

# XML Opportunities in Real Time Immersive Simulation & Visualization Based on Clusters of Commodity Computers

Marcelo Knörich Zuffo  
Marcelo de Paiva Guimarães  
Laboratório de Sistemas Integráveis  
Departamento de Engenharia de Sistemas Eletrônicos  
Escola Politécnica – Universidade de São Paulo, Brazil  
 [{mkzuffo, paiva}@lsi.usp.br](mailto:{mkzuffo, paiva}@lsi.usp.br)

---

## Abstract:

*Real Time Immersive Simulation and Visualization 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, performance scalability and use of legacy systems. This paper describes some opportunities related to the use of the XML (extensible Markup Languages) in such computational environments. 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 real time visualization applications.*

## Keywords:

*Commodity Clusters, eXtensible Markup Languages, Simulation, Visualization*

---

## 1. INTRODUCTION

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

Such High Performance Computing clusters are built by aggregating commodity pieces broadly available commercially, offering the same performance levels of a integrated HPC systems but with a considerable reduction of cost ranging from 5 to 50 times less cost.

Multiprojection Immersive Environments are an effective way to implement Immersive simulation and visualization applications. Such systems are based on multiple high resolution and high bandwidth projectors that can be assembled in several configurations, such as CAVE<sup>TM</sup>, Panoramas and Power Walls, offering a range of possibilities regarding immersion feeling, total number of users and functionality.

The numerical and graphics computational requirements to power such environments are given by the complexity of numerical simulations and the graphic realism of the target applications. These applications range from network videogame engines to virtual walkthroughs on massive databases.

New paradigms to implement these computational infrastructure based on commodity clusters computers have recently showed up, and in this paper discuss some possibilities of adopting the XML (eXtensible Markup Language) framework on such applications.

## 2. THE GRAPHICS CLUSTER ARCHITECTURE

There are many architectural options when implementing a graphics cluster. The tradeoffs are related to performance issues, cost, performance scalability (graphics and simulation) and flexibility. Moreover they should consider the particularities of each Multiprojection Immersive Environment.

Choosing the correct components of a cluster could be a difficult part considering the availability of many commodity parts in the market.

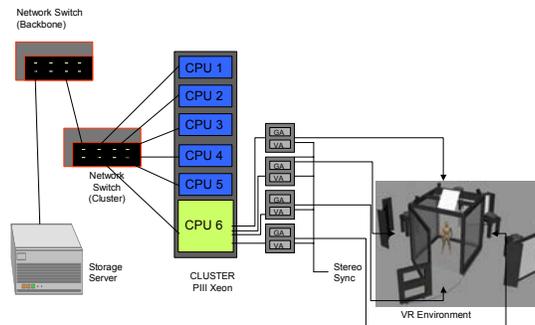
To better classify the different graphics cluster configurations we established a taxonomy based in two orthogonal characteristics: node configuration and device mapping. Nodes could be all identical (homogeneous) or with different configurations (heterogeneous) according budget or technical constrains. 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 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.

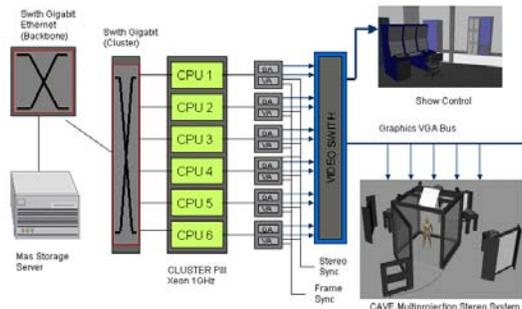
Heterogeneous asymmetric systems usually do not offer too much flexibility since they are implemented considering minimal resources.

However this system could be implement with considerable low budgets.



**Figure 1 – A Heterogeneous Asymmetric Graphics Cluster Organization**

An example of homogeneous symmetric cluster is presented in Figure 2. This cluster is based on a number of computers with identical configurations connected to an isolated 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.



**Figure2 – The Homogeneous Symmetric Graphics Cluster Organization**

The programming model of graphics clusters is based on message passing programming model with explicit use of communication primitives. Developers should care about data distribution data communication and data format across heterogeneous nodes. To improve the development of application on heterogeneous graphics clusters, this works propos the use of XML to distribute the information on all machines. It has been used in some related work with real time graphics visualization and simulation is the SVG [2], for graphics vector description and SMIL [1] for integrating and synchronizing multimedia data.

### 3. XML (EXTENSIBLE MARKUP LANGUAGE) ON GRAPHICS CLUSTERS.

XML provides a structured data representation with easy implementation and deployment. The XML is a subset of SGML, which is optimized to web distribution. It is a standard from the World Wide Web Consortium (W3C), which the structured data is uniform and independent of application and solution providers.

XML allows developers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications on differences nodes and between organizations. It can offer to Graphics Cluster developers the following relevant features:

- *Compression* :XML data could be compressed in order to minimize network contention;
- *It is a standard for data interchange*. Many products are now supporting XML interfaces natively so that they can be integrated more easily. Major vendors such as Microsoft, Sun, and IBM greatly increase the viability of the approach.
- *Platform independent*: Because XML is text-based; XML documents can be exchanged between virtually any operating system along any type of channel that supports text.
- *Easy to write and understand*: The authors of XML took the power of SGML and simplified it so that it could be easily parsed and easily created. Anybody who has ever seen HTML will immediately know how XML works.
- *Many XML tools*: There are an increasing number of tools available, many free, that support most languages. For additional resources, see the inset entitled, "Some Interesting XML Sites."

There are drawbacks to using XML. Since XML is interpreted, it is not as efficient as a binary model. In addition, it is necessary that participants in the exchange of XML data interpret the fields, or elements, or an XML document in the same manner.

This works presents the opportunities of provide a central communication layer for XML information exchange, offering a transparent connectivity between all nodes. According to developers-defined rules, the content and structure of XML data analyzed and routed to the appropriate node. Content-based routing rules use the values of specific XML elements and attributes. Using XML this solution integrate the legacy applications at the on the logic level by encapsulation legacy application functionality and invoking processes through a standard interface.

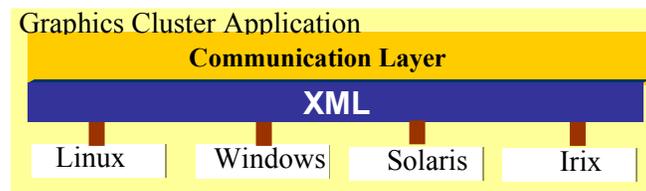


Figure 3 – Graphics Cluster Application Architecture

The Graphics Cluster Application proposed allows systems integration via an XML-based interface, regardless of node type or communications mechanism employed. In order to accomplish these objectives three basic architectural layers are defined: the Communications Layer, XML Layer, and the System Layer.

- **Communication Layer:** The outer-most layer supports the receipt and return of XML formatted messages through any given communications mechanism. The key concept here is that the communications mechanisms are distinct from the actual XML message parsing, message interpretation, and subsequent return message production. This approach makes the system more extensible. The Communications Layer receives XML messages and passes them to the XML Layer, and from a node out to other node.

- **XML Layer:** The second layer is responsible for interpreting XML formatted requests, invoking the appropriate operations upon the system (Linux / Windows / Solares / Irix), and formatting the responses into XML. Transformation of the original XML message is sometimes necessary to ensure that the data is delivered in a format that the receiving application understands.
- **System Layer:** it contains the application than runs on heterogeneous nodes (Linux / Windows / Solares / Irix). The l application does not need to know anything about XML. Furthermore, that application should not need to know about the new infrastructure in which it resides.

#### 4. CONCLUSION

Many applications related to the graphics cluster requires real-time performance, an strategy to achieve high performance is the implementation of efficient data sharing mechanism adopting standard formats for data transmission inter and intra-clusters.

This work presents a multi-layer architecture for transmitting data on homogeneous and heterogeneous graphics clusters. The key points in the architecture are the utilization of XML formatted messages, and a clean definition of layer responsibilities. There are additional benefits to the approach that we have outlined. First, the approach calls for an organization to establish and document (in XML) a transactional business process model for any given system. This model supports scalability, since nodes can be added. Also, following this architecture, despite the variety of node types supported, it will be completely isolated from the infrastructure.

#### 5. BIBLIOGRAPHY

- [1] Je. Ayars, Dick Bulterman, Aaron Cohen, Ken Day, Erik Hodge, Philipp Hoschka, Eric Hyche, Muriel Jourdan, Michelle Kim, Kenichi Kubota, Rob Lanphier, Nabil Layada, Thierry Michel, Debbie Newman, Jacco van Ossenbruggen, Lloyd Rutledge, Bridie Saccocio, Patrick Schmitz, Warner ten Kate --- eds. Synchronized Multimedia Integration Language (SMIL 2.0), W3C Recommendation 07 August 2001. <http://www.w3.org/TR/smil20/>. visited on 08/15/2002
- [2] Jon Ferraiolo -- ed. Scalable Vector Graphics (SVG) 1.0 specification, W3C Recommendation 04 September 2001. <http://www.w3.org/TR/SVG/>. visited on 08/15/2002.
- [3] 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.
- [4] 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
- [5] Schaeffer, B. "A Software System for Inexpensive VR via Graphic Cluster", Sept., 2000.