming powerful enough to present an acceptable level of graphics performance. Already, commodity PCs are driving virtual reality workbenches with stereo and tracking options. The next step is to have them drive multi-screen environments.

We present an experiment motivated by the low cost per performance of PC commodity clusters. The experiment is to replace a visualization super-computing platform driving a 4-wall immersive display system [1] with a PC commodity cluster. We describe the system, implementation and experimental testing in the paper.

1. Introduction

The cost per performance of PC commodity clusters is rapidly becoming a viable alternative to traditional high-end visualization supercomputers. Why should anyone use a commodity cluster when the capability already exists with solutions from SGI [9] and Sun [11]. Consumer electronics and computing technology has evolved at an astounding rate. This rapid evolution has both driven down costs and accelerated obsolescent computing cycles. A general rule to follow for buying graphics capability from SGI is to budget $250,000 per graphics pipe. For those who cannot afford to outfit an Onyx class computer with four graphics pipes, extra raster managers may be added to two pipes to drive a 4-wall display system. In contrast, our experimental cluster costs less than $1000 per node. With the addition of a video matrix switcher, the grand total was less than $15,000. Even with the fast obsolescent cycles the price difference is so great, that an organization could afford to replace or upgrade the graphics clusters many times. Another advantage of PCs is the wide availability of low cost parts, which can be used for repairs and upgrades. Overall, the PCs are cost effective, powerful and flexible.

We present an experiment that integrates a commodity cluster into an existing 4-wall display system—a Surround-Screen Visualization System (SSVR) [2] from Mechdyne Corporation. The objective is to attain active stereo visualization on multiple walls using genlocking, swap-locking and data-locking capabilities.

High-end visualization supercomputers offer multi-wall, active stereo visualization packaged together. Stereo presentation and coordination of scene graph data is automatically taken care of by the computer in hardware or by the invocation of proprietary software libraries. The cluster was designed from the beginning to attempt to replace aging SGI computing equipment used to drive our current 4-wall display system. We are finding ourselves taxing the capabilities of an Onyx 2 system with Infinite Reality 2 graphics by demanding increasing numbers of polygons to be rendered while needing a fixed frame rate for active stereo. When implementing a cluster, these issues must be dealt with in order to produce a coherent scene across many screens. We describe the system, implementation and experimental testing in the paper.

2. System Overview

We present the design strategies, system requirements and details of our system here.

2.1 Cluster Design Strategies

Communication between the cluster nodes is vital. Data such as pixels, geometric primitives, or even scene graph data is passed among the nodes. The way data is handled and the type of data passed greatly impacts the network bandwidth requirements of the cluster. Two basic approaches for setting up a graphics clustering communication software architecture are: Client/Server and Master/Slave [3]. Each method along with its benefits and disadvantages is described in more detail below.

Client/Server: The Client/Server approach consists of a single node cluster that serves data to the graphics rendering clients. The advantage to this arrangement is many applications may embed a server that works with the same rendering client nodes. This environment is very flexible. The disadvantage is a higher consumption of network bandwidth. Most Client/Server clusters rely

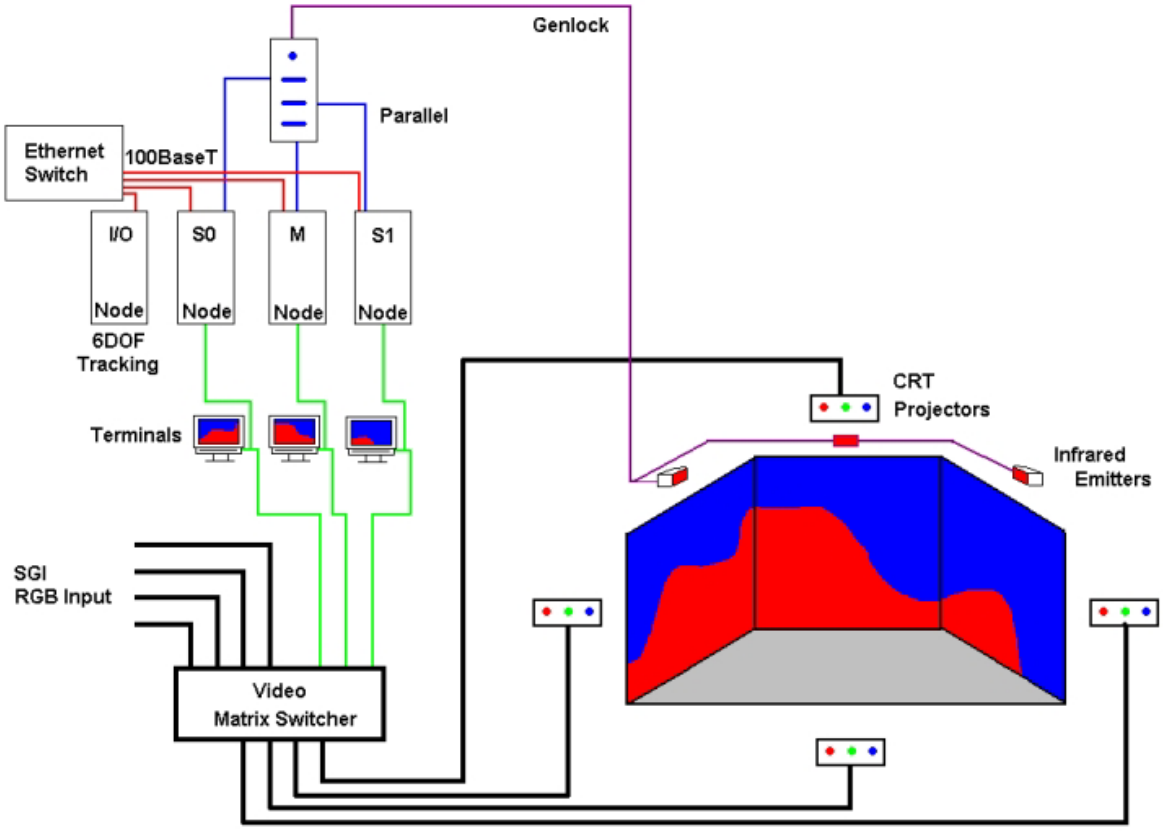

Fig. 1: Commodity Cluster Configuration Diagram

upon relatively expensive Myrinet [4] or gigabit networking hardware.

Master/Slave: The Master/Slave approach consists of multiple nodes, where each node of the graphics cluster locally stores and runs an identical copy of the graphics application. Consequently, only a small amount of information is required to be shared among the nodes, and network bandwidth becomes less of a concern. This information may simply include input device data and timestamps. In this configuration, the master node handles application state changes.

2.2 Graphics Cluster Requirements

All graphics clusters must satisfy three requirements: genlocking, swap locking, and data locking [3]. These are described in more detail below.

Genlocking: Genlocking is the process of synchronizing the video frames from each node in a cluster so that they produce a fluid, coherent image. Genlocking may be achieved through software or hardware.

Swap Locking: Swap locking is the process of synchronizing the frame buffer rendering and swapping. This is necessary since each view of a scene contains different amounts of data and numbers of polygons to render. These may produce different rendering times for each frame for each node

Data Locking: Data locking is the process of synchronizing the views to maintain consistency across the screens. This becomes an issue since each node in the cluster renders its frames from locally stored information.

2.3 Hardware Overview

We used a set of standard PC configurations equipped with MSI G4Ti4600 graphics adapters powered by NVidia Corp.'s GeForce4 Ti gpu and 128Mb of DDR video memory [5]. Although not completely necessary, the PCs were identical, which made software installation easier. The PCs communicated via 100BaseT networking adapters and a 100BaseT switch. We show a diagram of the complete system in Figure 1.

Fig. 2: A fabricated box that handles genlocking and data locking through parallel port connections.

The projectors of the SSVR are connected to an Extron CrossPoint Plus 124 matrix video switcher [6]. The switcher is capable of accepting video input from 12 sources and output to 4 sources.

Since genlocking and data locking are handled in software through the parallel ports, a special box (Fig. 2) was fabricated to handle the signaling appropriately. This box was also built from commercial off-the-shelf (COTS) hardware for less than $20. In addition, this box also outputs a genlocking signal to a set of Crystal Eyes [7] infrared emitters.

3. System Implementation

In this section we give an overview of the software and installation used to create the cluster environment.

3.1 Software

A variety of software is used to deal with the active stereo and scene synchronization needs of the cluster. The commodity cluster was built upon a standard installation of Red Hat Linux 7.2 [14] and the kernel was then patched to include the Real-time Application Interface (RTAI, [8]). RTAI allows for low latency and task completion timing, which can be determined with certainty.

SoftGenLock [9] and RTAI are used in concert to provide a software active stereo solution. The RTAI kernel module detects the vertical refresh of the monitor, and changes a pointer in the video card memory that tells the video card what to draw on the screen. A double sized buffer is provided to the application by specifying a virtual frame buffer in XFree86 [10], which is twice the size of the actual frame buffer being drawn to the screen. For example, if displaying at a resolution of 1024 x 768 in active stereo, the virtual desktop would need to be at a resolution 2048 x 768. The RTAI kernel module splits the frame buffer in half and alternatingly displays the scene in 1024 x 768 pieces. An application draws in stereo by rendering to the right and left sides of the X frame buffer for the right and left eye.

Genlock/data lock is achieved by synchronizing the machines in the cluster over the parallel ports. The RTAI kernel module writes to one pin on the parallel port and reads from another to make sure that the other machines in the cluster have completed a frame. The master tells all the other nodes in the cluster when to draw and the other nodes in the cluster report back when they are ready for a new frame. Data lock is achieved by making sure that the slave machines in the cluster are on the correct eye when the parallel port is in a certain state.

SoftGenLock does not synchronize applications between the nodes of a cluster; it provides data lock and stereo. It is the responsibility of the application to synchronize the viewing frustum and animations in the application. Lastly, since SoftGenLock only uses VGA registers, it potentially can work with any graphics card.

3.2 Installation

When installing a PC cluster, a lot of issues must be dealt with before setup. Here we list the important points and give some recommendations on how to setup the system properly:

- Be sure to have adequate HVAC, power and network access where the cluster will be set up.
- Set up the cluster and display system in separate rooms, since noise levels from fans, drives, etc. may be distracting.
- Prepare to deal with a lot of issues associated with laying down the cables. The time taken at this stage to label and check the cabling for proper size and lengths will save time later. This will alleviate problems with signal degradation and simply losing a cable in the “nest”.
- Create a master power switch to turn off all units at once.

We constructed and installed a 3-wall cluster in less than two weeks. This cluster was configured to use active stereo and has the ability to demonstrate swap locking and data locking.

Fig. 3: Shows a test application driving a CAVE using the pc commodity cluster.

4. Testing and Evaluation

We tested the PC commodity cluster by implementing a simple visualization software system. We utilized an interoperable software architecture, VR Juggler [11], which provides a set of programming abstractions for interfacing with a variety of display, tracking and computing systems, and a variety of interaction devices. The software, which is used for visualizing terrain information, has a variety of display modes including stereo. It can be recompiled to work on different computing architectures, and reconfigured at execution time by including different default device configuration files.

The application integrated smoothly with the cluster system and performed better than expected. The quality of the displays and stereo viewing was comparable to the same software running on three walls using an SGI Onyx 2 with IR2 graphics. However, the cluster was able to visualize the data with better performance. Figure 3 shows the cluster software running in the 4-wall display system on the three side walls.

5. Summary and Conclusions

The use of a graphics PC cluster is now becoming a viable low-cost alternative to the use of single large visualization supercomputers. The PCs and related video hardware are fairly cheap, computationally powerful, and flexible. There is also an abundance of software available for them, such as VR Juggler, which allow for applications to interface with a variety of display systems and interaction devices easily. Our cluster experiment explored the feasibility of phasing out and replacing existing expensive single large computing hardware. The results show we can make this kind of transition in the near future, and we believe our experiences will be motivation for others to follow suit.

References

[1] Cruz-Neira, C., D. Sandin, T. DeFanti, "Surround-screen, projection-based virtual reality: The design and implementation of the CAVE," *Computer Graphics* (Proc. Siggraph 93), 135-142, July 1993.

[2] Mechdyne Corporation, web page www.mechdyne.com/pssvr.shtml, accessed August 2002

[3] H. Kaczmarski, M. K. Zuffo, C. Goudesune, B. Schaeffer, P. Augerat, B. Raffin, P. Bressan, L. Soares, "Commodity Clusters for Immersive Projection Environments", Course Notes 47, Siggraph 2002.

[4] Myricon, Inc., web page www.myri.com, accessed August 2002.

[5] NVidia, web page www.nvidia.com, accessed August 2002.

[6] Extron Electronics, web page www.extron.com/product/product.asp?id=crosspointplus124, accessed August 2002.

[7] StereoGrpahics, web page www.stereographics.com, accessed August 2002.

[8] DIAPM RTAI – Real-time Applications Interface, web page www.aero.polimi.it/~rtai/index.html, accessed August 2002.

[9] SGI, web page www.sgi.com , accessed August 2002.

[10] SoftGenLock, web page netjuggler.sourceforge.net/SoftGenLock.php, accessed August 2002.

[11] SUN Microsystems, web page www.sun.com, accessed August 2002.

[12] Xfree86, web page www.xfree86.org, accessed August 2002.

[13] VR Juggler – Open Source Virtual Reality Tools, web page www.vrjuggler.org, accessed August 2002.

[14] Redhat, web page www.redhat.com, accessed August 2002.