# Managing Commodity Computer Cluster Oriented to Virtual Reality Applications

Luciano Pereira Soares, Marcio Calixto Cabral, Paulo Alexandre Bressan*,
Hilton Fernandes, Roseli de Deus Lopes, Marcelo Knörich Zuffo
{lsoares, mcabral, pbressan, hgfernan, roseli, mkzuffo}@lsi.usp.br

Laboratório de Sistemas Integráveis
Departamento de Engenharia de Sistemas Eletrônicos
Escola Politécnica – Universidade de São Paulo, Brazil

**Abstract**

*Virtual Reality (VR) applications 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. The main goal of our proposal is offer both a High Performance Computing System (HPC) and High Availability (HA) simulation environment supporting the features demanded by Virtual Reality applications. In this paper we will approach the challenges related to manage and control this distributed environment in a unique and integrated interface, offering to the user better use of the available distributed resources. Design issues and results are presented considering architectural options and final applications.*

**Keywords:** Commodity Clusters, Virtual Reality, High Performance Computing, High Availability

## 1 Introduction

The use of the commodity computer cluster paradigm to approach the problem of building complex High Performance Computing (HPC) infrastructure is being used with success in many application fields. Therefore, these applications have specific computing and communication demands, such as numerical computing, multimedia delivery and database transactions, among others.

In this article we proposed the management and control of commodity computer clusters oriented for virtual reality applications, called by VR clusters. Such high performance clusters are built by aggregating commodity pieces broadly available commercially, offering the same performance levels of integrated HPC systems but with a considerable reduction of cost, decreasing from 5 to 50 times.

Currently we consider the main disadvantage of commodity HPC cluster systems is the lack of High Availability (HA) characteristics when compared with other cluster-integrated systems. The HA problem on VR clusters is also approached in this article.

The article is organized as follows: section 2 describes the motivation of our work based on the framework of virtual reality simulation environments. Section 3 describes related work

* Project supported by Fapesp #99/12693-1 (Brazilian Research 3Foundation).

to our proposal. Section 4 describes the requirements of a VR cluster. Section 5 describes taxonomy of alternatives of implementing VR clusters according final application requirements. Section

## 2 Motivation

Nowadays we can observe the dissemination of Virtual Reality technology in Brazil and similar developing countries through several successful applications in science, art 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 currently available. 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 discuss how to management and control low cost scalable VR clusters of commodity computers and components that could offer the same performance levels than HPC systems. We believe that this approach will broadly disseminate the VR technology through the cost viability of many applications in society, and add a significant impact on the development of innovative application offering unlimited access to developing platforms.

## 3 Requirements to Graphic Cluster

### 3.1 Synchronization Signals

Basically VR clusters should offer three minimum requirements: data-lock, frame-lock and gen-lock.

To synchronize all the VR application instances distributed across nodes in the cluster, the data between the nodes have to be somehow synchronized and keep consistent taking advantage of the network resources available. This data consistency is called **data-lock**. In many situations replication of such data and proper synchronization is enough for data-locking VR applications, however some VR applications demands huge datasets or complex scene graph environments. These situations demands more sophisticated distributed protocols to keep data and simulation consistent in the virtual distributed environment.

**Frame-Lock** is related with the frame synchronization among all buffers in each graphic card in the VR cluster. The objective is to offer the appearance of single and continuous frame buffer that in fact is distributed among all graphic cards across the cluster nodes. From the visual point of view, users will perceive a consistent image across all screens in the multiprojection VR environment.

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.

**Gen-Lock** is related to active stereo visual consistency among video signal generated from video sources distributed across the cluster. In this case the video refresh of all video sources 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 and display of coherent stereo images to the user.

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 an 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 to synchronize the video refresh.

Nowadays data-lock and frame-lock could be supported by synchronization methods upon fast network technologies and protocols. An important issue to be considered is more focus on the latency than bandwidth. Besides very low latency protocols are required in order to match the timing imposed by the VR application. Some network technologies available that match these requirement are MYRINET and Gigabit Ethernet. Another important practical issue to be considered is the protocol stack. TCP/IP despite is broadly adopted and incorporates a heavy protocol stack imposing extra computation costs that can affect latency. Some alternatives to TCP/IP are been adopted, such as the VIA (Virtual Interface Architecture) protocol stack.

### 3.2 SoftGenLock

Using multi-display system for computer graphics require some schema to synchronize the images. But stereo projections systems need a very accurate way to make the image and video refresh swap at same time in each video output. SoftGenLock [Allard02] proposed a solution that does not require a specific hardware. It also was integrated with NetJuggler and is used to immersive applications.

### 3.3 Node Configuration

The most open decision to cluster designer is the node configuration. The availability of several commodity CPU motherboards with many options can become a difficult decision. The choices are among microprocessor performance and processors number (single, dual or quad); bus and memory bandwidth; AGP and PCI protocols; I/O resources and others characteristics. Dual microprocessor motherboards are usually adopted on graphics clusters considering the performance balancing among microprocessors to perform networking, graphics and simulation processing.

### 3.4 Graphic Cards

There are some graphics performance characteristics that need to be considered to select a graphics card, mainly 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 or Direct3D programming platforms. OpenGL is a de facto standard in many professional applications and is supported by a broad range of operating systems including Windows, Linux, Irix, Solaris and others. Direct3D is offered only on Windows platforms and is 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. Besides considering graphics cluster for immersive VR, active stereo support is an important feature to allow the connections with the stereo shutter glasses.

### 3.5 Interconnection Network

The choice of the interconnection network is direct related with 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 such as Myrinet (www.myricom.com). We observed that most applications FastEthernet and GigabitEthernet switches are enough so sustain frame rate and data consistency.

### 3.6 Implementation Issues

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

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.

### 3.7 Operating System Issues

The two main operating system architectures to commodity 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 graphic cards 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.

## 4 Management System

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.

There are many ways to define a cluster management system, which ones we will focus in some issues:

- Control the nodes – it consists of to know the status and resources of each node.
- Jobs scheduling – it consists of to choose which node the job will be delivered or run. This is very interesting for heterogeneous clusters and solves balance issues.
- Process monitoring – it is important to know what is happening with the process in the cluster during the execution and if it is running in right way.

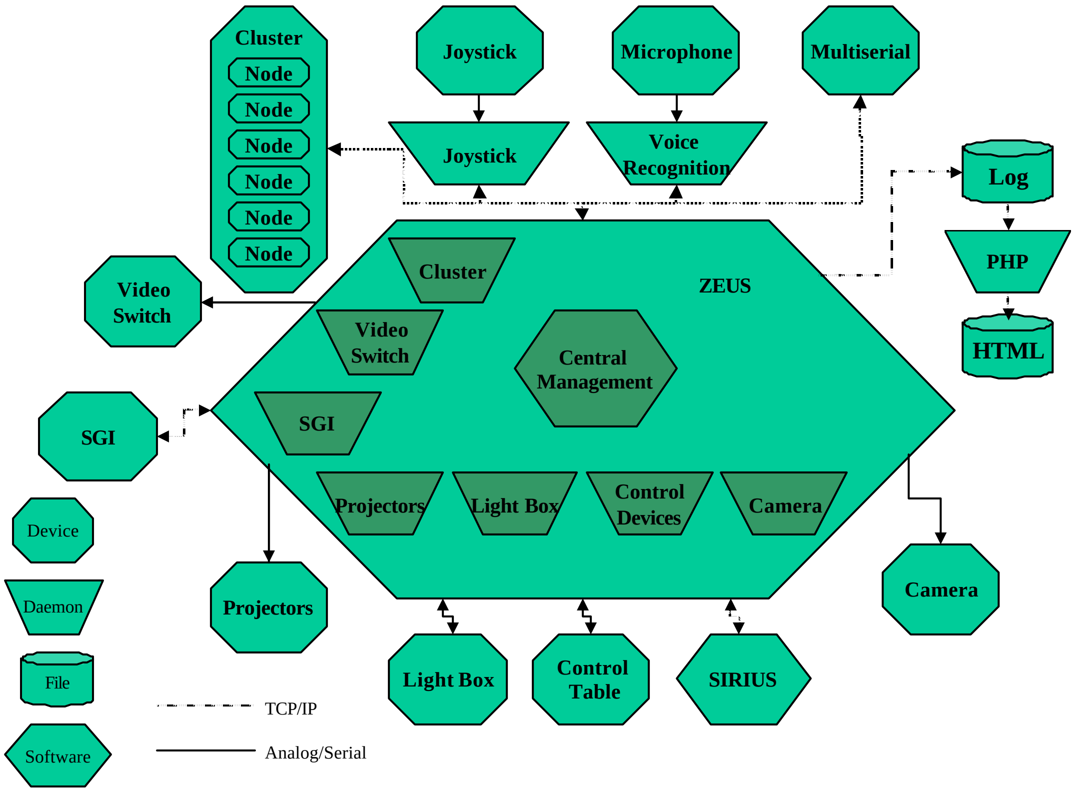


**Figure 1. The CAVERNA Digital Software Management Environment**

### 4.1 ZEUS System

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. 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 and it is doing by controlling devices with multiserial interfaces.

The management software is called ZEUS and the user interface is called SIRIUS. ZEUS is the server platform implemented in C++ running in Linux. Figure 1 presents the architecture of the ZEUS management software and it relationship with all subsystems in the Multiprojection Immersive Environment.

### 4.2 SIRIUS System

In our system there are several ways to control the devices. The most important is the SIRIUS, 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. SIRIUS is the graphics interface implemented in JAVA and considers heterogeneous and remote control. SIRIUS has many interfaces: light, video, demos, tracking, audio, remote and camera.


**Figure 2. Light Control on SIRIUS Interface.**

The light control (Figure 2) is used in immersive applications, which need a very specific kind of illumination. The infrastructure environment should be hide, the user have to pay attention only in the simulation and not in the system behind it. Usually focal lights and directed to an outside user.

To start-up the application, a menu is used with one icon associated with each application that is ease to the user recognized. Also there are folders that allow the applications be separated by topics (Figure 3). In this same menu a dialog box appear and shows many information about the software in question.

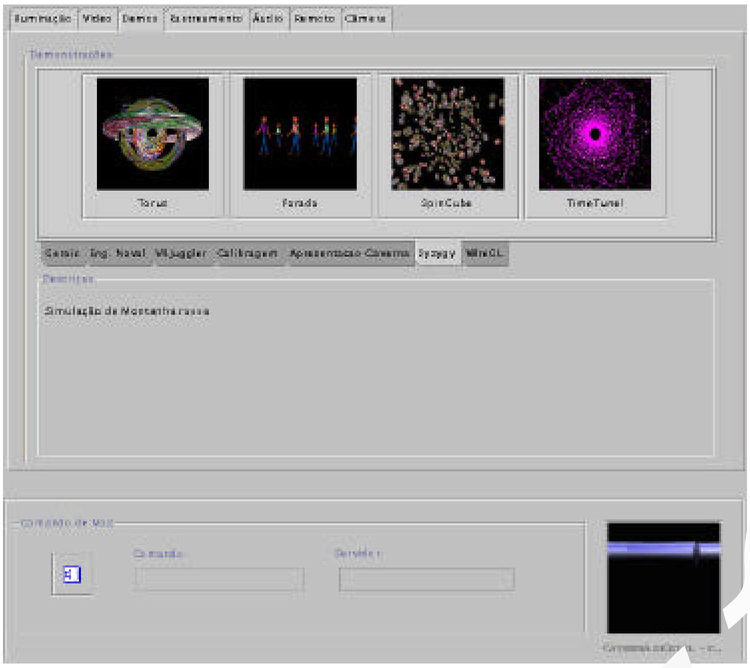


**Figure 3. Application Runner on SIRIUS Interface**

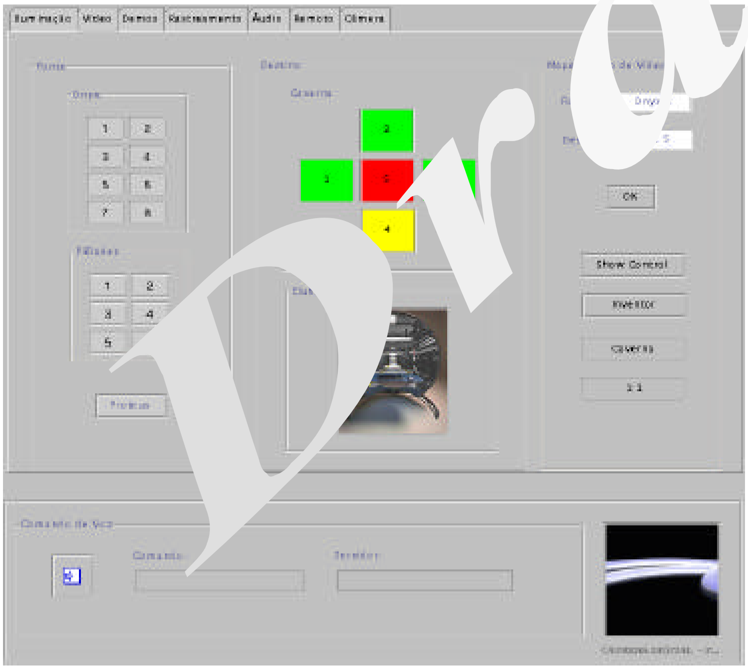


**Figure 4. Video Controller on SIRIUS Interface.**

Another issue regards the projection system control, video controller is responsible to turn on the projection, set-up the right resolution, choose the video source and the projector connected. A menu let the user select which projector he is interested in and show a sub-menu with the command that will be send. Other menu let the user choose which video source it intend to get, and connect to any projector.

Because Microsoft Windows does not fully implement standard protocols, it was created an especial interface that sends jobs to the cluster when it is running Windows. There are icons to easily choose the node and a line to submit the command. Also there is a menu to choose predefined applications.

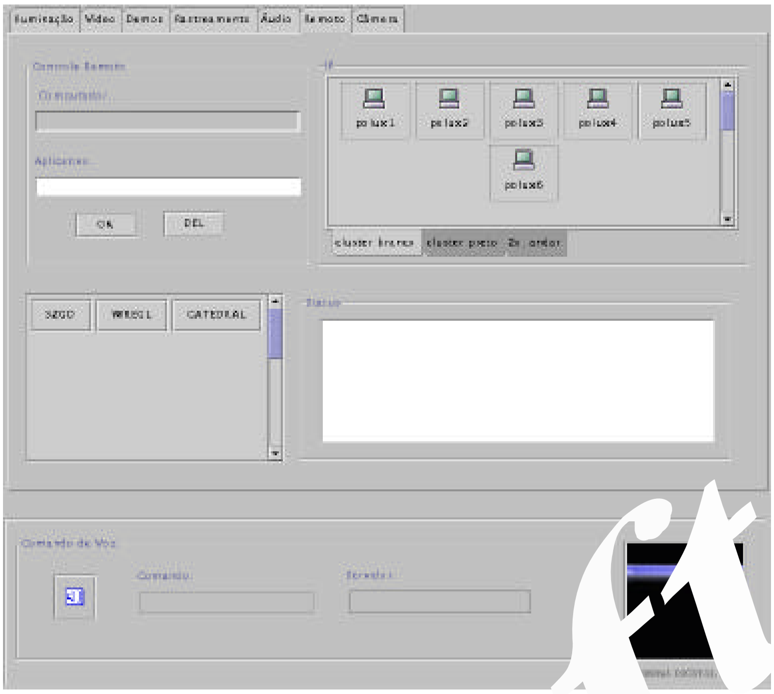


Figure 5. Remote Control on SIRIUS Interface.

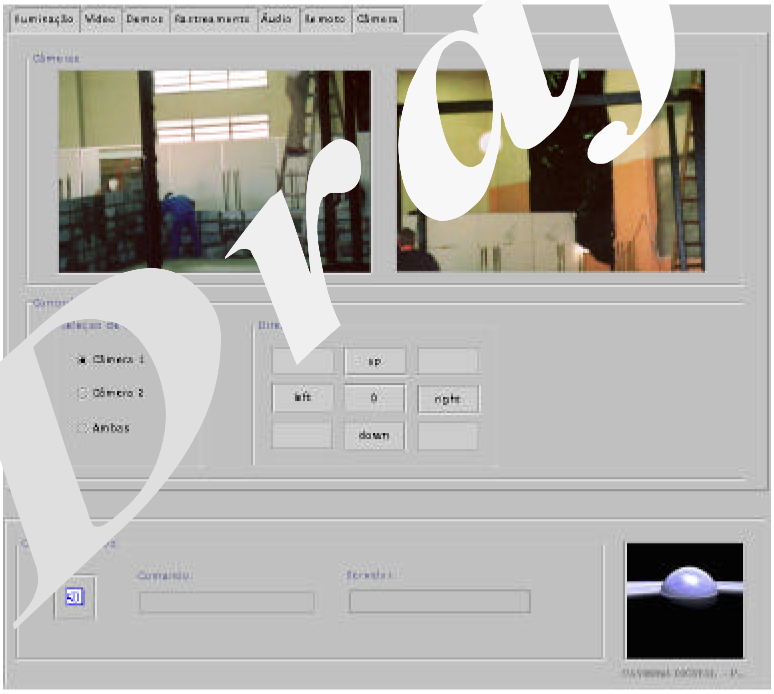


Figure 6. Camera Control on SIRIUS Interface.

Could the user use camera in the place that the virtual reality system is installed, there are many application for it. The camera tracking system in Figure 6 finds where the user is and uses this information to control the environment.

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

## 5 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 management of VR Cluster is an open area and there are many possibilities to increase the power of commodity clusters. The option by independent platform help the system be very accessible, users can access the system without the necessity to go to the cluster. It needs a simple interpreter in his computer and can use the cluster.

## 6 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.

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

## 7 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", Sept., 2000.

[Molnar94] S. Molnar et. al., "A sorting classification of parallel rendering," IEEE Computer Graphics and Applications, 23-32, July 1994.

[Allard02] Allard, J. Gouranton, V. Lecointre, L. Melin, E., Getting Started with Netjuggler and SoftGenLock, VR Juggler tutorial at IEEE VR, Orlando, Florida, March 2002.