PixelVizion: An NPU-based Compositor for the Visualization of Large Datasets

Carolyn Connor Davenport, Andrew DuBois, David DuBois, Stephen Poole, HPC-5; Laura Monroe, David Pugmire, HPC-4

PixelVizion received a 2006 R&D 100 award.

Advances in high-performance computing have enabled scientists from a variety of disciplines to perform calculations that generate more data than is stored in the entire Library of Congress print collection (17-20 terabytes). A large 3-D problem to visualize a dynamic process may have more than 2 billion cells, with tens of variables per cell, and hundreds of timesteps producing terabytes to petabytes of simulation data to visualize and analyze. Just the sheer volume of data presents a challenge to even the highest-end graphics hardware available today.

Although many advances in speed, memory, and performance have been made in graphics hardware in recent years, the improvement in such hardware is driven by market forces, which are not always aligned with high-performance scientific computing and visualization needs. Much research has been carried out to leverage these advances so that they may be used in visualizing large datasets.

Visualization clusters composed of commodity nodes are often used to perform distributed rendering of large datasets. This type of large-scale data visualization usually involves workload distribution across a cluster of computational nodes, since the datasets are too large to fit on a single graphics processor. This rendering distribution necessitates a final compositing step to blend the rendered images into one image.

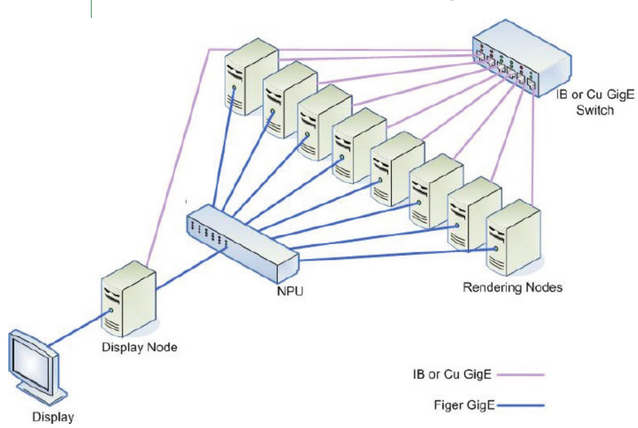

Fig. 1.
An NPU-enhanced visualization cluster.

This compositing step is a well-known bottleneck that places an upper bound on performance. Thus, although distributed commercial off-the-shelf (COTS) systems do offer increased graphics horsepower and a better cost/benefit ratio, the compositing step must be improved in order to truly realize the benefits of this type of rendering system.

We have recently developed a novel use of Network Processing Units (NPUs) to perform the compositing step on a COTS cluster [1]. The NPU uses a heterogeneous multicore compute architecture that efficiently maps onto the streaming nature of image composition [2]. We streamed images generated on current graphics cards into the NPU via a simple application-programming interface (API) and allowed the NPU to composite the images.

The NPU-enhanced visualization system makes use of a completely COTS NPU rather than the interconnect to perform high-speed hardware-assisted image composition. The NPU augments a visualization system composed of a traditional COTS rendering cluster equipped with COTS graphics cards and an InfiniBand 4X network connection, with GigE connecting the NPU compositing system to the main cluster (Fig. 1).

We attained compositing speeds beyond that of the comparison software-only composition even when the software-only method had the advantage of a faster high-performance network. Unlike traditional software-based methods that do not scale with cluster size [3], the NPU compositor is able to maintain a nearly constant frame rate for arbitrarily large clusters. The theoretical frame rate for such an NPU system built using a GigE network is 28.68 frames per second, and this frame rate is attained by our system.

At the 2005 International Conference on High-Performance Computing, Networking, Storage, and Analysis, we demonstrated 1024x1024 image composition speeds nearly 4X faster than the standard fast binary-swap software-based composition (Fig. 2). This was done using just the compositor with prerendered images.

We also incorporated NPU-based composition into VisIt, the Lawrence Livermore National Laboratory (LLNL) visualization tool [4], to demonstrate the integration of this technology to an existing, full-featured visualization system (Fig. 3), and applied it to real Los Alamos National Laboratory simulation datasets (Fig. 4) [5]. Integrated NPU-accelerated composition achieved a 2X increase over the performance of VisIt using the software binary-swap composition that was implemented.

The difference in speedup between the movieplayer case and the VisIt full-package case above (4X vs 2X) is due to (1) the nonoptimized application pipeline, and (2) the inherent system overhead penalty associated with the use of any full-featured visualization package.

We have since integrated stereo into the VisIt/NPU system. We have also achieved a preliminary integration of another scientific visualization software package, ParaView [6] into the NPU compositor system.

[1] D. Pugmire, et al., "NPU-Based Image Compositing in a Distributed Visualization System," to appear in IEEE Transactions on Visualization and Computer Graphics 2007.
[2] B. Bouzas, et al., "MultiCore Framework: An API for Programming Heterogeneous Multicore Processors," Information Sciences Institute, University of Southern California (March 2006).
[3] A. D. Brydon, et al., "Simulation of the Densification of Real Open-celled Foam Microstructure," *J. Mech. Phys. Solid* **53**, 12, 2638 (Dec. 2005).
[4] M. Ogata, et al., "The Design and Evaluation of Pipelined Image Compositing Device for Massively Parallel Volume Rendering," Proceedings of the 2003 Eurographics/IEEE TVCG Workshop on Volume Graphics, 61 (2003).
[5] VisIt home page, http://www.llnl.gov/visit/
[6] ParaView home page, http://www.ParaView.org/

For more information contact Carolyn Connor Davenport at cmcd@lanl.gov.

Funding Acknowledgements
This project was supported through the LANL ASC Simulation and Computer Science Production Visualization project and the LANL ASC Weapons-Supported Research project.

Image	Nodes	NPU fps	BSC fps	Speedup
512^2	8	89	26	3.4X
1024^2	2	28		
1024^2	3	28		
1024^2	4	28		
1024^2	5	28		
1024^2	6	28		
1024^2	7	25		
1024^2	8	22	6	3.7X
2048^2	8	5.5	1.5	3.7X

Fig. 2.
Comparison of binary swap compositing (BSC) to single NPU compositing on a variety of render node configurations.

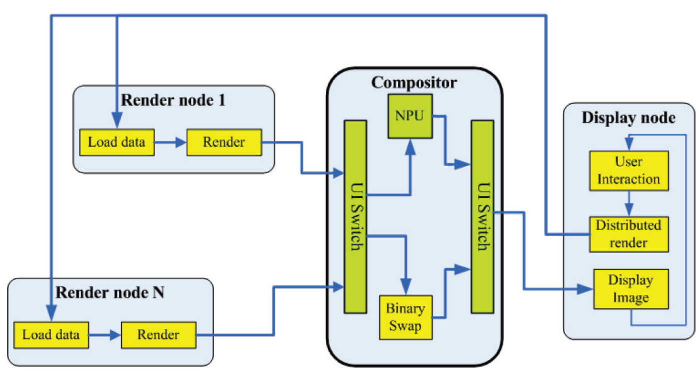

Fig. 3.
The VisIt distributed rendering pipeline. The green blocks indicate modules added to the VisIt pipeline to incorporate NPU-enhanced rendering: specifically, UI switches select between the binary swap or NPU compositor.

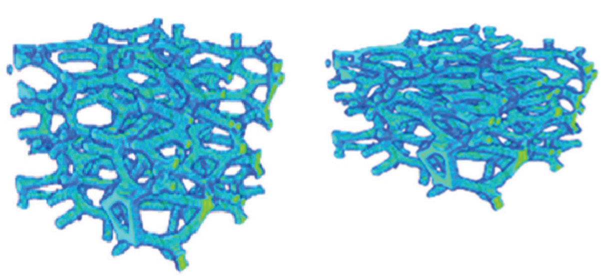

Fig. 4.
Low-density polymeric foam calculation. On the left is the initial configuration; on the right is a deformed intermediate state.