Coupling Parallel Simulation and Multi-Display Visualization on a PC Cluster

Jérémie Allard, Bruno Raffin, Florence Zara

Coupling Parallel Simulation and Multi-display Visualization on a PC Cluster

Jérémie Allard, Bruno Raffin, and Florence Zara

Laboratoire Informatique et Distribution
Projet APACHE ID-IMAG
CNRS - INPG - INRIA - UJF
38330 Montbonnot, France

Abstract. Recent developments make it possible for PC clusters to drive multi-display visualization environments. This paper shows how we assembled parallel codes with distributed 3D graphics rendering on a multi-display environment, to enable interaction with complex simulations.

1 Introduction

Visualization appears as an efficient way to analyze results of complex simulations. Immersive environments, like CAVEs [1], enhance the visualization experience. They provide a high resolution and large surface display created by assembling multiple video projectors. Interactivity and stereoscopic visualization further improve the possibility of these workspaces. These environments are classically powered by dedicated graphics supercomputers like SGI Onyx machines.

Today, the anatomy of supercomputing is quickly and deeply changing. Clusters of commodity components are becoming the leading choice architecture. They are scalable and modular with a high performance-price ratio. Clusters have proved efficient for classical (non interactive) intensive computations. Recently the availability of low cost high performance graphics cards have foster researches to use these architectures to drive immersive environments [2, 3]. The first goal was to harness the power of multiple graphics cards distributed on different PCs. But the scalability and performance of PC clusters allow to go beyond only distributed graphics rendering. While some cluster nodes have graphics cards to power the immersive environment, complex simulations can take advantage of extra nodes to decrease their execution time and reach an update rate suitable for interactivity.

The demo we propose is based on two interactive applications running on a PC cluster driving a multi-display environment: a cloth simulation [4] and a fluid simulation [5]. Both applications involve a parallel simulation coupled with a distributed graphics rendering.

2 Softwares and Environments

The applications were developed on the following open source softwares:

- **CLIC:** Clic [6] is a Linux distribution dedicated to PC clusters. It includes several pre-configured free softwares tools to ease cluster installation, administration and monitoring.
- **Net Juggler:** Net Juggler [3] enables an application to use the power of multiple graphics cards distributed on different PCs. It parallelizes graphics rendering computations for multi-display environments by replicating the application on each node and using MPI to ensures copies are coherent and displayed images are synchronized. Net Juggler is based on VR Juggler [7], a platform for virtual reality applications.
- **Athapascan:** Athapascan [8–10] is a parallel programming language designed to develop portable applications and to enable efficient executions on PC clusters. An Athapascan program consists of a description of parallel tasks communicating through a shared memory. At runtime, Athapascan builds a data flow graph from the data dependencies between tasks. Based on this graph and on a cluster description, it optimizes task scheduling and data distribution.

3 Coupling Parallel Simulation with Parallel Rendering

We present two applications, a fluid and a cloth simulation. Both consist of a parallel simulation coupled with a parallel graphics rendering. They use two different paradigms to parallelize the simulation part: the fluid simulation uses MPI while the cloth simulation uses Athapascan.

3.1 First Case Study: Fluid Simulation

The fluid simulation is based on the implementation of the Navier-Stokes equation solver proposed by Stam [11]. The space is discretized on a grid of cells, where the fluid can flow. Each cell holds a fluid velocity vector and a scalar fluid density characterizing the fluid present in a given cell. At each time step, the solver updates these data. These computations require simple matrix computations, a conjugate gradient and a Poisson solver.

The solver is implemented with PETSC [12], a mathematical library distributing matrix-based operations on the cluster using MPI. As Net Juggler already sets up an MPI environment, it is easy to integrate PETSC in Net Juggler. More details can be found in [5].

This fluid simulation displays the density of two fluids in a two dimensional grid (see Fig. 1). The user can interactively move a pointer that applies a force on the fluid. He can also add or remove obstacles. The simulation and the visualization are executed in a single loop, so they are updated synchronously.

With the solver executed on one node, the obtained frame rate is about 8 frames per second (fps). When the solver is parallelized on four nodes, the frame rate is close to 20 fps.

3.2 Second Case Study: Cloth Simulation

This application uses two different parallelization paradigms. The simulation is parallelized with the parallel programming environment Athapascan [10]. Rendering computations are distributed with Net Juggler [3] on graphics nodes.

The cloth is modeled as a triangular mesh of particles linked up by springs. The object is block partitioned in Athapascan tasks [4]. At each time step, a cloth simulation consists of two steps of Newton's equation integration in order to obtain particle positions. Data mapping, task scheduling and communications are managed by Athapascan at runtime.

By the end of each time step, computed positions are sent in a socket to Net Juggler for graphics rendering (Fig. 2).

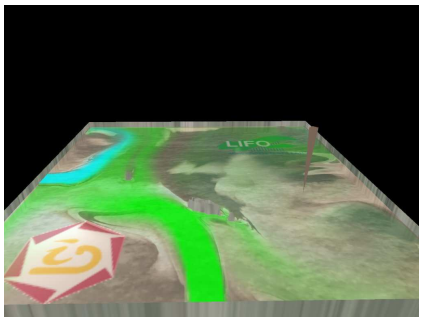

Fig. 1. Interactive simulation of two 2D fluids.

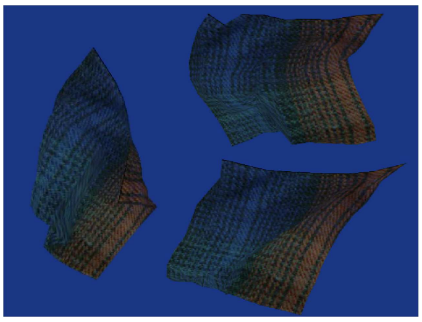

Fig. 2. Piece of cloth of 100 particles with two or one corner(s) fixed.

4 Demo Setup

We will bring to Europar a 6 nodes cluster using a fast Ethernet network and driving 3 video-projectors. We will alternatively show the fluid simulation demo and the cloth simulation demo. The attendees will be able to evaluate by themself the quality and performance of commodity graphics cards, the synchronization level provided by Net Juggler to create a seamless 3 projector display, the interactivity level reached on complex simulations when parallelized. These demo will also give attendees the opportunity to have some insights about the CLIC distribution.

Short videos of the demos are available at http://netjuggler.sf.net/videos.

5 Conclusion

Interactive simulation execution coupled with advanced visualization environments can greatly help to analyze and understand complex data. We showed in

this paper how PC clusters can be used to execute such kind of applications. The fluid and cloth simulations take advantage of two levels of parallelism to enable the interactive execution of complex simulation. The first one distributes graphics rendering on different graphics cards and the second one parallelizes the core of the simulation.

Future works will focus on developing solutions for automatic coupling, targeting executions on large clusters and grid computing infrastructures. Interactive processing and visualization of large data sets will also be considered.

References

1. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.V., Hart, J.C.: The Cave Audio VIsual Experience Automatic Virtual Environement. Communication of the ACM **35** (1992) 64–72
2. Samanta, R., Funkhouser, T., Li, K., Singh, J.P.: Hybrid Sort-First and Sort-Last Parallel Rendering with a Cluster of PCs. In: SIGGRAPH/Eurographics Workshop on Graphics Hardware. (2000)
3. Allard, J., Gouranton, V., Lecointre, L., Melin, E., Raffin, B.: Net Juggler: Running VR Juggler with Multiple Displays on a Commodity Component Cluster. In: IEEE VR, Orlando, USA (2002) 275–276
4. Zara, F., Faure, F., Vincent, J.M.: Physical cloth simulation on a PC cluster. In D. Bartz, X.P., Reinhard, E., eds.: Fourth Eurographics Workshop on Parallel Graphics and Visualization 2002, Blaubeuren, Germany (2002)
5. Allard, J., Gouranton, V., Melin, E., Raffin, B.: Parallelizing pre-rendering computations on a net juggler PC cluster. In: Immersive Projection Technology Symposium, Orlando, USA (2002)
6. MandrakeSoft, Laboratoire ID, B.: (The clic linux cluster distribution) clic.mandrakesoft.com.
7. Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., Cruz-Neira, C.: VR Juggler: A Virtual Platform for Virtual Reality Application Development. In: IEEE VR 2001, Yokohama, Japan (2001)
8. Roch, J.L., *et al.*: Athapascan: Api for asynchronous parallel programming. Technical report, INRIA Rhône-Alpes, projet APACHE (2003)
9. Galilée, F., Roch, J.L., Cavalheiro, G., Doreille, M.: Athapascan-1: On-line building data flow graph in a parallel language. In IEEE, ed.: Pact'98, Paris, France (1998)
10. Gautier, T., Revire, R., Zara, F.: Cluster computing with athapascan. Submitted to Euro-Par 2003 (2003)
11. Stam, J.: Stable Fluids. In: SIGGRAPH 99 Conference Proceedings. (1999) 121–128
12. Balay, S., Gropp, W.D., McInnes, L.C., Smith, B.F.: PETSc 2.0 Users Manual. Technical Report ANL-95/11 - Revision 2.0.29, Argonne National Laboratory (2000)