Powering Multiprojection Immersive Environments with Clusters of Commodity Computers

Luciano Pereira Soares, Marcio Calixto Cabral, Paulo Alexandre Bressan*,
Roseli de Deus Lopes, Marcelo Knörich Zuffo
Laboratório de Sistemas Integráveis
Departamento de Engenharia de Sistemas Eletrônicos
Escola Politécnica – Universidade de São Paulo, Brazil
{lsoares, mcabral, pbressan, roseli, mkzuffo}@lsi.usp.br

Abstract

Multiprojection Immersive Environments are used in many fields such as science, engineering and art. Virtual Reality (VR) oriented systems have been powered traditionally by high-end graphics workstations or supercomputers but recently, clusters of commodity computers (PCs, Macintoshes, low cost workstations) have become a practical alternative. The advantages of a commodity cluster include low cost, flexibility, access to technology, and performance scalability. In this paper we describe the family of graphics clusters developed to power Multiprojection Immersive Environments. Design issues and results are presented considering the CAVERNA DigitalTM, the first five-sided cubic Multiprojection Immersive Environment implemented in South America.

Keywords

Multiprojection Immersive Environments, CAVERNA Digital, Commodity Clusters

1 INTRODUCTION

Among many paradigms to implement Virtual Reality systems, the Multiprojection Immersive Environments have been successfully used in many applications related to science, engineering and art ([Cruz93], [DAI97], [Disz97]).

These Multiprojection Immersive Environments are based on multiple high resolution and high bandwidth projectors that can be assembled in several configurations. It includes CAVE™, Panoramas and Power Walls, which provides higher immersion felling, wide number of users and larger resolution.

The numerical and graphics computational demand requirements to power such environments are usually very high considering the complexity of numerical simulations and the graphics realism of the targeted applications.

This work presents a recent paradigm based on clusters of commodity computers to support the computational demand required by applications in Multiprojection Immersive Environments.

Particularly we will present results based on our experience powering a five-sided cubic Multiprojection Immersive Environment with clusters of commodity computers, which we named CAVERNA Digital™.

To differ the proposed VR oriented commodity clusters of computers from numerical oriented commodity clusters (*numerical dusters*), we will call our proposed system the *graphics cluster*.

2 MOTIVATION

Nowadays we can observe the dissemination of Virtual Reality technology in Brazil through several successful applications in science and engineering such as oil exploration, weather forecasting, civil archictecture, learning, energy and aeronautics.

A severe limitation of such systems is the high cost and the rapid obsolescence cycles of the computational infrastructure. These limitations and budget constraints can significantly delay the acceptance to the technology and slower the design of innovative applications related to the specific needs of each society.

The main motivation of this work is to design and implement scalable graphics clusters that could offers the same performance levels but with a factor among 10% to 50% lower costs, than current commercial available systems.

3 THE CAVERNA DIGITAL INFRASTRUCTURE

Figure 3-1 presents the topological configuration of the CAVERNA Digital, the Multiprojection Immersive

* Project supported by Fapesp #99/12693-1 (Brazilian Research Foundation).

environment implemented at the Universidade de São Paulo in Brazil.

The system is based on 5 screens with 3 x 3 meters each. Image generation is based on CRT (*Cathode Rays Tubes*) projectors.

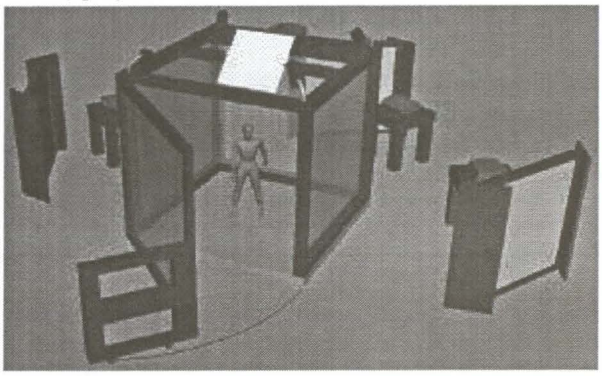

Figure 3-1 – The CAVERNA Digital Layout

A number of 16 infrared emitters (http://www.stereographics.com) that are distributed around the CAVERNA Digital provide active stereoscopy to the user. The screen supporting structure is made on wood and plastic in order to offer less electromagnetic interference to use pointing and positioning six degrees-of-freedom (DOFs) tracking sensors.

The computation and graphics demand of such Multiprojection environment are based on the following requisites:

1. Active stereo computation: two images need to be rendered for each screen, frame rate rendering should be superior that 10Hz, to be well accepted by most users;
2. Multiprojection rendering: 5 stereo-pairs need to be rendered with absolutely no latency among them;
3. Motion Parallax correction: considering the users displacement in the CAVERNA interior during a given simulation, motion parallax correction should be calculated in real time with minimal latency usually less than 30*ms*;
4. Graphics resolution, usually 1024x1024 pixels at 96Hz video refresh rates, the projectors adopted could support up to 2000x2500 pixels in 50Hz;
5. Polygons/scene complexity: depending of each application the total number of polygons range from hundreds to millions polygons/scene.

Since the beginning the design of the CAVERNA Digital considered the use of graphics clusters and an essential issue the full control and monitoring of all resources offered by the facility.

We developed a special console called *Pandemonium* to physically operate, monitor and manage resources in the CAVERNA Digital. Pandemonium integrates 6 high resolution 21'' monitors with active stereo support, a wireless Radio Frequency keyboard, physical device controls (luminescent push buttons, robot sliders, micro joysticks and standard joystick) to operate and calibrate screens and to control light intensity. The system also has video switching buttons to route video signals to projectors and a USB Keyboard and Mouse switch to select cluster nodes. Figure 3.2 shown the user interaction with the Pandemonium console.

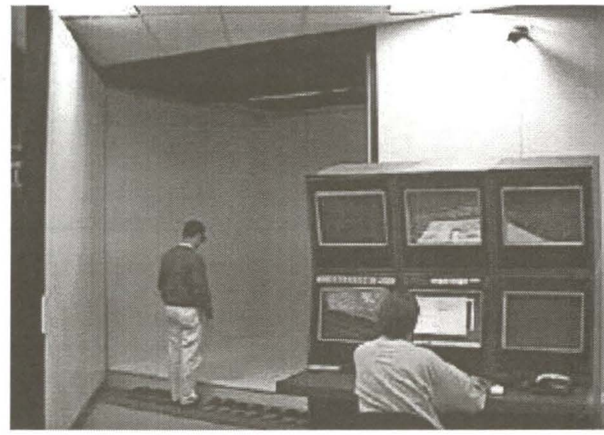

Figure 3-2 – Operating the Graphics Cluster using the Pandemonium Console.

The computational infrastructure to drive applications and software developing in the CAVERNA Digital is a high-speed backbone based on the Gigabit Ethernet technology.

4 THE GRAPHICS CLUSTER ARCHICTECTURE

There are many architectural options when implementing a graphics cluster [Belleman01]. The tradeoffs are related to performance issues, cost, performance scalability (graphics and simulation) and flexibility. Choosing the correct components of a cluster could be a difficult part considering the availability of many commodity parts in the market.

To better classify the different graphics cluster configurations we established taxonomy based in two orthogonal characteristics: node configuration and device mapping. Nodes could be all identical (homogeneous) or with different configurations (heterogeneous) according budget or technical constrains. The way that devices are mapped to nodes could significantly facilitate software developing and enhance its portability from traditional VR computing platforms. This mapping could be symmetric in such way that all nodes are able to access the same physical devices, or the mapping could be asymmetric when devices are connected to a specific node usually enhanced to support it.

Considering this taxonomy cluster organization options range from heterogeneous asymmetric systems to homogeneous symmetric systems.

An example of heterogeneous asymmetric cluster is presented in Figure 4.1. This cluster is based on a network of different configuration computer nodes (numerical or I/O oriented) connected to a graphics node based on a commodity computer, with several graphics cards or a graphics workstation with multiple video outputs. Usually each video output is attached to a particular projector and screen.

Heterogeneous asymmetric systems usually do not offer too much flexibility since they are implemented considering minimal resources. However this system could be implement with considerable low budgets.

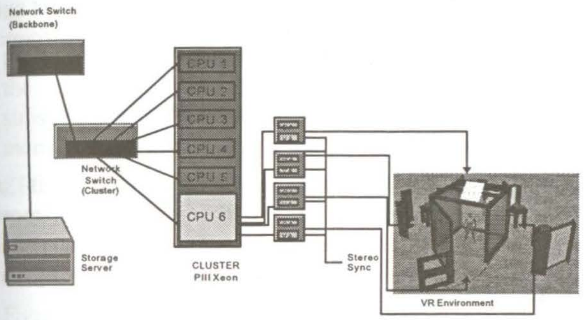

Figure 4-1 – A Heterogeneous Asymmetric Graphics Cluster Organization

An example of homogeneous symmetric cluster is presented in Figure 4.2. This cluster is based on computers with identical configurations connected to a network, the mapping of graphics cards to projectors are highly reconfigurable, offering maximum flexibility. Homogeneous symmetric systems offer better flexibility and performance scalability, since the range of applications requirements could be better supported.

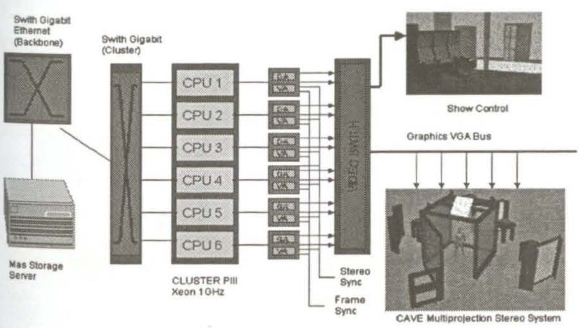

Figure 4-2 – The Homogeneous Symmetric Graphics Cluster Organization

4.1 Video Synchronization

There are two main issues regarding stereoscopic image synchronization on Multiprojection Immersive Environments: Gen-Locking and Frame-Locking.

4.1.1 Analog Video Synchronization Gen-Locking

Gen-Locking is related to active stereo synchronization. In this case the video refresh of all projectors should be done simultaneously and all vertical and horizontal analog *sync* signals need to be absolutely synchronized, in order to have proper use of the active stereo shutter glasses.

Some supercomputers could have many graphic outputs together, and in this case there is no problem to synchronize the video refresh among the graphic outputs. But in clusters as the computers have theirs graphic outputs separated, the video refresh is not necessarily synchronized.

To solve this problem, we need to synchronize the analog video refresh signals among all projectors. Nowadays the only way is the use of signals to control the electron gun scanning to swap all screens at the same time. This analog video synchronization is based on a additional analog Gen-Lock signal that should be distributed to all graphics cards on a single master many slaves configuration.

To implement Gen-Locking some graphic cards offer the physical connection to an external timing source.

4.1.2 Frame Buffer Synchronization Frame-Locking

Frame-Locking is related with the frame synchronization among all frame buffers in each graphics card in the cluster.

The objective of such synchronization is to offer the appearance of a single and continuous frame buffer that in fact is distributed among all graphics cards across the cluster nodes.

Usually the commodity graphic cards use the double buffer technique to couple image rendering rates with video refresh rates. In this situation one frame is displayed while the other one is being rendering. When the frame is ready for display the graphic card swap frames and start to redraw over the old frame. If this swapping is not synchronized users will realize an uncomfortable visual discontinuity of image across the screens.

The Frame-Locking techniques should consider also the different rendering times for different views of the same scene, since each frame buffer will be associated to a different screen and respective view. In Section 7 we describe some implementation issues related to Frame-Locking.

4.2 Node Configuration

The most open decision to cluster designer is the node configuration. The availability of several commodity CPU motherboards with many options among microprocessor performance and number (single, dual or quad), bus and memory bandwidth, AGP and PCI protocols, I/O resources, among other characteristics, can make the choice of such motherboards a quite difficult decision.

A good criterion that always could be use is a detailed understanding of the application computational demands. Dual microprocessor motherboards are usually adopted on graphics clusters considering the performance balancing among microprocessors to perform networking, graphics and simulation processing.

4.3 The Graphics Card

There are some graphics performance characteristics that need to be considered to select a graphics card: fill rate and polygons per second.

The fill rate is related to the pixel throughput, resolution and image refresh rate.

The polygons per second are mostly related with frame rate and polygon resolution of a given scene, and how smooth will be the simulation.

An important feature in a graphics card is its support for 3D acceleration, in OpenGL (Unix, windows) or Direct3D (Windows only) programming platforms. OpenGL is a de facto standard in many professional applications and supported by a broad range of operating systems including Windows, Linux, Irix, Solaris among others.

Direct3D is offered only on Windows platforms and oriented basically for game applications.

Another features should be also considered such as graphics memory (frame-buffer, texture and general purpose memory) and anti-aliasing support. That is particular important for realistic applications.

Finally, considering graphics cluster for Immersive VR, active stereo support is a important feature to allow the connections with the stereo shutter glasses.

4.4 Interconnection network

The choice of the interconnection network is direct related with the graphics cluster performance both for frame synchronization and data consistency.

Network switches can range from hundred dollars to hundred of thousand dollars, and its price is relative to its total internal bandwidth, number of interfaces and protocol technology.

In some particular cases, the total bandwidth required by some applications can be superior than the bandwidth offered by commodity systems, requiring the use of low-latency high bandwidth switches such as Myrinet (www.myricom.com).

We observed that in most applications FastEthernet and GigabitEthernet switches are enough so sustain frame rate and data consistency.

4.5 Controlling Devices with Multiserial Interfaces

A Multiprojection Immersive Environment is a complex environment that becomes even more sophisticated when powered by a graphics cluster. All its components subsystems need to be transparently and easily controlled.

From the hardware point of view these subsystems are usually controlled by serial communication interfaces (RS232 or USB protocols). Among these subsystems we should consider: computer nodes, network switches, video switches, projectors, audio interfaces, control devices, tracker devices, input devices and no-breaks. To support all these serial interfaces (many times easily higher than 16 ports) a multi-serial interface become an essential module of the graphics cluster.

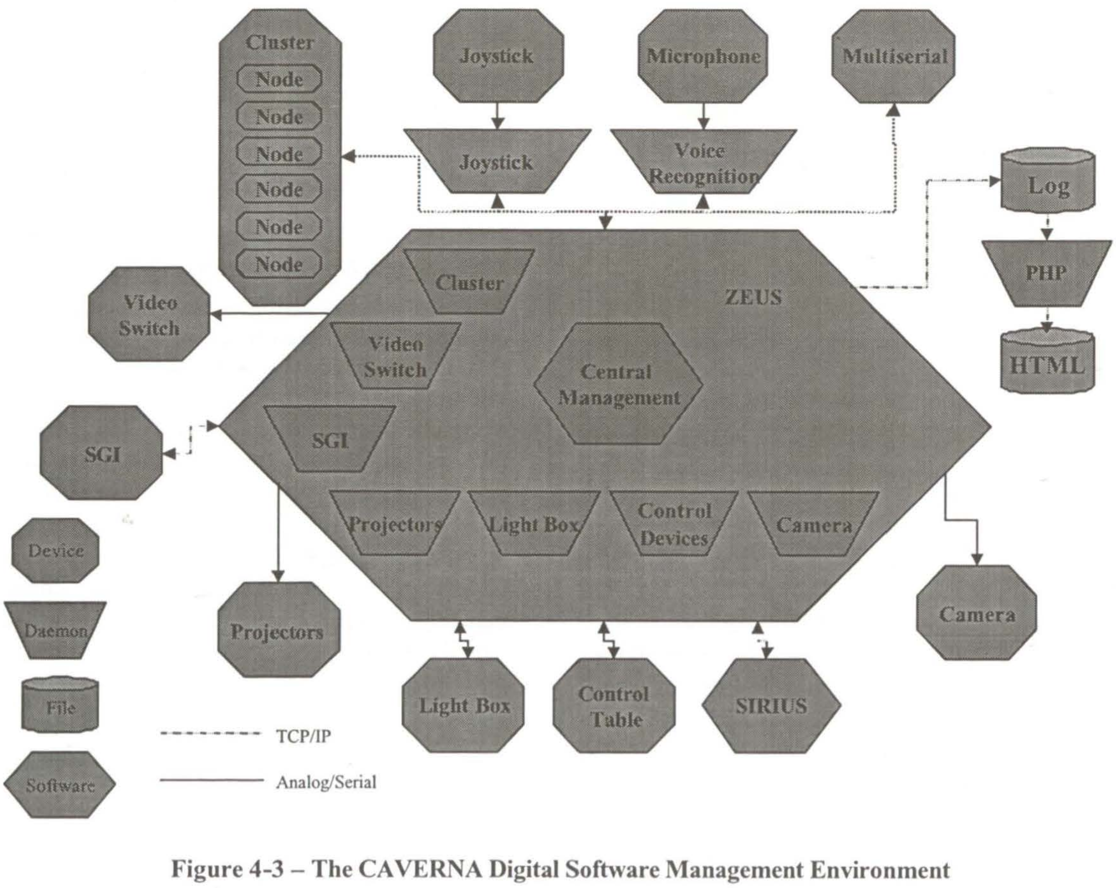

Figure 4-3 – The CAVERNA Digital Software Management Environment

5 SYSTEM MANAGEMENT AND OPERATION

From the software point of view, the whole system needs to have management software to interact with the several daemons for each particular subsystem and establishes a general interface to the user. This management software is called ZEUS, and the user interface is called SIRIUS.

These daemons are responsible for the common protocol interactions between the central server and the particular communication protocol of each subsystem. They are also responsible for status monitoring of each subsystem.

ZEUS is the server platform implemented in C++ running in Linux. SIRIUS is the graphics interface, and considers heterogeneous and remote control, SIRIUS is implemented in JAVA.

Figure 5-1 presents the archictecture of the ZEUS management software and it relationship with all subsystems in the Multiprojection Immersive Environment.

In our system there are several ways to control the devices. The most important is the SIRIUS (Figure 5.2), which communicates with Zeus. The commands are changed in the pre-defined common protocol among all devices that is based in ASCII text and TCP/IP.

Figure 5-1 – SIRIUS Graphics Cluster Administration Interface

Another possibility is the physical control that is the physical counterpart of the SIRIUS interface, incorporating push buttons, micro joysticks and robotic controlled sliders.

This system is platform independent, and it will allow the user to choose which part of the frame-buffer he intend to use in each screen that is very important and possibly will appear in the next version of the management system, as it already exist in other solutions.

6 DEVELOPING APPLICATIONS IN THE GRAPHICS CLUSTER

We consider that the main challenge on the acceptance of graphics clusters in the Virtual Reality community is the availability of powerful developing tools that can facilitate the porting of existing applications from traditional platforms.

6.1 Software Legacy

For the past twenty years, software and algorithms developed for Virtual Reality applications were primarily based on single memory machines concerning texture and graphics processing.

The advent of commodity, based Virtual reality oriented clusters based on distributed graphics processing connected by Gigabit networks, provided a new software developing environment concerning the implementation of real-time parallel graphics rendering.

Every screen should be carefully managed and rendered by the graphics cards in one or more nodes of the cluster. This parallelism is attractive from the performance point of view: While we used to be concerned with limitations for each one of the screens or rendered views not to suppress the performance limits and bus pipe bandwidth of our computer when put together now the limit for each screen is as high as the individual graphics node can handle.

7 IMPLEMENTATION ISSUES

Although programmers does not need to be strongly concerned with graphics performance due the availability of many graphics cards on a cluster, a primary programming concern should be how to synchronize all data and frames being rendered on the screens ([Singhal99], [Schaeffer00]).

The coherency among all screens must be dealt with. When one object travels through screens, the object is actually traveling through cluster nodes. Every node has its own control flags and access to some restricted shared memory area, allocated in a master node. All nodes to maintain consistency access this memory area. As the core communication interface, synchronization libraries make use of optimized network connections to ensure minimum latency and higher bandwidths.

Frame locking is achieved by a combination of graphics hardware specific calls and fast network protocols. These low level specific calls are device dependent, which mean that for every graphics card used, different system-dependent device control system calls must be implemented. Gen-Locking is also achieved through hardware specific calls.

We have developed a synchronization library called DICELib. This library takes care of distribution and synchronization among the nodes as well as view offset corrections for each screen. DICELib will be described in the detail in section 7.4.

7.1 Operating System Issues

The two main operating system architectures running in our cluster are Windows and Linux. We develop applications in both of them respecting each other constrains and restrictions.

Windows based software programming makes use of the easily accessible drivers for professional graphics card available in the market. Drivers are also easily upgradeable which does not interfere in the development process. However, low level native calls to perform hardware specific functions may not be available and costs associated with commercial software are also expected.

Linux based software programming has no cost at all and source code is available for enhancements and modifications. Using our own library synchronization tools together with open source graphics library such as Mesa (a public open-source implementation of OpenGL) we are able to produce high quality Immersive applications from any available common 3D application which are generally faster and much more robust.

Users and programmers must concern regarding drivers availability and kernel configuration, which are sometimes unavailable and hard to configure and require specific technical knowledge intrinsic to the Linux operating system.

7.2 Developing Native OpenGL Applications

Using our synchronization library DICELib we are able to modify native OpenGL applications in order to support multiple views rendering and maintaining data and frame consistency through the nodes.

We developed our own 3D file exporter/importer that is able to convert geometry data files from commercial available modeling software, taking advantage of the OpenGL archictecture we are able to interact and manipulate the geometric data in real time.

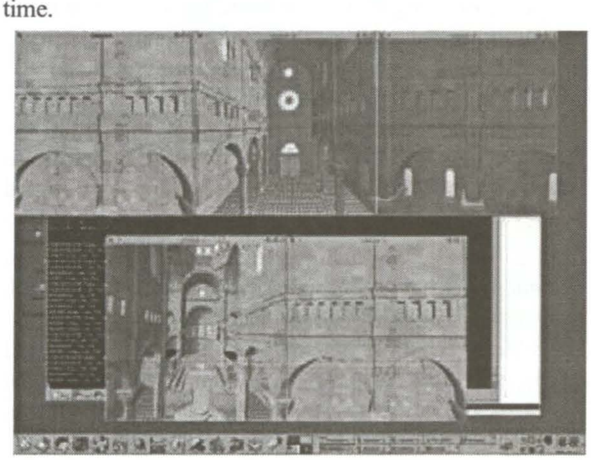

Figure 7-1 – Multiprojection Native OpenGL Application

Figure 7-1 shows five different views of the Sibenik Cathedral with around 90.000 polygons. With little modification in a configuration file of DICELib, we are able to render each one of these screens in a different cluster node and make its display on screens.

7.3 Developing Performer Applications

The DICELib library can also support enhanced graphics programming platforms such as the OpenGL Performer [Rohlf94] for Linux.

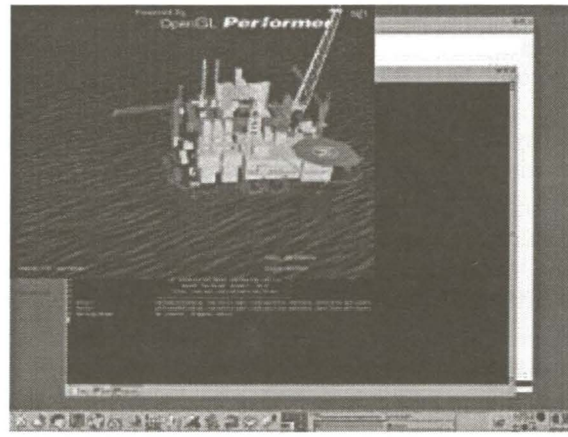

Figure 7-2 – Oil Exploration Application build OpenGL Performer

When developing OpenGL Performer applications for cluster computing using DICELib we don't need to be concerned with specific issues related to performance handling and data manipulation. Another advantage using OpenGL Performer is the use of software available for conventional platforms. Figure 7.2 shows an oil Exploration application build upon OpenGl Performer, the Linux distribution is Debian 2.2r5. The total number of polygons in such oil platform is around 100Kpolygons. These software have a first round in a numeric cluster that calculate the deformation of the pipes and the movement that the waves can do in the oil tanker.

7.4 The DICELib Library

DICELib (DIstributed CAVERNA Engine Library) is a software effort to facilitate the use of graphics clusters in Multiprojection environments [Gnecco01].

The main objectives of DICELib are: efficient synchronization, low processing time and easiness to adapt existing programs to use it.

In order to match these requirements we decided to implement it using TCP/IP, instead of libraries such as MPI or PVM.

DICELib uses client-server architecture internally, but this is transparent to the developer and does not influence on the architecture of the application being developed.

During the application execution, DICELib takes its control and runs the internal server. This server spawns the processes among the cluster nodes. This approach has some advantages: developer can see the cluster as one monolithic single supercomputer, instead of a number of nodes; and communication is reduced, since nodes are connected only to the server, considering that the server can filter redundant updates.

Currently DICELib implements synchronization and synchronous data sharing among the nodes. The functionality is very simple, the server waits until that every nodes sends a message to it, and then the server sends a message to each node.

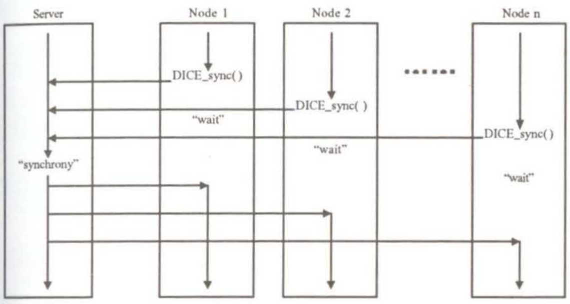

Figure 7-3 Process Synchronization In DICELib

Another facility provided by DICELib is synchronous data sharing. Users can declare, on-the-fly, variables that are updated in other nodes only when the DICE_sync function is called. This is extremely handy for Multiprojection Immersive Environments, because while different nodes will be rendering different views, all have to be done exactly from the position. The developer may create new variable types, ensuring that structures or complex data is easily spread over the cluster. The most common data types (floats, integers, strings, etc) are available by default.

The server acts as a manager, avoiding data coherence issues that are usual in shared memory systems, such as different nodes updating the same variable with different values, creating variables with the same name, etc. To solve these problems, each node is given an ID; lower IDs have higher precedence. DICELib even has a mode in which only the node with lowest ID may create, delete or update variables. This mode saves bandwidth and increases speed of other nodes, which do not need to check if variables were updated upon synchronization, and is very useful for most graphical applications. These applications could use one of the nodes to process input and physics, and the other nodes would only render the data, all that transparently to the developer.

8 CONCLUSIONS

This paper described some experiences with the design, implementation and operation of graphics clusters for Multiprojection Immersive environments.

The availability of a wide range of low cost commodity parts of such clusters, could implicate on difficulties to aggregate parts with the best cost performance ratio. In spite of this, if the final application and the particularities of each VR system is well know, graphics clusters could be built delivering the same performance levels of conventional high-end graphics platforms with a significantly lower budget.

The lack of know-how related to design and implementation of applications in such clusters and the software legacy from conventional platforms still a barrier for the acceptance of graphics clusters in the VR community. However, much of the software developed for commodity computer can become available on graphics cluster with little modifications regarding data synchronization and multiple screen display. We believe that in few years graphics clusters will substitute almost all current available computing platforms with considerable gains to the spread of VR in society.

9 ACKNOWLEDGEMENTS

The CAVERNA Digital and the graphics cluster project was sponsored by FINEP Financiadora de Estudos e Projetos Ministério de Ciência e Tecnologia, Project: “Hipersistemas Integráveis”. The graphics cluster developing was also supported by Intel Semicondutores do Brasil.

Authors would like to thanks Mr. Ruy Rodrigues de Castro from Intel Semicondutores do Brasil for his enthusiast support and good advising on technical issues related to Intel microprocessors family. The Dept. of Naval and Oceanic Engineering at USP developed the oil exploration modeling and simulation under supervision of Prof. Kazuo Nishimoto. Many thanks to Marko Dabrovic (mdabrov@rna.hr) who kindly provided the cathedral model used in the OpenGL application.

Finally we would like to thanks Professor João Antonio Zuffo for all rich discussions related to the whole system design and implementation.

10 REFERENCES

[Belleman01] R. G. Belleman, B. Stolk, R. de Vries, Immersive Virtual reality on commodity hardware, ASCI 2001 Conference (ASCI2001), Heijen, the Netherlands, May/June 2001.

[Cruz93] Cruz-Neira, C.,D.J. Sandin, and T.A. DeFanti. Surround-screen projection-based virtual reality: The design and Implementation of the CAVE. In SIGGRAPH 1993. ACM SIGGRAPH, Anaheim, July 1993.

[Dai97] Dai, P., Eckel G., Göbel, M., Hasenbrink, F., Lalioti, V., Lechner, U., Strassner, J., Tramberend, H., Wesche, G., "Virtual Spaces: VR Projection System Technologies and Applications", Tutorial Notes, Eurographics '97, Budapest 1997, 75 pages.

[Disz97] Disz, T. "Introduction - The CAVE: family of Virtual Reality devices.", Argonne National Laboratory, Proceedings of the USENIX Windows NT Workshop, August 11-13, 1997, Seattle, Washington, USA

[Gnecco01] Bruno Barberi Gnecco, Paulo Alexandre Bressan, Roseli de Deus Lopes, Marcelo Knörich Zuffo. DICELib: a Real Time Synchronization Library for Multi-Projection Virtual Reality Distributed Environments.

[Rohlf94] Rohlf, J. e Helman, J. "IRIS Performer: A High Performance Multiprocessing Toolkit for Real-Time 3D Graphics", SIGGRAPH'94, Orlando, Florida, July, 1994 pp.381-394

[Singhal99] Singhal, S. e Zyda, M. "Networked virtual enviroments : design and implementation", Addison Wesley, 1999.

[Schaeffer00] Schaeffer, B. "A Software System for Inexpensive VR via Graphic Cluster", http://www.isl.uiuc.edu/ClusteredVR/paper/dgdpaper.pdf, Sept., 2000.