Clustering for Virtual Environments Using Open Source Tools

Felipe Gómez-Caballero[1], Carlos Delgado-Mata[2], and Jesús Ibáñez[3]

[1] Furui Laboratory, Department of Computer Science,
Tokyo Institute of Technology, Tokyo, Japan
[2] IPIT (Instituto Panamericano de Investigación Tecnológica),
Escuela de Ingenieria, Universidad Panamericana campus Bonaterra,
Aguascalientes, México, C.P. 20290 cdelgado@up.edu.mx
[3] Grupo de Tecnologias Interactivas,
Departmento de Información y Tecnologias de Comunicación,
Universitat Pompeu Fabra,
Barcelona, España

Abstract. This paper presents the design and development of Virtual Reality and Virtual Environment technologies destined to foment computer science knowledge and to increase the use of top range technologies among undergraduate students in a university. The prototype developed is capable of creating virtual environments for diverse analysis and is based on computer clusters (distributed systems) of type *n servers-client* for virtual reality and object-oriented programming in C++ language, using Open Source software. Thus, significantly reducing the investment needed to use virtual reality technology.

1 Introduction

The interest for making a virtual environment project was born when a myriad of feasible applications on diverse fields was envisioned. One such field is the interaction of human beings through computer systems. For an extensive overview of interaction with Virtual Environments, the reader is referred to [8]. Moreover, these systems provide a very useful tool to visualize and to understand objects and phenomena from diverse fields of study (some examples are: climatological phenomena, medical studies, new product prototypes, building visualization).

The virtual environments can go further, for example: by introducing time dependent processes and phenomena, and self dependant elements within the environment. We can also create applications that don't need external intervention to represent certain level of "life". An example of this is the use intelligent agents by applying the concepts of Artificial Intelligence (AI), for more information about intelligent agents in virtual environments, please refer to [2].

However, Virtual Reality and Artificial Intelligence are tasks that require a lot of computer power to run on a virtual environment. The kind of processing power that is needed is provided, in most cases, by expensive computers; therefore implying a high

© *C. Delgado, C. Gutiérrez, R. Velazquez, H. Sossa (Eds.)* *Received 23/01/11*
Advances in Computer Science and Electronic Systems *Accepted 20/02/11*
Research in Computing Science 52, 2011, pp. 14-25 *Final version 07/03/11*

investment. In order to solve this problem we propose the use of computer clusters [1], which are made up of several PC's working together to perform a common task, dividing the processing load among them.

To carry out the research, by means of this technique, we implemented the solution with the smallest feasible cost, and that was achieved by using Open Source [12] tools as our primary development platform. This platform includes the operating system and the necessary software to manage a computer cluster [6][7].

The project described herein is relevant at this moment because Universidad Panamericana campus Bonaterra has research related to the Virtual Reality and fields derived from it [4][5][10][3]. For example, three-dimensional modelling and video game development [11]. However, the resources needed to complete these projects were not available to obtain the results that were pursed.

Because of that, it was necessary to create a virtual environment in which research within CINAVI (Research Centre for Intelligent Virtual Environments, by its abbreviations in Spanish) could be carried out. Furthermore, the aim is also to make this kind of technology available for other fields that can take advantage in using the provided technology. This gives a technological and academic advantage for Universidad Panamericana campus Bonaterra and thus can be applied with the regional industry. Therefore the technology becomes more accessible.

2 Project overview

2.1 System description

The final prototype of the project was developed so that it could be used as a tool that allow to demonstrate the advantages of Virtual Reality technologies by using low cost computers with average processing power (desktop computers). This tool lets us visualize any 3D model created following the VRML standard and later loaded within the application. The model is visualized through the displays of the computers working as graphical servers. The application allows 2 types of navigation within the virtual environment to let the user feel free to explore the ”world”. The framework developed in this project is expected to be used in further research applications and projects and thus increasing its potential.

2.2 Restrictions

To develop this project we had four main restrictions, which were addressed by means of different development methods.

The first and most important, was the cost. The available budget was very reduced, so a study was carried out to find alternatives to accomplish the final prototype using limited resources. The chosen option was to use existing computers in one of the labs in our university.

Because of that, the second restriction arose, which is that the processing power of the available computers in the labs was average, an study was carried out to analyse a solution that could allow the development of an application that should use the computers' processing power available to the maximum.

The third restriction was the time available for the creation of the virtual 3D models for the environments of the University. Thus, the used models were developed by members of CINAVI.

The last restriction was that we wanted to let the students use the project and continue to support and improve it with the aid from other researchers. The licensing model to enable this is free software and Open Source [12], therefore it was chosen.

The project development was conducted as follows: Creation of 3D virtual models for the tests, comparison of clustering technologies, application development and performance tests.

As result, a prototype was created with the following characteristics:

- Any three-dimensional model created or exported on the VRML language can be loaded.
- Separation of processing loads by means of a cluster architecture, in such way that can work in several desktop computers to visualize simple and complex 3D models.
- The number of cluster nodes within the architecture can be changed without having to make modifications to the system.
- Several visualization windows in a single computer can be created.
- The navigation in the virtual environment follows the well known metaphor of mouse and keyboard.
- There are two types of navigation: ”fly” navigation and ”walk” navigation.

We followed the next requirements to build up the system. These are listed by priority:

1. The system should be able to use the maximum processing power of each computer within the cluster
2. The system should be able to display real-time images when responding to user movements.
3. The system should be able to load different three-dimensional models.
4. The system should provide an adequate navigation within the environment.
5. The system should be easy to use.
6. The system should be able to use as many servers as needed.
7. The system should be multi-platform.

As it can be seen, the project was designed and documented with the intention of being improved in future projects.

2.3 Creation of 3D Virtual Models

Different three-dimensional virtual models of Universidad Panamericana campus Bonaterra were developed (Engineering School, main building, etc.). The models were created with AC3D [9], a 3D modelling, multi-platform, software with a simple and intuitive interface. The models were created by students –members of CINAVI.

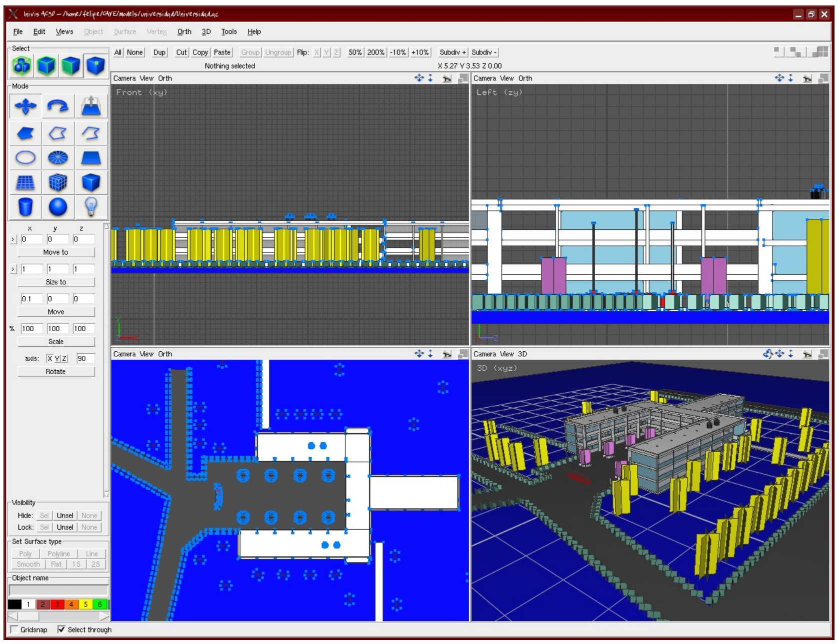

Fig. 1. Main building 3D model.

All the models were created based on the available blue-prints of each one of the university's buildings in order to capture all their details. Fig. 1 and Fig. 2 show some of the models created for the virtual world.

2.4 Comparison of clustering technologies

The reason for implementing a cluster configuration for this project was the demand of processing power required to handle 3D models, render tasks, visualization and manipulation in real time within the virtual environment. This is usually done in high performance computers [15], which are expensive.

Several applications and architectures for cluster implementation [6][7] were analysed during this stage, and it was found that there are a wide variety of cluster configurations as well as other options in software for high performance data processing. The

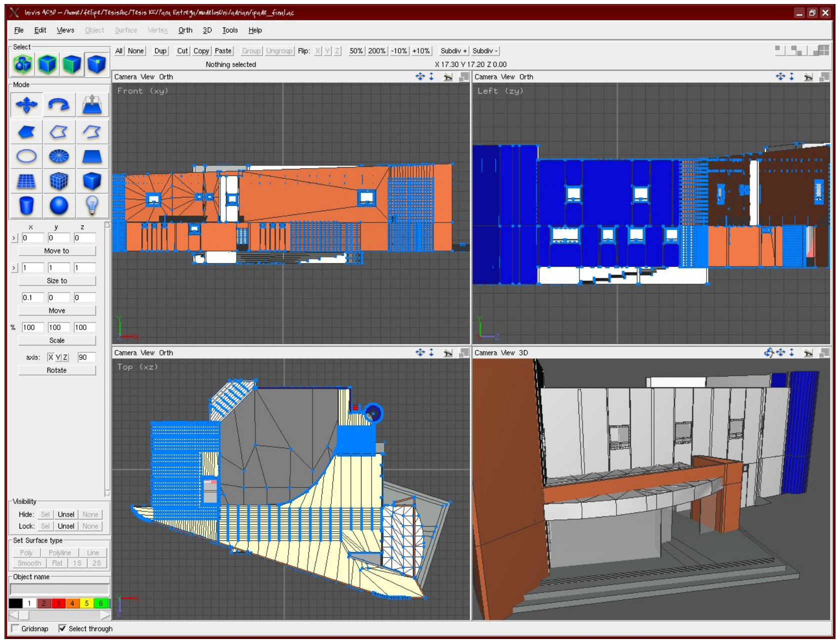

Fig. 2. IPADE building 3D model.

options vary in cost, platform and kind of use for the final application. Among popular applications that were found, are Beowulf, OSCAR and some proprietary solutions from Hewlett Packard and IBM, but none of them fulfilled the requirements to cover the expectations of the final application, because these required an extra configuration for the data flow between the different PC's to create a "n Servers-Client" configuration (Fig. 3). this configuration is similar to the one used for the prototype application on this project.

In order to cover the specifications, we used a cluster implementation from OpenSG scene-graph system [13], which sits on top of OpenGL[14]. OpenSG was chosen as the main tool to run the virtual environment scenes and handle the 3D graphics due to advantages on its competitor OpenGL Performer, like cost and portability. OpenSG was developed following Open Source principles and free software licenses (LGPL). Further, it's a multi-platform software. Another advantage of using OpenSG over the mentioned solution is that the graphical interface and the cluster administration is unified in a single application, reducing the configuration and implementation difficulty.

2.5 Project codification

The project was coded in C++ language using GNU/g++ libraries from GNU/Linux Slackware 10.2 operative system, as well as graphic and clustering libraries from OpenSG.

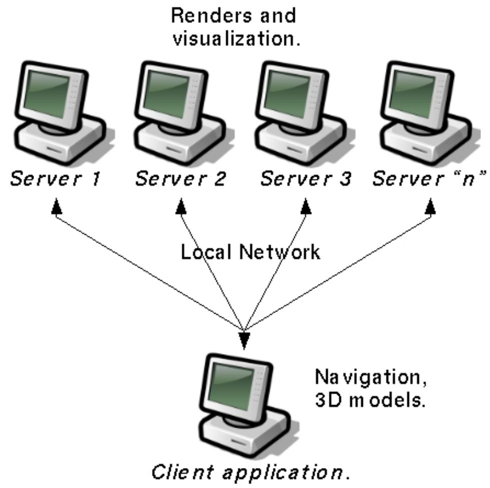

Fig. 3. Cluster architecture.

The project is based on two main modules: Server program and client application. In order to use the n Servers configuration, the client application was coded in such way that it could receive extra parameters through the command line, like the names or IP's of the servers that are being executed in the network and thus to distribute the process load of graphic visualization tasks among them.

One task of the client application is the interaction with the user of the virtual environment, using the GLUT library (GL Utility Library). This library creates the window where the movements of mouse and the events of the keyboard are captured. On the other hand, we created the scene-graph using OpenSG on the client to manage the resources used by the objects within the environment, and therefore we could access any of them.

The process to optimize the resources consists on constant examination to recognize non visible objects and then omit the render process for these objects in order to perform a better and faster visualization.

Another task performed in the client program, is the organization of every object into nodes, the assignation and management of all the matrix transformations related to them, and that are going to be performed within the environment. The process is carried out in the following order:

Firstly, a root node is created and thence all objects are linked to it (land geometry, creature control, navigation control and camera). The land node contains a single geometry file for the land, in such way that there is just one contact point for the navigator, making it easier to manage collision detection of the user. At the same level, OpenSG

creates a node for navigation to manipulate the navigation process.
A hierarchic node tree is created like the one shown in Fig. 4. Also, the client program manages the network connections between itself and the server programs, sending the necessary information of the 3D models through the connection, as well as the events that modify the visualization, this is shown in Fig. 5.

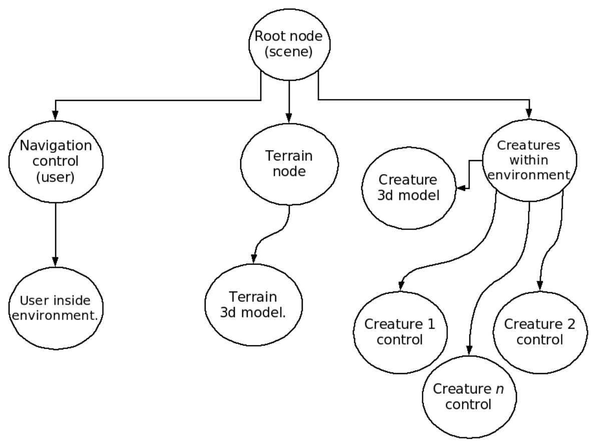

Fig. 4. Environment node tree.

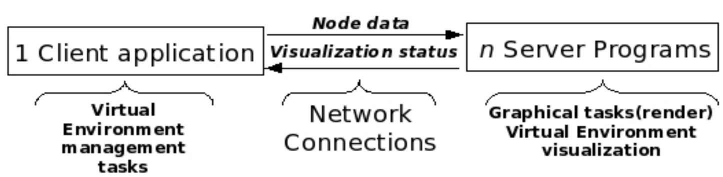

Fig. 5. Network connections client/server.

Secondly, a program was developed for the computers that work as servers. Those computers just need to respond to requests from the client program and receive data through the network connection. Each one of these servers is executed specifying a unique name, so that they can be recognized by the client program when it is executed and it tries to make the connections.

Each server program is executed as an independent process on the network, so each one could carry out the graphical tasks, and it initializes the visualization processes as soon as the execution of the client application is detected. The latter one sends the three-dimensional model data to the servers.

The server programs are always listening into the network looking for possible running client applications, in this way different environments can be loaded from the client with no need to reinitialise the execution of each server program.

The client application and the servers, won't show any visualization windows until the connection between all the cluster nodes are established, after completing this operation, an automatic synchronization between all the involved elements is performed in order to begin the visualization.

However, if there is more than one client in execution, the servers would take care of requests from the first client that handles the connection and synchronization successfully. At the end of the execution of the first client, they would initiate the connections with another one in standby mode. Fig. 6 shows a possible scenario to illustrate the cycle that the cluster would follow in the previously stated case.

There are two types of network connections: Multicast connection (automatic search for servers) and stream sock connection (using server addresses). The multicast connection tries to find the servers checking the whole network for online computers running the server program, this operation can cause a reduction in connection speed, in the worst case scenario a deadlock can be caused when not finding a suitable connection.

Therefore, it was decided to use StreamSock connections, because the cluster was tested and since the application will be used within a university, we tried to affect as little as possible the network traffic. Besides, this kind of connection is safer and more reliable than multicast and therefore the network reliability is not affected.

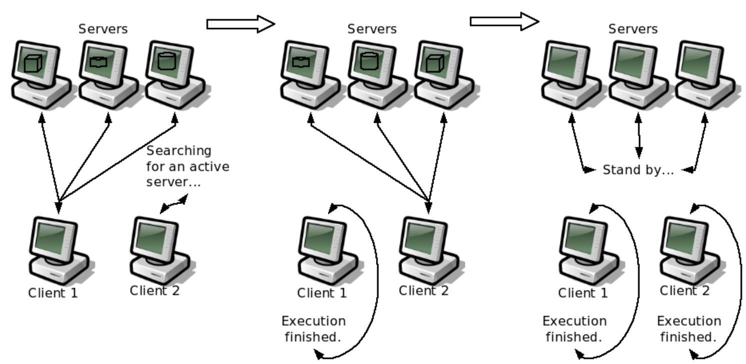

Fig. 6. Life cycle scenario.

3 Performance tests

Several tests for different architecture configurations, different cluster servers and different 3D models of the university were carried out. The architectures studied were using a single computer and from two to four computers processing the task load as a single system.

We obtained better results using a four computer configuration: three of them processing graphical tasks (servers) and one of them working as the client application or navigator for the virtual environment. This configuration was chosen among the others tested because the visualization and distribution of processing loads were optimal to show detailed 3D models on common desktop computers with average processing power.

For the project test and implementation three desktop computers were used to work as servers, with the following technical specifications:
Hardware: Intel Pentium 4 1.6 Ghz processor, 256 MB RAM, network card 10/100, nVidia GEForce AGP 64 MB video card and 80 GB hard disk.
Software: GNU/Linux Slackware 10.2 operative system , GNU gcc/g++ 3.3.4 compiler, OpenSG (cvssnapshot 02-02-06).

Also, one computer was used to work as client with the following specifications:
Hardware: Intel Pentium 4 2.6 Ghz processor, 512 MB RAM, network card 10/100, nVidia GEForce 4 mx440 AGP 128 MB video card and 40 GB hard disk.
Software: GNU/Linux Slackware 10.2 operative system , GNU gcc/g++ 3.3.4 compiler, OpenSG (cvssnapshot 02-02-06) and AC3D 4.0.

All the virtual three-dimensional models of Universidad Panamericana campus Bonaterra were loaded on the computer running the client application.

4 Results

At the end of this research project, a virtual environment that shows Universidad Panamericana campus Bonaterra was developed, using existing resources within the engineering school. Furthermore, this prototype is going to be used as a virtual tour guide application. Moreover, the project is seen as a foundation for new projects within CINAVI.

The clustered VE is shown in Fig. 7. The servers work together to carry out the visualization of the virtual environment which is divided between the number of graphic servers within the cluster architecture. For example, if there are three servers running, the whole view of the environment will divide itself in three screens. Also, there is an extra computer working as a client (or navigator) within the environment. This does not have any visualization, and the movement within the VE depends on the mouse interface connected to the client and it is reflected on the screens of the graphical servers.

The work described herein demonstrates that a virtual environment can be implemented without the need to acquire computers with high processing power. The processing power required can be achieved by dividing the processing loads between the servers and the client. Highly detailed and very exact three-dimensional models can be visualized, this technology can be used for any other application in which high processing power is needed.

A side result of this project was the creation and/or conversion to VRML of 3D models of Universidad Panamericana campus Bonaterra. The models used on this project are available for future projects.

This project opens a possibility for a future research, as similar kind of technology [16] can be used with the approach of a cooperative virtual environment, in which different users can participate and interact within the environment with the purpose of making studies or tasks with a common aim. Besides the processing loads between the different computers connected (as clients or servers) can be distributed in a cooperative VE.

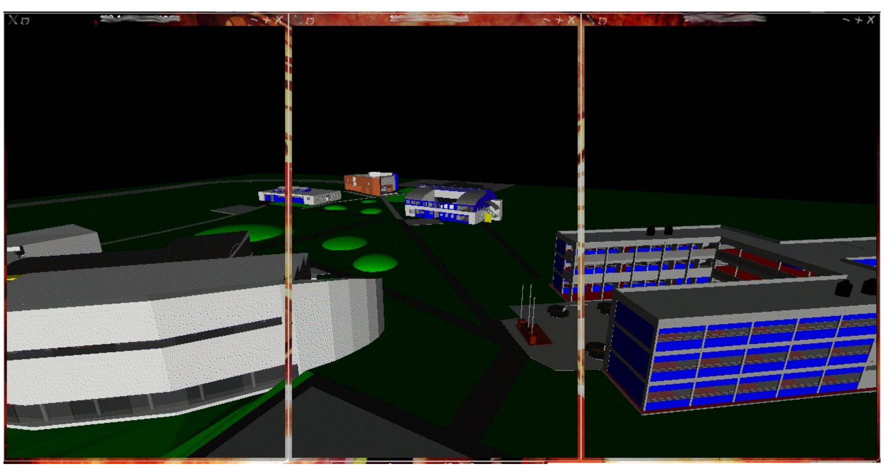

Fig. 7. Windowed cluster view.

5 Conclusions and future work

The design and implementation of this project has brought advance on the computer science academic field within the university. It has brought the possibility for further research projects and the liaison with the regional industry. Also, the interest from university students has arisen and this has caused the growth of the CINAVI group. An important point about this research is the generated cost for the university was minimal for the implementation and tests. The only acquisition were AC3D licenses at a discount price of $29 USD each. Everything else was coded and created with free Open Source software.

On the other hand, the cluster architecture allows the use of average desktop computers from the university labs to perform the tests and execution of the virtual environment, and thus eliminates the necessity for new equipment to access of high processing power like the one required to run virtual environment applications. Furthermore, this project is a technological contribution not only for computer science field and the engineering school, due to the nature of this technology, but to other fields of study that can create "virtual worlds" to extend its use inside the university in particular and the region in general.

References

1. Baker, M.: Cluster computing white paper. University of Portsmouth. United Kingdom (2000)
2. Delgado-Mata, C., Ibanez, J., Bee, S., Ruiz, R., Aylett, R.: On the use of virtual animals with artificial fear in virtual environments. New Generation Computing Journal Vol. 25 (2007)
3. Delgado-Mata, C., Cosio, B.M.L.: Hmm and nn for gesture recognition. In: IEEE CERMA. pp. 56–61. IEEE Computer Society, Cuernavaca, Mexico (September 2010)
4. Delgado-Mata, C., Ibáñez Martínez, J., Gómez-Caballero, F., Guillén-Hernández, O.M.: Transactions on edutainment i. chap. Behavioural reactive agents to define personality traits in the videogame Uberpong, pp. 135–149. Springer-Verlag, Berlin, Heidelberg (2008), http://portal.acm.org/citation.cfm?id=1809375.1809388
5. Delgado-Mata, C., Ruvalcaba-Manzano, R., Quezada-Patino, O., Gomez-Pimentel, D., Ibanez-Martinez, J.: Low cost video game technology to measure and improve motor skills in children. In: IEEE Africon. pp. 1–6. Nairobi, Kenya (September 2009)
6. Harbaugh, L.G.: Building high-performance linux clusters. Appro. USA (2004)
7. Hochstetler, S.: Linux clustering with csm and gpfs. IBM. USA (2004)
8. Ibanez, J., Delgado-Mata, C.: Lessons from research on interaction with virtual environments. Journal of Network and Computer Applications 34(1), 268 – 281 (2011), http://www.sciencedirect.com/science/article/B6WKB-50J9H7B-1/2/1fc48e7fa2b044cdfc8e9a9d73de9fde
9. Inivis-AC3D: http://www.inivis.com/
10. Ibáñez Martínez, J., Delgado-Mata, C.: From competitive to social two-player videogames. In: Proceedings of the 2nd Workshop on Child, Computer and Interaction. pp. 18:1–18:5. WOCCI '09, ACM, New York, NY, USA (2009), http://doi.acm.org/10.1145/1640377.1640395
11. Policarpo, F.: 3D Games Real-time rendering and Software Technology. ACM Press, USA (2001)
12. Raymond, E.S.: The cathedral and the bazaar. O'Reilly Media (2001)
13. Reiners, D.: Open source tools for virtual environments: Opensg and vrjuggler. UPGRADE, Vol. VII, issue no. 2 (2006)
14. Shreiner, D., et al.: OpenGL:Programming Guide. Addison Wesley, 6th edition edn. (2008)

15. Singhal, S., Zyda, M.: Networked Virtual Environments, Design and Implementation. ACM press SIGGRAPH Series (1999)
16. Yu, L., Yu, C., Wang, Y.: Multi-projector seamless displays system based on pc-cluster. Proceedings of the 16th international conference on Artificial Reality and Telexistence pp. 125–130 (2006)