Parallel visualization on leadership computing resources

T Peterka,[1] R B Ross,[1] H-W Shen,[2] K-L Ma,[3] W Kendall,[4] H Yu,[5]

[1]Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA

[2]Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210, USA

[3]Department of Computer Science, University of California at Davis, Davis, CA 95616, USA

[4]Department of Electrical Engineering and Computer Science, University of Tennessee at Knoxville, Knoxville, TN 37996, USA

[5]Sandia National Laboratories, California, Livermore, CA 94551, USA

E-mail: tpeterka@mcs.anl.gov

Abstract. Changes are needed in the way that visualization is performed, if we expect the analysis of scientific data to be effective at the petascale and beyond. By using similar techniques as those used to parallelize simulations, such as parallel I/O, load balancing, and effective use of interprocess communication, the supercomputers that compute these datasets can also serve as analysis and visualization engines for them. Our team is assessing the feasibility of performing parallel scientific visualization on some of the most powerful computational resources of the U.S. Department of Energy's National Laboratories in order to pave the way for analyzing the next generation of computational results. This paper highlights some of the conclusions of that research.

1. Introduction

Visual representations can provide compact summaries of large amounts of scientific data. As numerical simulations and experiments produce higher quantities of information, the challenge for visualization researchers is to continue to discover graphical methods that scale with the growth in data size and complexity. Since its recognition in 1989 as a core component in the scientific process [1], visualization has evolved with changing technologies, mainly influenced by consumer-grade graphics accelerators and relatively low-cost clustered architecture. Today, thanks to research in the field over the last twenty years, visualization has its own place in the science workflow pipeline shown in Figure 1. This is a data flow model—bytes move from one stage to the next. Data size in the hundreds of terabytes or petabytes, however, can overwhelm this model as the cost of managing data movement between stages overtakes the actual work done in the stages. The high cost of data movement is magnified by the relatively short duration of the visualization step. An alternative approach is to substitute some of the data flow with control flow—by moving some of the tasks to the data rather than the other way around.

This implies performing some or all of the analysis and visualization operations on the same supercomputers as the simulation. In the U.S. National Laboratories, the most powerful of these machines are called *leadership computing resources.* There are three potential benefits

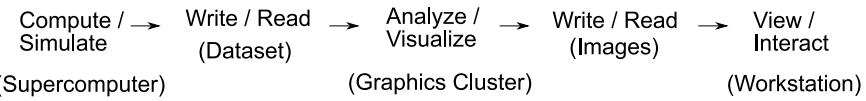

Figure 1. The stages in computational science: simulation, analysis, and interaction, in a customary data-flow pipeline. As the cost of data movement becomes prohibitive, performing visualization on machines traditionally reserved for computation can shorten overall time.

to this shift in performing visualization on leadership machines, compared to graphics clusters. First, end-to-end performance can improve because of reduced data movement. Second, this performance savings can afford higher quality visualization in the form of rendering improvement; for example, incorporating more complex lighting models or processing a larger portion of the total dataset in core. The third benefit is economic. As leadership machines continue to scale, deploying and operating proportional-size data analysis clusters becomes more costly. At some point, it will not be economically feasible for a leadership facility to maintain both types of architecture at these scales.

The U.S. Department of Energy has established Leadership Computing Facilities at Argonne and Oak Ridge National Laboratories. Each facility maintains a different supercomputing architecture: the Argonne Leadership Computing Facility (ALCF, [2]) has an IBM Blue Gene/P (BG/P) system while the Oak Ridge National Center for Computational Sciences (NCCS, [3]) maintains a Cray XT4 system. The BG/P has 160 K cores while the XT4 has approximately 31 K cores. The BG/P has access to a 5 petabyte PVFS parallel file system, while the XT4 is connected to a Lustre storage system.

As datasets grow in size and number of variables, the time that it takes to transfer each time-step (in the case of time-varying data) from storage to memory becomes significant. In fact, Peterka et al. [4, 5] showed that I/O can consume up to 90% of the total visualization time. One of the research directions of the SciDAC Institute for UltraScale Visualization [6] is to evaluate how parallel I/O techniques [7, 8] and libraries [9] can expedite the I/O stage in data-intensive visualization and analysis.

2. Selected Results of Case Studies

In one study, we benchmarked parallel volume rendering on the BG/P at the ALCF. The data for this test are simulation results produced by John Blondin of North Carolina State University and Anthony Mezzacappa of Oak Ridge National Laboratory. The VH-1 hydrodynamics code simulates the static accretion shock of a core-collapse supernova [10], and it generates a time-varying dataset. Each time step is a netCDF file with five variables.

The parallel volume rendering algorithm consists of three stages. In the first stage, I/O, the time-step file is read into memory by all processes in parallel. The second stage, rendering, has all processes concurrently performing software ray-casting through their respective data subdomains. At the end of the second stage, each process contains a volume-rendered image of its subdomain. The images are merged together in the third stage, called compositing. The relative time that each stage requires changes with the system scale; our experiments range from 64 processes to 32 K processes. The data size ranges from 1120^3 elements to 4480^3, and the image size ranges from 1024^2 to 4096^2 pixels.

A performance summary of three data sizes and system scales appears in Figure 2. The time axis represents the total end-to-end time needed to visualize a single time step, including I/O, rendering, and compositing. The 1120^3 dataset with 1024^2 image size scales to 16 K processes; beyond this point the subdomains are too small to justify using more cores. The upper two curves show scalability out to 32 K processes on larger data and image sizes. The uppermost curve represents a data size of 4480^3 with an image size of 4096^2; with 32 K processes, this is

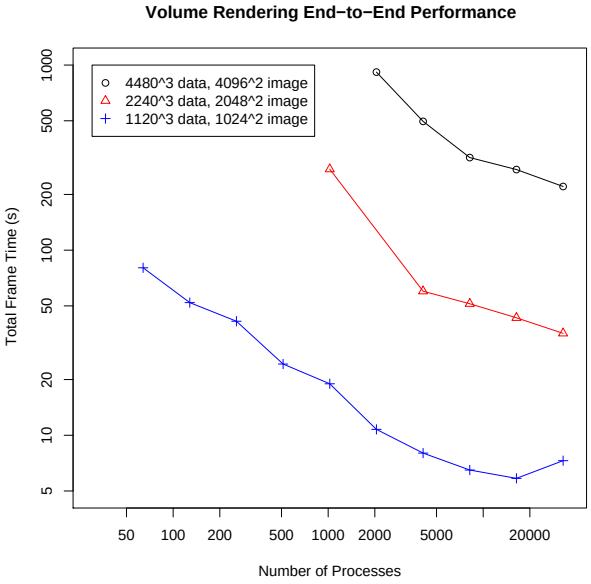

Figure 2. Scalability over a variety of data, image, and system sizes. Several performance points are available for each curve; one may select a smaller system scale, for example, based on resource availability, and still complete the visualization in a matter of seconds or minutes.

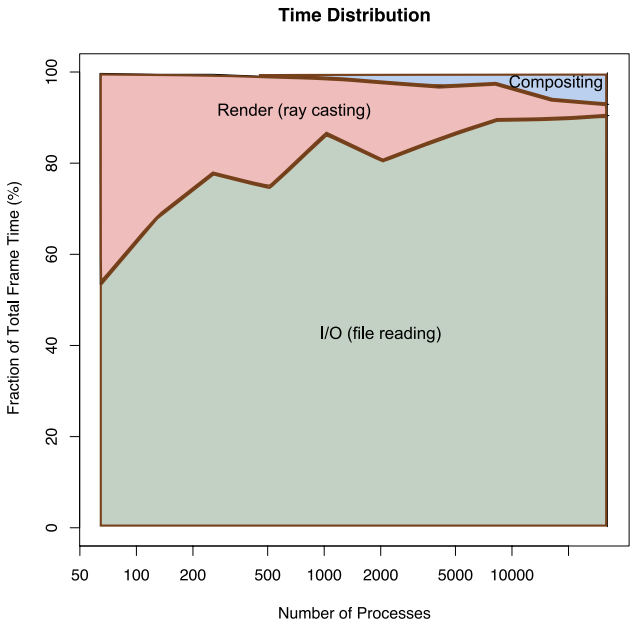

Figure 3. The relative percentage of time spent in I/O, rendering, and compositing as a function of system size. At large system scale, visualization is dominated by data movement: primarily by I/O and secondarily by compositing. The data size and image size for this test is 1120^3, and 1024^2, respectively, but the pattern shown is typical of all of our tests.

one of the largest in-core parallel visualization results published to date.

Figure 3 illustrates how the relative proportion of time spent in each stage changes as the system scale grows. At the left side of the plot, rendering occupies 50% of the total time, but as the system scale increases, the fraction of time spent in rendering is almost negligible. This is one reason why software rendering is practical on relatively low-power processors such as BG/P's. Most of the time is spent in I/O, and even compositing overtakes rendering at 8 K cores. This pattern is typical of all of our tests: large scale visualization is dominated by data movement.

Because I/O is such a critical part of the large-scale visualization process, our team devotes considerable effort to its study. Figure 4 shows the result of one such experiment. The diagrams represent a log of disk activity, when reading one variable out of the five variables contained in a netCDF dataset. The dark rectangles represent physical file blocks that are read in as a result of this collective I/O request. Starting with the original implementation (left), reorganizing the data layout and using a new 64-bit version of the netCDF format (right) resulted in a nearly three times shorter I/O time. Comparing the left and right diagrams, fewer blocks total blocks are accessed in the right diagram, and their physical placement on disk is contiguous. Our collaborators in the SciDAC SDM center provided an early release of the new netCDF format for this research.

One pattern that often occurs in scientific data organization is one time-step per file, resulting in many files for a single dataset. In a study on the Cray XT reading several hundred time-steps of NASA's Moderate Resolution Image Spectroradiometer (MODIS) dataset, our team is

researching how to efficiently read all of these files in parallel into the memory of thousands of cores. This situation occurs, for example, when comparisons across time steps are required in the analysis. By using a greedy algorithm that assigns files or portions of files to processes in a round-robin fashion, and by placing files on all 144 Lustre *Object Storage Targets* (OSTs), we achieved the 28 GB/s aggregate read bandwidth shown in Figure 5. This is approximately 75% of the maximum peak bandwidth measured by the IOR benchmark.

Many application teams are experimenting with "I/O subsetting," or performing I/O operations from only a fraction of the nodes used for computation. We found that controlling the number of processes that perform the actual writing of the output image (*writers*) improves performance and memory usage. The right plot in Figure 6 shows that between 32 and 512 writers is appropriate. Figure 6 also shows that we are able to achieve better writing performance by controlling the subsetting through the application (right), rather than by supplying hints to MPI-IO (left). In both cases a minimum time occurs at 8 or 16 aggregators (the MPI-IO entities actually performing the writing on behalf of MPI processes), but by controlling the number of writers explicitly, this optimal setting is achievable in a wider range of configurations.

3. Summary

As leadership computing machines and the simulations that run on them extend to petascale and beyond, analysis of the results must scale accordingly. In order to meet this challenge, we are studying what circumstances favor performing more of the analysis and visualization operations directly on these flagship systems that have tens of thousands of cores, a high-throughput interconnect, and parallel storage infrastructure. As our research shows, these characteristics are critical at large scales because analysis and visualization are dominated by data movement. Through careful study of data movement patterns, especially to and from storage, entire datasets

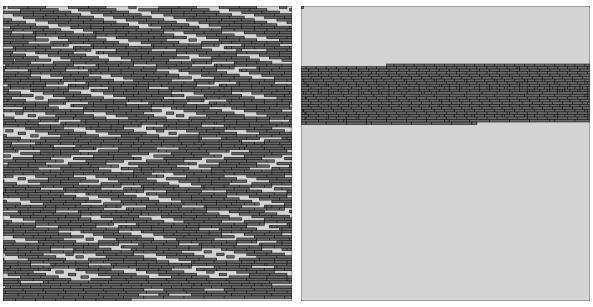

Figure 4. Log of disk activity. Dark rectangles represent physical file blocks that are touched in order to retrieve one out of five variables in a netCDF data file. Left: most of the file is read, resulting in wasted effort. Right: Through our collaboration with the SciDAC SDM center, a new 64-bit version of netCDF results in fewer physical blocks accessed, which are contiguous. This resulted in a three times speedup in I/O time.

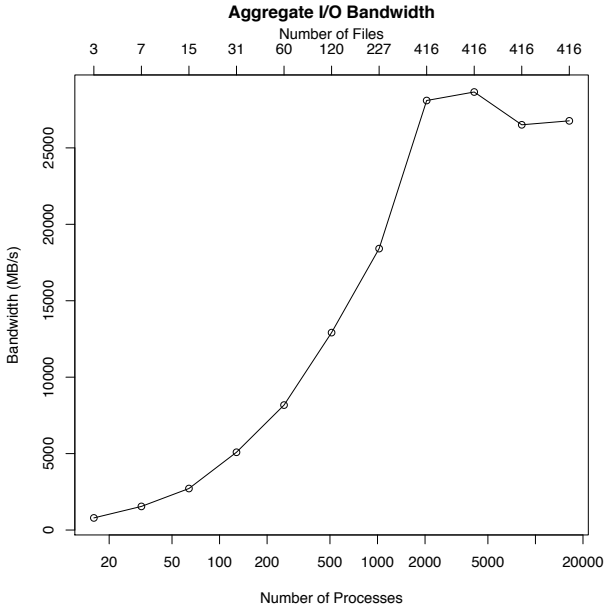

Figure 5. Aggregate bandwidth for reading up to 416 time steps of the MODIS full resolution dataset simultaneously into up to 16 K processes peaks at 28GB/s, or 75% of the IOR benchmark.

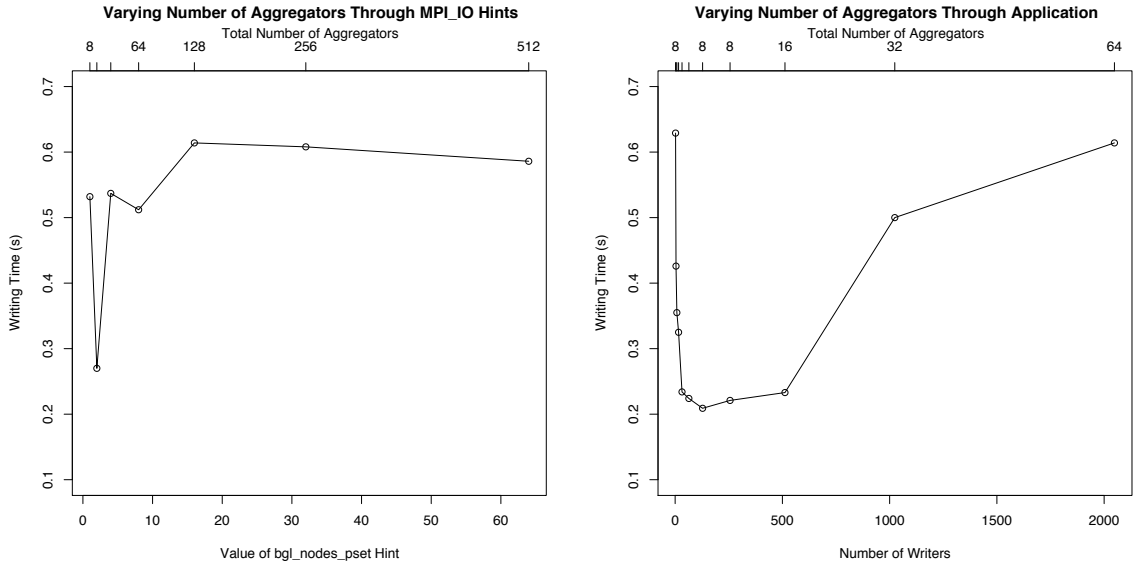

Figure 6. The effect of I/O subsetting, via MPI-IO hints (left) vs. using a custom communicator in the application code (right). As the right plot shows, manually assigning a number of processes to perform parallel image writing produces better a shorter time over a wider range of settings.

can be analyzed in-core and in full resolution. We are experimenting with other visualization techniques and more complex data sets that involve both scalar and vector time-varying data. We will continue to produce benchmarks that other researchers can use to perform parallel visualization on leadership computing resources.

Acknowledgments

We thank John Blondin and Tony Mezzacappa for their dataset and valuable feedback. We gratefully acknowledge the use of the resources of the Argonne Leadership Computing Facility at Argonne National Laboratory and the National Center for Computational Sciences at Oak Ridge National Laboratory. This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357. Work is also supported by DOE with agreement No. DE-FC02-06ER25777.

References

[1] DeFanti T A, Brown M D and McCormick B H 1989 *Computer* **22** 12–25 ISSN 0018-9162
[2] 2009 *Argonne Leadership Computing Facility* http://www.alcf.anl.gov/
[3] 2009 *National Center for Computational Sciences* http://www.nccs.gov/
[4] Peterka T, Yu H, Ross R and Ma K L 2008 Parallel volume rendering on the ibm blue gene/p *Proc. Eurographics Parallel Graphics and Visualization Symposium 2008* (Crete, Greece)
[5] Peterka T, Ross R, Yu H, Ma K L, Kenall W and Huang J 2008 Assessing improvements in the parallel volume rendering pipeline at large scale *Proc. SC 08 Ultrascale Visualization Workshop* (Austin TX)
[6] 2009 *SciDAC Institute for Ultra-Scale Visualization* http://ultravis.ucdavis.edu/
[7] Carns P, Ligon W B I, Ross R and Thakur R 2000 Pvfs: A parallel file system for linux clusters *Proc. 4th Annual Linux Showcase & Conference* (Atlanta, GA) p 28
[8] Thakur R, Gropp W and Lusk E 1999 Data sieving and collective i/o in romio *Proc. 7th Symposium on the Frontiers of Massively Parallel Computation* pp 182–189
[9] Li J, Liao W k, Choudhary A, Ross R, Thakur R, Gropp W, Latham R, Siegel A, Gallagher B and Zingale M 2003 Parallel netcdf: A high-performance scientific i/o interface *Proc. Supercomputing 2003* (Phoenix, AZ)
[10] Blondin J M, Mezzacappa A and DeMarino C 2003 *The Astrophysics Journal* **584** 971

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government.