

Development of a Scalable Concurrent Visualization Approach for High Temporal- and Spatial-Resolution Models

 Bryan Green & Chris Henze (NASA Ames Research Center), Bo-wen Shen (University of Maryland and NASA Goddard Space Flight Center)

INTRODUCTION

As recent advances in supercomputers enable the deployment of high-resolution global and regional models, managing the data produced by high temporal- and spatial-resolution models becomes a challenge. For example, although important insights into model behavior and physical processes can be gained from examining results at a high temporal resolution, the overhead of mass storage I/O typically necessitates a coarser-grained output rate. Concurrent visualization (CV) provides a solution to this problem by directly visualizing data extracted from the simulation in progress, bypassing mass storage. CV can take advantage of a high-speed network fabric to transfer the simulation's domain from the compute nodes to the visualization nodes, without significantly impacting simulation performance. Since it is not desirable to repeat the same simulation over and over again to produce additional visualizations, the CV system is designed to generate numerous visualizations from a single run. Results are automatically encoded as movies and delivered to the end-user. The CV architecture is currently being adapted to scale to support simulations running on thousands to tens of thousands of processors.



The Pleiades supercomputer cluster has 9,216 nodes (81,920 cores). Jobs running on Pleiades have direct access to the hyperwall-2 visualization cluster.

CONCURRENT VISUALIZATION ARCHITECTURE

The first-generation CV architecture exploited a large-scale, shared-memory architecture (NASA's Columbia supercomputer) to quickly coalesce the simulation domain to a single buffer for broadcast to NASA's hyperwall visualization cluster (Figure 1). At the time, only 2D visualizations (slices of the full domain) were supported due to network bandwidth limitations. Recently, the approach was adapted for a large, distributed-memory supercomputer cluster (NASA's Pleiades supercomputer), and the network infrastructure has improved to the point 3D visualizations become feasible (Figure 2). Initially, the simulation nodes could not "see" the visualization cluster's nodes, so it was still necessary to coalesce and rebroadcast the entire domain through a single gateway node—a configuration that would limit the system's scalability. Now, this limitation has also been overcome, making possible a more scalable "M-on-N" approach to CV (Figure 3). In the third-generation CV architecture with the M-on-N model, M simulation nodes send data directly to N visualization nodes. The domain may be recomposed in part, or not at all, and parallel visualization techniques can operate directly on the existing decomposition. Once the data is transferred, the simulation continues while data processing and visualization takes place in parallel.

PERFORMANCE

Table 1 provides details on several successful concurrent visualization runs. Of these, "stagger-code," a solar convection code, used the M-on-N model to process 47 Terabytes (TB) of data and generate twenty-four 3D volume visualizations (Figure 6).

Table 1

Example of CV Runs in Chronological Order					
Code	Run Time	Processors	System	Data Processed	Visualizations
GEOS4	35 minutes	486	Columbia/HW-1	63.0 TB	24 2D
MITgcm	4.5 days	1920	Columbia/HW-1	63.0 TB	48 2D
WRF	6.5 days	224	Columbia/HW-2	21.0 TB	74 2D/3D
stagger-code	4.5 days	2016	Pleiades/HW-2	47.0 TB	24 3D Volume

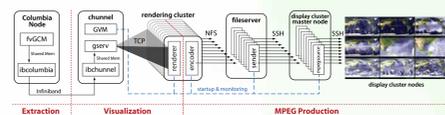


Figure 1: The original Columbia/hyperwall-1 concurrent visualization pipeline used shared-memory (NUMA) to copy data out of the simulation. Rounded rectangles indicate systems, and rectangles indicate processes. Images on the left are results from a GEOS4 CV run.

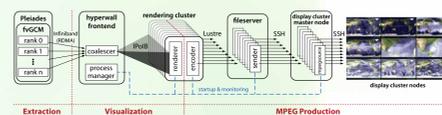


Figure 2: The second-generation concurrent visualization pipeline, adapted for Pleiades and hyperwall-2. Data from simulation processes are coalesced into a buffer on a front-end system, processed, and broadcast to visualization clients.

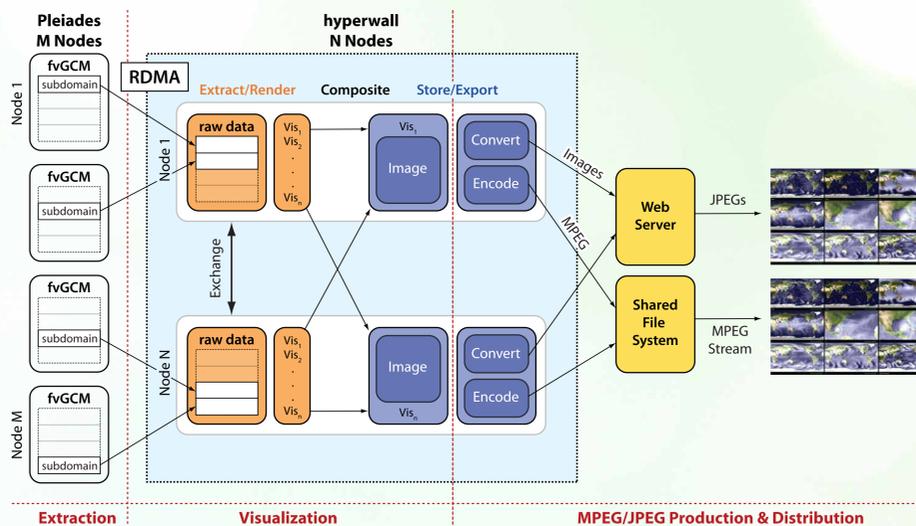


Figure 3: The M-on-N configuration of the concurrent visualization pipeline. The boxes on the left represent the processes of an (fvGCM) MPI job, each responsible for a portion of the domain. At startup, each process creates an InfiniBand (IB) connection to a process that is part of an MPI job running on the hyperwall, using an M-on-N mapping where $M \geq N$. At the end of each timestep, raw data is transferred directly via InfiniBand RDMA to the hyperwall nodes. The hyperwall job then performs feature extraction and "sort-last" rendering; each process renders an image from the portion of the data it receives, then the partial images are sorted and composited into a complete image. The completed images are converted to JPEG and delivered to a web server, as well as being passed to an encoder for movie generation.

Multiple visualizations are produced from a single run. When timestep data arrives, the visualizations are processed in round-robin fashion ($Vis_1, Vis_2, \dots, Vis_N$), producing one image per visualization request. The destination node for the final composite image is also assigned in a round-robin fashion, so that the encoders are spread out across the cluster.

Depending on the visualization produced, some data exchange may occur between the hyperwall nodes. For example, it may be convenient to reconstruct the levels of a sub-region, creating a 2D vertical decomposition, for performing vector visualization techniques at pressure levels. However, visualizations such as scalar volume rendering, cutting planes, and isosurfaces are easily implemented within the "sort-last" renderer, which only ghost-cell exchange is needed for the sub-domain boundaries.

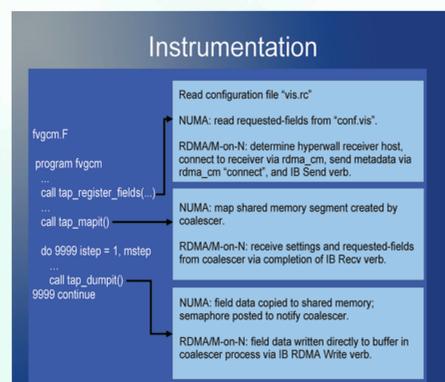


Figure 4: Overview of the fvGCM instrumentation required for CV, with comparison of Columbia shared-memory (NUMA) vs. Pleiades RDMA/M-on-N methods.

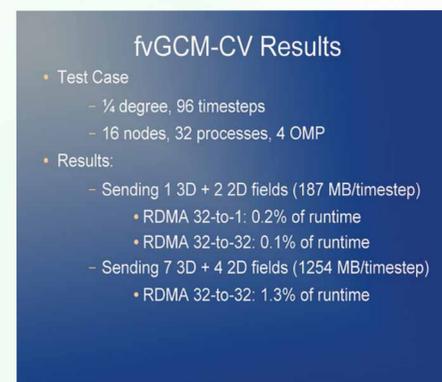


Figure 5: Sample measurements from one fvGCM test case. Results show communication overhead imposed on fvGCM by RDMA for 32-to-1 (2nd Gen CV) and M-on-N where $M=32$ and $N=32$ (M indicates number of MPI processes, not nodes).



Figure 6: The hyperwall-2, with concurrent visualization results from a Weather Research and Forecasting Model (WRF) simulation of 2009 Typhoon Morakot.

CV applied to the finite-volume GCM (fvGCM), a numerical representation of the complex equations used to predict the evolution of physical quantities associated with Earth's atmosphere, was achieved with very few modifications to the fvGCM source code. Three function calls added to the main source file is all that was necessary (Figure 4). Using a $1/4$ -degree test case, with a 32-process, 4-OMP fvGCM job running on 16 Pleiades nodes, the overhead for transferring one 3D and two 2D fields (187 MB/timestep) is about 0.2% of runtime when coalescing to a single visualization cluster node. For M-on-N to 32 nodes, the overhead is about 0.1%. When transferring seven 3D fields and four 2D fields (1,254 MB/timestep), the overhead for M-on-N increased to about 1.3% (Figure 5). In separate tests, values for M and N increase and the achievable bandwidth between the two clusters increases with aggregate bandwidths up to about 15 GB/second achieved to date.

CONCLUSION

The first- and second-generation concurrent visualization systems have been used to produce high-temporal visualizations for several scientific codes, including mitGCM, OVERFLOW (computational fluid dynamics), fvGCM, WRF (a regional weather model) (Figure 6), and a solar convection code (Figure 7), delivering results in the form of thousands of images and hundreds of high-resolution movies. The high-temporal results have led to new insights for the researchers involved. The CV2Web tool has also been implemented for delivering real-time visualization results to the web—any researcher using a web browser can track the progress of his/her simulation and easily obtain the visualization results.

A parallel volume renderer that uses the M-on-N model to receive multiple 3D fields from a simulation in progress for visualization was implemented, and is being used to produce visualizations from large-scale simulations. The tool is being adapted for fvGCM and fvMMF (the Multiscale Modeling Framework), and work is in progress to support more advanced parallel visualization techniques, such as streamlines.



Figure 7: The hyperwall-2, showing a selection of results from a 2016-on-104 M-on-N run of a 2 billion grid cell solar convection simulation.

REFERENCES

- Ellsworth, D., B. Green, C. Henze, P. Moran, T. Sandstrom, 2006: Concurrent Visualization in a Production Supercomputing Environment. *IEEE Transactions on Visualization and Computer Graphics*, 2006.
- Lin, S.-J., B.-W. Shen, W. P. Putman, J.-D. Chern, 2003: Application of the high-resolution finite-volume NASA/NCAR Climate Model for Medium-Range Weather Prediction Experiments. *EGS - AGU - EUG Joint Assembly*, Nice, France, 6 - 11 April 2003.
- Shen, B.-W., G. Bryan, W.-K. Tao, C. Henze, S. Cheung, J.-L. F. Li, and P. Mehrotra, 2010: Coupling NASA Advanced Multi-Scale Modeling and Concurrent Visualization Systems for Improving Predictions of Tropical High-Impact Weather (CAMVis). *Earth Science Technology Forum (ESTF) 2010*. Arlington, Virginia, June 22-24, 2010

ECCO. <http://www.ecco-group.org/>

WRF. <http://www.wrf-model.org/>

ACKNOWLEDGMENTS

We would like to thank the following organizations for their support: NASA Earth Science Technology Office (ESTO); Advanced Information Systems Technology (AIST) Program; NSF Science and Technology Center; NASA Modeling, Analysis Prediction (MAP) Program; the Energy and Water Cycle Study (NEWS); the NASA High-End Computing (HEC) Program, and the NASA Advanced Supercomputing (NAS) facility at Ames Research Center, and the NASA Center for Computational Science (NCCS) at Goddard Space Flight Center.