Visualization Viewpoints

Editor:
Theresa-Marie Rhyne

Increasing Scientific Data Insights about Exascale Class Simulations under Power and Storage Constraints

James Ahrens
Los Alamos National Laboratory

Over the past three decades, supercomputing systems have progressed to compute the results of extremely accurate scientific simulations. These simulations help us understand complex real-world phenomena such as our climate, energy sources, and the progression of natural disasters. Additional computing power supports the computation of higher-quality simulations, and that in turn provides higher fidelity results. Using the number of floating-point operations per second (flops) as a measure of progress, we have progressed through terascale machines that compute 10^{12} flops to petascale machines that compute 10^{15} flops. A number of open source efforts provide a robust scalable visualization and analysis capability such as ParaView (www.paraview.org) and Visit (https://visit.llnl.gov) for these levels of performance. These tools traditionally focus on a postprocessing approach. That is, during a simulation run, representative results are written to storage for later visualization.

The international community is looking toward the next jump in performance: exascale supercomputers that compute 10^{18} flops. Creating an exascale simulation environment will be a significant challenge due to power and storage technology trends. Responding to these challenges will require rethinking and reframing how we approach visualization and analysis at the exascale. A key difference from the terascale and petascale eras is the need to keep track of a cost per insight in terms of power and storage used. This notion of constraints on our insights challenges the premise of our traditional postprocessing approach.

In a traditional postprocessing-oriented visualization and analysis approach, temporal simulation snapshots are saved at regular intervals. This approach incorporates the process of saving checkpoints for later restart in case of errors. Traditionally, these full simulation checkpoint snapshots and additional smaller visualization and analysis data are interactively analyzed after the simulation run is complete. The visualization and analysis community has identified this approach as unworkable at the exascale because of power and storage constraints. An emerging consensus is that significantly more visualization and analysis should occur in situ—that is, during the simulation run while the data is resident in memory. Thus, emerging research challenges include exploring what types of analysis questions can be answered during postprocessing with compact data products that are generated in situ and what mathematical or statistical techniques will best support this process.

Current Constraints

Power constraints are driven by reducing the many financial costs, including facility, power, and cooling costs, associated with the massive power requirements that are projected for an exascale machine without research and development interventions. Specifically, the United States Department of Energy's exascale strategy identifies target goals for peak performance to increase three orders of magnitude while system power is only targeted to increase by a factor of two. To keep within an extremely limited power budget, locality during computation is extremely important. The most expensive operation is data movement, from both the power and performance perspective, moving data up from the CPU out through the memory hierarchy including out to persistent storage and the network. Figure 1 shows the approximate power cost of moving a single bit.[1]

0272-1716/15/$31.00 © 2015 IEEE

Storage constraints are also driven by financial costs including power costs. Future storage technology projections suggest that the gap between both capacity/bandwidth and flops will widen as we move toward exascale. Therefore, we expect the storage system of an exascale supercomputer to be smaller and slower compared in a relative way with the peak flops of today's generation of supercomputers for a proportionally similar level of investment.

Deliberate Analysis Choices Are Necessary

In a traditional postprocessing approach, during a simulation run, full simulation snapshots are saved. This has led to the belief that these snapshots can answer arbitrary analysis postprocessing questions because "all the data has been saved." However, as we have discussed, this is not necessarily true for the time domain. A related belief about in situ techniques is that automatic selection of data at runtime reduces the type of questions that can be asked about the data during postprocessing analysis. It is important to recognize the traditional postprocessing approach of saving full simulation snapshots is, in and of itself, an inherently in situ activity. Saving full simulation snapshots in time is simply one choice among many for extracting data and information from running simulations.

An alternative perspective is to consider what scientific insights are sought, balanced by power and storage constraints, and then output only the minimal analysis data needed during the simulation run. In the observational/experimental community, preplanned data reducing streaming analysis is common practice. Custom software and hardware accelerators are typically employed to reduce and analyze data in real time for accelerator physics, fusion reactors, and cybersecurity. Our focus on in situ approaches aligns the supercomputing community with the observational/experimental community supporting synergistic approaches in the future. In this case, there are key research questions to answer: How general and with what quality can analysis questions be answered during postprocessing with compact data products that are generated in situ? What new mathematical or analysis techniques will support this process?

In Situ Sampling

During in situ data analysis, the analyst has access to the entire simulation data in all its complexity, including spatial, temporal, multivariate, and variable type domains. This data is available only briefly at simulation runtime when it resides in memory, and it is then overwritten when the simulation advances. Given our budgeting constraints, it becomes clear that in situ analysis is a form of sampling. The traditional workflow samples fully on the spatial, multivariate, and variable type domains at the expense of sampling fully in the temporal domain.

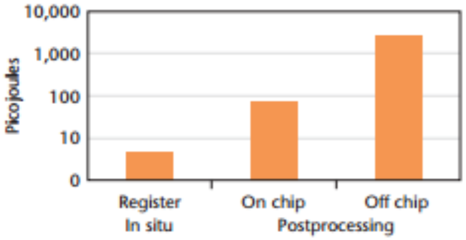


Figure 1. Approximate power cost of moving a single bit, assuming the bit starts in a register and moves to the location listed on the *y* axis.[1] Our expectation is that the power cost of in situ analysis is significantly less than postprocessing analysis because in situ occurs while simulation data is in register or memory and has not been written to storage.

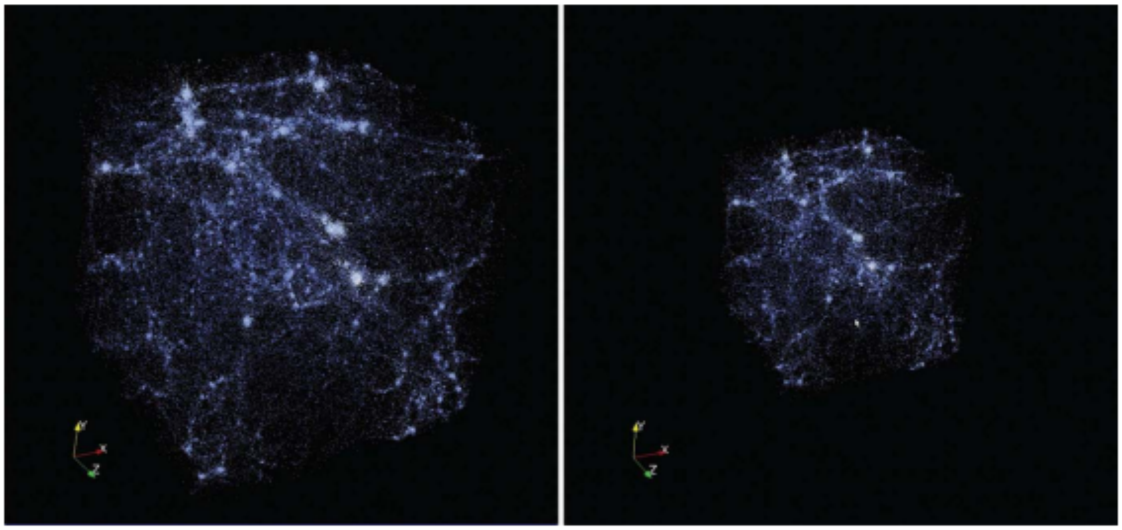
Spatial Sampling

Simulation scientists have the opportunity to significantly increase the quality of their analysis results by choosing how to sample from each domain. The quality of their results can be measured through combined in situ sampling and uncertainty quantification techniques. For example, in our work, we statistically sample using a stratified random sampling approach on the MC^3 cosmological particle simulation. We store these samples in a level-of-detail organization for later interactive progressive visualization and feature analysis. By sampling during the simulation, we are able to analyze the entire particle population to record full population statistics and quantify sample error.[2] Figure 2 shows this sampling visually. In the figure, a set of two camera positions that differ by zoom level of the same comsology simulation result are shown. The visualization system streams in the particles it needs from the stored multiresolution sampled result to achieve the same screen density.

The key idea of this work is to only save and use the amount of data needed to complete the visualization and analysis task. Thus, conceptually with this technique we could sample and render a massive petascale (10^{15}) or exascale (10^{18}) sized result to an image with on the order of only 10^6 pixels.

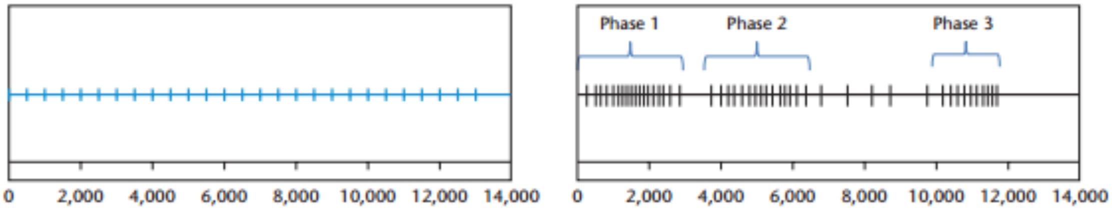
Temporal Sampling

Our goal is to reduce the simulation data stream to a compact analysis product that fits within a given budget. This reduction does not have to be via a



(a) **(b)**

Figure 2. A visualization of the MC^3 cosmology simulation. (a) Full simulation visualization. (b) In this second image, the number of particles sampled and rendered is significantly less and is chosen to achieve a constant screen density.



(a) **(b)**

Figure 3. Time step plots for an asteroid deflection simulation. (a) Traditional plot when time steps are saved on a regular basis every 500 time steps. (b) Plot of when time steps are saved based on their entropy. Images with the highest entropy that are saved to storage are recorded as tick marks in this plot.

statistical sampling; visualization operations and feature extraction algorithms can also be considered a type of sampling strategy. An interesting way to approach the inclusion of the most important data within a budget is to prioritize data using a greedy algorithm saving the highest priority information as the simulation progresses. For example, in recent work we measured temporal entropy in a running simulation.[3] A memory buffer collected time steps with the highest entropy by having time steps with higher entropy overwrite ones with lower entropy. The resulting collection of high entropy time steps provides a summary of the phases of the simulations in which the most change occurs. Figure 3 presents plots of when imagery is saved from an asteroid deflection simulation. The *y* axis is the time-step number. A tick mark records that an image was saved at a specific time step.

Figure 4 show the results from a 2D asteroid deflection scenario simulation. The sequence of images provides a content driven summary of the phases of the simulations. The images were selected as representatives of the three phases shown in Figure 3b.

The sampling and automated selection techniques highlighted here are offered as ideas as to how supercomputing visualization and analysis will change in the future. As the supercomputing visualization and analysis community transitions to in situ approaches, there are numerous opportunities for new research and development in representations, algorithms, and automation techniques to enable scientific discovery at the exascale. The community is open to new ideas and looks forward to your participation as we tackle these emerging challenges.

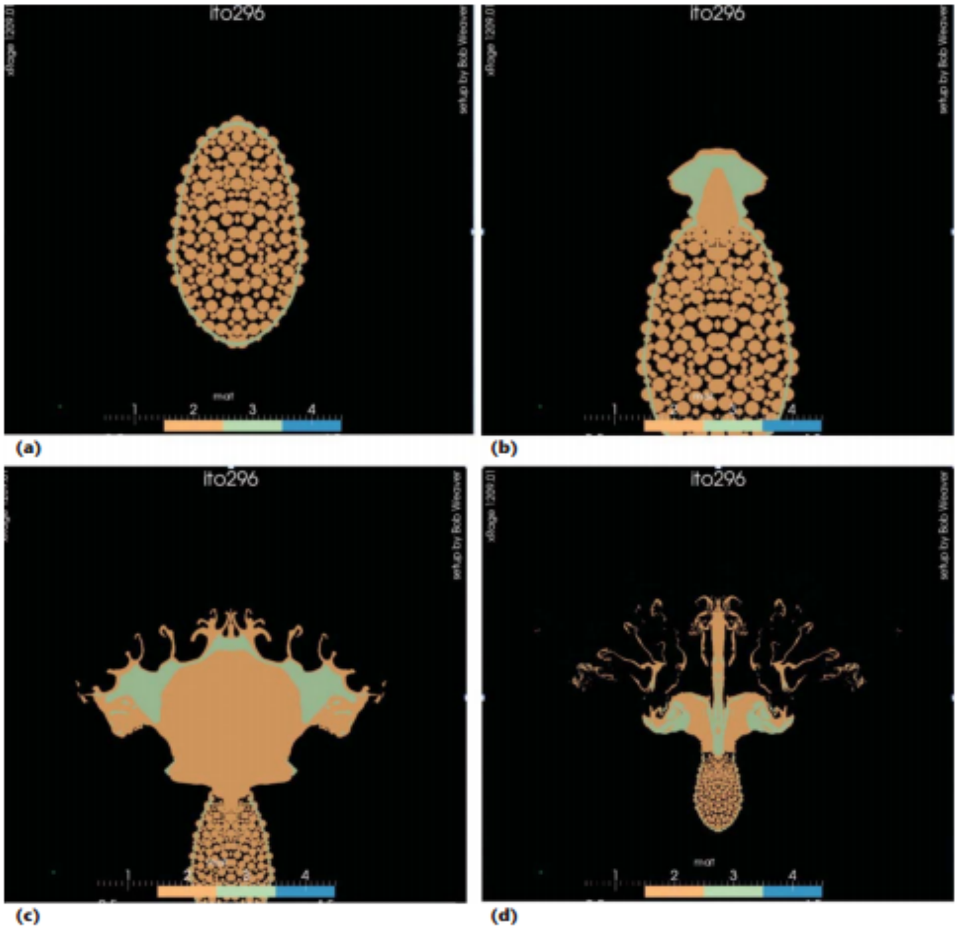


Figure 4. A sequence of images from an asteroid impact simulation. The times of the images in the sequence are (a) 0, (b) 1153, (c) 4831, and (d) 11405, illustrating the three phases shown in Figure 3b. The camera automatically adjusts to keep the highest entropy portion of the simulation centered in each view.

Acknowledgments

I work with a great team of people at Los Alamos. More information about our team members and projects can be found at http://datascience.lanl.gov. The work is supported by the Department of Energy's Office of Science, Advanced Scientific Computing Research (ASCR) program office (Lucy Nowell) and National Nuclear, Security Administration (NNSA) program office (Thuc Hoang).

References

1. J. Shalf, S. Dosanjh, and J. Morrison. "Exascale Computing Technology Challenges," *Proc. High Performance Computing for Computational Science* (VECPAR 2010), 2011, pp. 1–25.
2. J. Woodring et al., "In situ Sampling of a Large Scale Particle Simulation for Interactive Visualization and Analysis," *Computer Graphics Forum*, vol. 30, no. 3, 2011, pp. 1151–1160.
3. B. Nouanesengsy et al., "ADR Visualization: A Generalized Framework for Ranking Large-Scale Scientific Data using Analysis-Driven Refinement," *Proc. IEEE Symp. Large-Scale Data Analysis and Visualization* (LDAV), 2014.

***James Ahrens** is a technical staff member at the Los Alamos National Laboratory and the director of the data science at scale project. Contact him at ahrens@lanl.gov.*

Contact department editor Theresa-Marie Rhyne at theresamarierhyne@gmail.com.

Selected CS articles and columns are also available for free at http://ComputingNow.computer.org.