# In Situ Visualization at Extreme Scale: Challenges and Opportunities

Kwan-Liu Ma
University of California at Davis

The advent of supercomputers gives scientists the power to understand the evolution of the universe, predict climate change and variability, design more efficient energy, and study many other complex scientific phenomena and engineering design problems with numerical simulations. As the speed and capacity of supercomputers continue to increase, scientists begin to model physical phenomena and chemical processes at an unprecedented level of detail and accuracy, making possible new discoveries and breakthroughs in many areas of study. The rate of increase in supercomputing power is not slowing down. For example, the U.S. Department of Energy has made a significant investment through its SciDAC (Scientific Discovery through Advanced Computing) program on computational science research, scientific computing software tools, and hardware infrastructure. Similar investments are being made in China, Japan, and some of the European countries. Petascale computers, capable of performing $10^{15}$ operations per second, have become available and we are moving toward exascale computing (i.e., achieving $10^{18}$ operations per second). While making a simulation code utilize the full power of a petascale supercomputer remains a daunting task, a bigger challenge facing scientists is handling and analyzing the output of the simulation. The current practice is to dump a temporal subset of the data to disk and examine the data in an offline post-processing step using a dedicated visualization machine, as shown in Figure 1. A large-scale simulation typically runs from tens to hundreds of thousand time-steps but scientists cannot afford to store every time-step output. They usually only save every 3-5 hundred time-steps, resulting in several terabytes of data. At petascale and exascale, the amount of data would be several orders of magnitude greater, making the simulation spend most of the supercomputing time doing I/O and take much longer to finish. Even in the case that scientists can afford to keep most of the data for analysis, they must transfer the data to a machine that has sufficient capacity and processing power to do the desired visualization and analysis jobs. However, the data transferring cost is likely too high to be acceptable and the visualization machine would need to be almost as powerful as the supercomputer producing the data. The alternative is keeping and looking at an even smaller temporal subset (and in some cases also a spatial subset) of the data, which defeats the purpose of performing the original high-resolution simulation enabled by petascale computing.

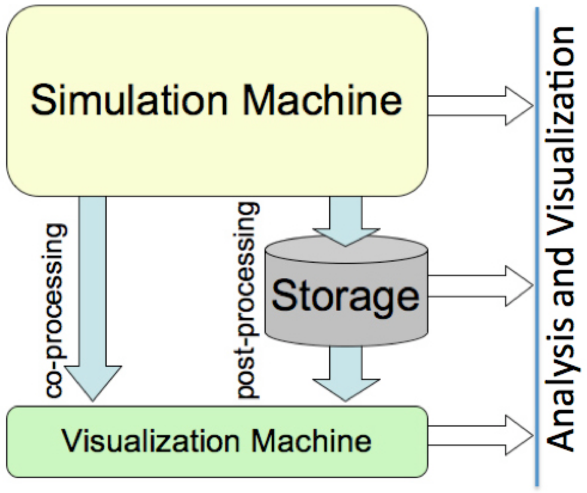

Figure 1. *Transferring the whole data generated by a petascale simulation to a storage device for post-processing visualization or a visualization machine for co-processing could become a serious bottleneck because I/O would take most of the supercomputing time.*

A better approach is not to move the raw data, or to keep the data that must be stored and moved to a minimum. One strategy is to have both simulation and visualization calculations run on the same parallel supercomputer so the data can be shared, as shown in Figure 2. Such simulation-time *co-processing* can render images directly or extract features - which are much smaller than the full raw data - to store for later examination. As a result, reducing both the data storage and transfer cost early in the data analysis pipeline optimizes the overall scientific discovery process. Such simulation-time visualization, also known as *in situ* visualization, is not a new concept. In practice, however, this approach was seldom adopted because of two reasons. First, most scientists were reluctant to use their supercomputer time for visualization calculations, especially when the calculations are expensive. Second, it could take a significant effort to couple the parallel simulation code with the visualization code. In particular, the domain decomposition optimized for the simulation is often unsuitable for parallel visualization, resulting in the need to replicate data for speeding up the visualization calculations. Nevertheless, it becomes clear that in situ visualization is a feasible solution for the upcoming extreme-scale data problem presented by scientific supercomputing. It is a particularly desirable solution because during the simulation time all relevant data about the simulated field and any embedded geometry are readily available for the visualization calculations. All such relevant data and information would be prohibitively expensive to collect again and compute during a post-processing step. We can conduct in situ processing to compress data, extract salient features from the data, create a data hierarchy, and/or build indexing for fast data access. We can conduct in-situ visualization to achieve runtime monitoring and even steering of the simulation, or, based on domain knowledge, to create snapshot images or an animation characterizing the simulation. Figure 3 displays selected time steps from in situ visualization of a turbulent combustion simulation running on the Cray XT5 supercomputer operated at the Oak Ridge National Laboratory [1].

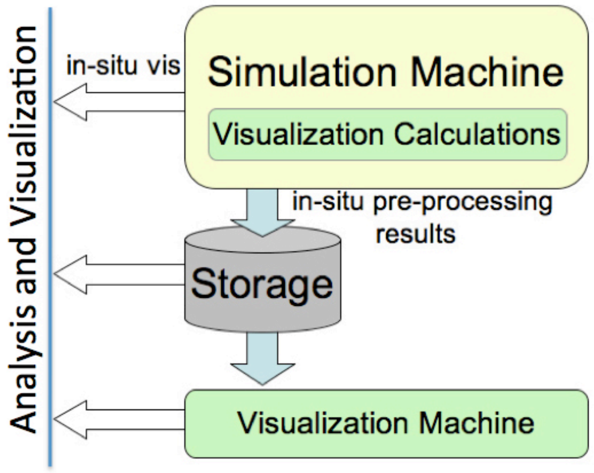
Analysis and Visualization
in-situ vis
Simulation Machine
Visualization Calculations
in-situ pre-processing results
Storage
Visualization Machine

Figure 2. *A more feasible approach to data analysis at extreme scale is to reduce and prepare the data in situ for subsequent visualization and data analysis tasks. The reduced data, which typically are several orders of magnitude smaller than the raw data, can greatly lower the following data transferring and storage costs.*

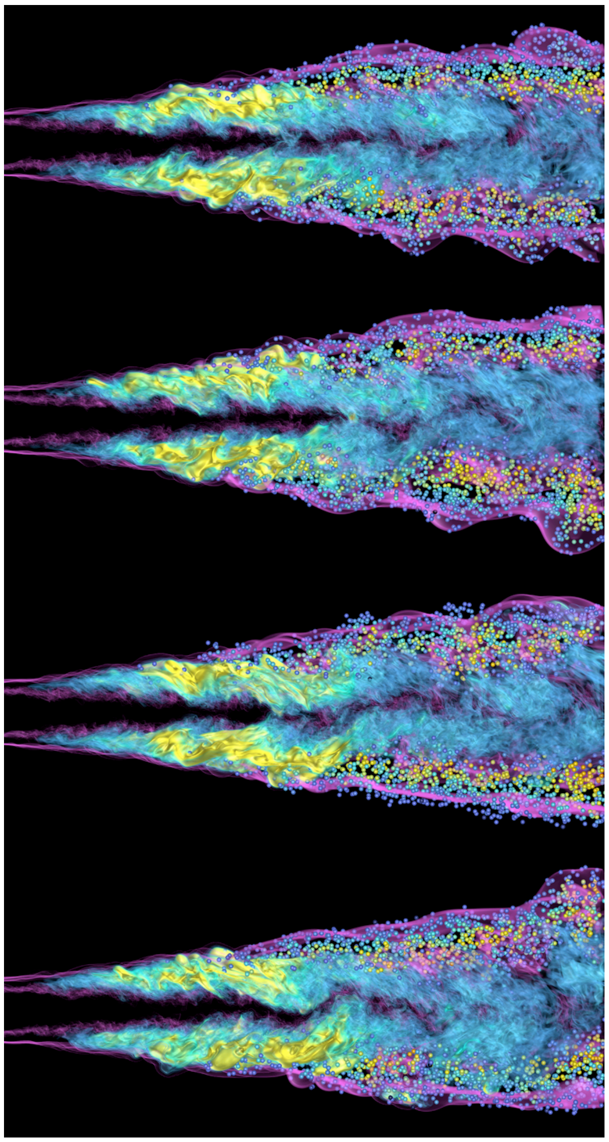

Figure 3. In situ visualization of a large-scale turbulent combustion simulation running on 6,480 processors of a Cray XT5. Four selected time steps are displayed to show mixing of $Y\_HO_2$ concentration and particle data colored based on Y_OH. (Image was created by Hongfeng Yu of the Sandia National Laboratories and the SciDAC Institute for Ultrascale Visualization, and data was provided by Jacqueline Chen of the Sandia National Laboratories.)

In situ visualization presents many new challenges to both the simulation and visualization scientists. Before this approach can be realized, several questions must be answered first:

- How do simulation and visualization calculations share the same processor, memory space, and domain decomposition?
- What fraction of the supercomputer time

should be devoted to in situ data processing/visualization? As in situ visualization becomes a necessity rather than an option, scientists need to accept visualization as an integral part of the simulation.

- What are the data processing tasks and visualization operations best performed in situ? To what the extent does the monitoring scenario stay relevant, and how is monitoring effectively coupled with knowledge-driven data reduction.
- What are the unique requirements of in situ visualization algorithms?
- As we store less raw data to disk, what supplemental information should be generated in situ?
- To provide generic support for in situ visualization, how do we abstract out the complexity of the simulation, mesh structures, parallel implementation, etc.?
- Can existing commercial and open-source visualization software tools be directly extended to support in situ visualization at extreme scale?

To answer these questions, and to understand the impact of this new approach on the simulations and subsequent visualization tasks, we must embark upon a new line of research investigations with close collaboration with simulation scientists. The U.S. National Science Foundation's cross-cutting programs such as the Cyber-Enabled Discovery and Innovation and the Department of Energy's SciDAC program encourage such research efforts and open many opportunities for visualization researchers. The remaining sections of this article address two crucial research topics: parallel algorithms and data reduction methods.

## Parallel Algorithms

Since all large-scale simulations run on parallel supercomputer computers, parallel algorithms for both data reduction and rendering must be devised to realize in situ visualization. Due to the increasing scale of the hardware, simulations, and output data, the parallel algorithms must be highly scalable. Some of the parallel visualization algorithms developed previously were no longer usable. For example, parallel image compositing algorithms such as binary swap and SLIC only scale up to hundreds of processors. Even though refined algorithms are being introduced recently showing improved performance using thousands of processors [2], petascale computing involving several hundred thousands of processors demands new and more scalable algorithms.

It is challenging to design scalable parallel in situ visualization algorithms because of several reasons. First, as pointed out earlier, the visualization algorithms should use the same domain decomposition made by the simulation to avoid data duplication and interprocessor communication as much as possible. Even though the same domain decomposition is used, most distributed visualization algorithms still require duplicating data at partition boundary. The resulting memory overhead must be minimized. For in situ processing, communicating data as needed rather than replicating data up front likely gives better results. Second, when using tens of thousands of processors, interprocessor communication, if not carefully managed, would become a bottleneck, making the associated visualization solution not acceptable. An effective approach seems to be precomputing an optimal schedule for packing and ordering communication. Third, the visualization computation should take a small fraction of the overall simulation time. Sophisticated visualization methods though offering visually compelling results often are not acceptable. A plausible direction to pursue is making use of domain knowledge whenever possible to subsample the data to reduce the amounts of computation. On the other hand, as in situ visualization technology becomes more mature and its benefit becomes substantial, scientists will be willing to trade off simulation time for more analysis. In fact, in situ visualization suggests scientists to rethink the computational scientific discovery process. Finally, for volume rendering, how do we derive appropriate color and opacity transfer functions to best characterize the modeled phenomena in both the spatial and temporal domains of the data? It is clear we need to develop an adaptive method without constantly acquiring global information about the spatial and temporal domains.

In many applications, it is desirable to visualize and understand the intrinsic interaction between different variables over time. There may be tens or even hundreds of variables and chemical species used in a simulation. Such multivariate visualization is thus best done in situ since all relevant data are available. In situ visualization enables scientists to visualize the full extent of their data in combination and at the fidelity and density not possible with postprocessing methods. Figure 4 shows simultaneous visualization of three variables from the same turbulent combustion simulation used in Figure 3. Figure 5 shows simultaneous pathline visualization and volume rendering of the magnitude of angular momentum from a supernova simulation. This visualization was made based on two vector fields: velocity and angular momentum. The corresponding animation allows the scientists to see how the bundles of counter rotating flow lines interact with the horizontally moving shock. Such type of visualization is more costly to make since it requires six times more data as well as calculations that are difficult to parallelize. The visualization in Figure 3 was created in an offline postprocessing step using a scalable parallel pathline visualization algorithm was introduced [3], but that parallel algorithm is not suited for in situ visualization. We need to develop either a clever way to trace particles in situ or a new way to depict vector fields that will scale with the size of the supercomputer.

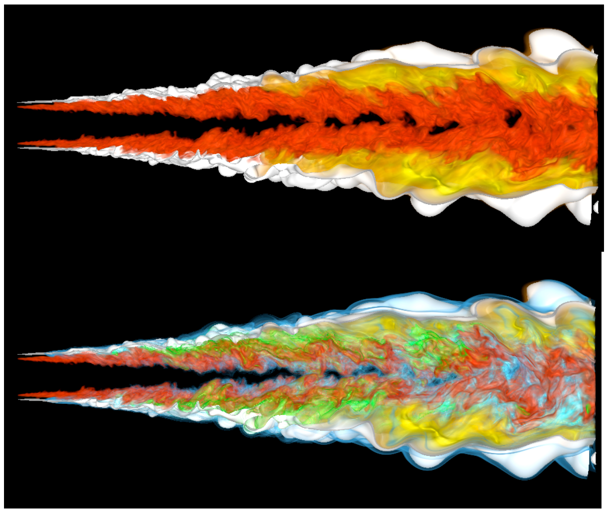
Figure 4. Simultaneous visualization of temperature, fuel, and mixture fraction isosurface from a turbulent combustion simulation. Top: Temperature and mixture fraction isosurface representing the most active reaction layer. Bottom: Adding fuel concentration to the picture to show how it relates to temperature change and the reacting flow. (Image was created by Hongfeng Yu, and data was provided by Jacqueline Chen.)

## Data Reduction Methods

For petascale simulations, reducing data in situ is very appealing since the reduction can benefit all the subsequent data movement and processing.
Data reduction may be achieved with compression or by extracting features in the data. What exactly is a *feature*? Different disciplines clearly carry different definitions of features. Generally speaking, a feature is a small subset of the raw data isolated with domain knowledge and represents a particular physical structure, pattern, or event of interest. Some examples include vortices, shocks, eddies, critical points, etc. These features can be categorized, and used to characterize and break down the overall modeled physical phenomenon. The saving in storage space with feature extraction with or without contextual information can be very significant, much greater than the compression based methods; however, scientists do not always know completely what to extract and track in their data. In three-dimensional volume visualization, for example, a feature is commonly referred to as a certain space-time coherent region or object of interest. Basic visualization algorithms exist for feature identification, extraction, and tracking, which incorporate principles from image processing, computer vision, and machine learning. Feature extraction and tracking is a very common approach to visualizing time-varying flow data [4]. In order to track features through time, features must be correlated across a sequence of time steps. This task is frequently referred to as the *correspondence problem*. Many of the previous algorithms correspond features based on whether or not their regions overlap in adjacent time steps. Other algorithms correspond features based on attributes of the region such as position, size, shape, orientation, or topology. All these correspondence based, feature tracking algorithms have to make special cases for regions that split or merge over time. They also have to be careful with features that move far enough between time steps such that their regions do not overlap in consecutive time regions. A novel and more universally applicable approach is to employ learning-based or evolutionary methods. The basic idea is to capture scientists' domain knowledge through interactive visualization augmented with an intelligent system, and then apply that knowledge to feature extraction and tracking. Better results can be obtained as more knowledge is accumulated over time. However, the robustness of this approach remains to be determined.

Since scientists typically store only one out of every few hundred time steps due to the storage space limitation, the resulting temporal gaps compound the problem of feature tracking. A longer time interval between time steps means that a feature must move farther enough to be trackable. Therefore, it is very beneficial to perform feature extraction and tracking at the simulation time, so that the tracking algorithm can utilize all the time steps that would otherwise be skipped or discarded in a post-processing setting. While scientists need to understand the benefit of in situ visualization comes at certain cost, it is possible to reduce the cost. For example, we should make tracking use information from the smallest possible time interval. As such, correspondence-based feature tracking algorithms would not work well because they require the calculation of regions in each time step independently before any tracking is performed. In addition, the storage overhead of these algorithms and those based on machine learning is often too high to make them suitable for in situ processing. A new prediction-correction algorithm recently introduced for tracking volumetric features has shown great promise. It makes the best guess of the feature region in the next time step, followed by growing and shrinking the border of the predicted region to coherently extract the actual feature of interest [5]. This algorithm makes use of the temporal-space coherency of the data to accelerate the extraction process while implicitly solving the tedious correspondence problem. Most importantly, this algorithm is straightforward to parallelize. The development of similar algorithms is needed for visualizing other types of features. In situ feature visualization can guide the scientists to set the appropriate level of data reduction, leading better data throughput of the simulations.

If data has to be reduced as a result of either subsetting or feature extraction, the corresponding information loss must be conveyed to the users. Quality assessment thus plays a crucial role in large-scale data analysis and visualization since in many cases the reduced version of data, rather than the original version, is used for evaluating the simulation and modeled phenomena. To compare the quality of the reduced or distorted data, it is imperative to identify and quantify the loss of data quality. Most existing data quality metrics, such as the mean square error and the peak signal-to-noise ratio, require access to the original data. Although these metrics are easy to compute, they do not correlate well with perceived quality measurement [6]. More importantly, these metrics are not applicable to petascale applications simply because the original data is too large to acquire or compare in an efficient way. One viable approach is to compute feature information in situ [7]. Then we are able to use it as the basis metric for offline quality assessment without the need to access the original data. This may be achieved by computing the distance of feature components of the reduced or distorted version of data with those derived from the original data, which gives us an indication of quality loss in relation to the original data. Such feature information also enables us to perform a cross-comparison of data with different reduction or distortion types.

Finally, to enable interactive data analysis and visualization of the reduced data, efficient organization of the data to facilitate fast access is key. Data subsets need to be indexed rather than stored as flat files [9].

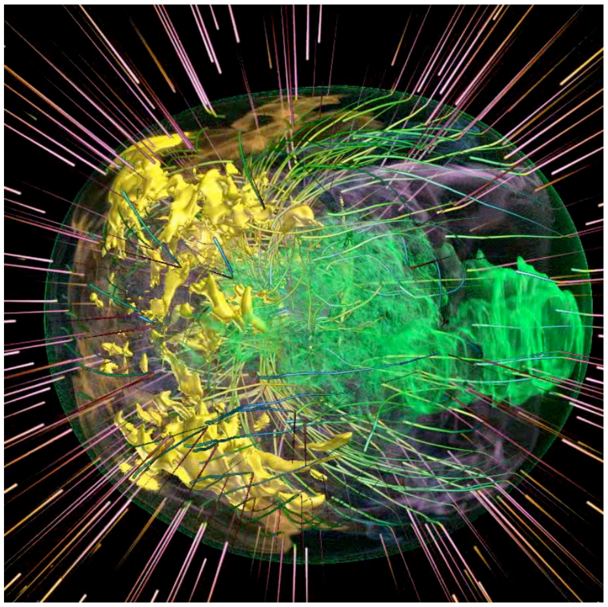

Figure 5. Simultaneous pathline visualization and volume rendering of the magnitude of angular momentum from a supernova simulation. Such visualization is desirable, but generating it in situ with the simulation requires the development of new scalable parallel algorithms, which are not available to date. (Image was made by Hongfeng Yu and data was provided by John Blondin at the North Carolina State University.)

## Summary

High performance computing systems running at sustained petaflop speeds are becoming increasingly available for scientists and engineers to perform simulations at the peta- and exa-scale. To possibly understand such extreme-scale simulations and extract meaning from petabytes of the simulation output, it is imperative that simulation scientists and visualization researchers begin to work closely together so a viable solution will be ready. Otherwise, much of the value of petascale and exascale simulations will go underutilized.

In situ visualization is clearly a promising solution for ultrascale simulations. While we have seen some success in realizing this solution [1,8] and also ongoing efforts to add support for in situ visualization to open source visualization toolkits such as ParaView and VisIt, further research and experimental studies are needed to derive a set of guidelines and usable visualization software components for others to adopt the in situ approach. If successful, this will lead to a new visualization and data understanding infrastructure, potentially change how scientists do their work, and accelerate the process of scientific discovery.

## References

1. H. Yu, C. Wang, R. W. Grout, J. Chen, and K.-L. Ma. A Study of In Situ *Visualization for Petascale Combustion Simulations.* Technical Report No. CSE-2009-9, Department of Computer Science, University of California at Davis, April 2009.
2. H. Yu, C. Wang, and K.-L. Ma. *Parallel Volume Rendering using 2-3 Swap Image Compositing for an Arbitrary Number of Processors.* ACM/IEEE Supercomputing 2008 Conference, November 2008.
3. H. Yu, C. Wang, and K.-L. Ma. *Parallel Hierarchical Visualization of Large 3D Time-Varying Vector Fields*. ACM/IEEE Supercomputing 2007 Conference, November 2007.
4. F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee. *The state of the art in flow visualisation: Feature extraction and tracking.* Computer Graphics Forum, Volume 22, Number 4, 2003, pp. 185–197.
5. C. Muelder and K.-L. Ma. Interactive Feature Extraction and Tracking by Utilizing Region Coherency. In Proceedings of 2009 IEEE Pacific Visualization Symposium, April 2009, pp. 17-24.
6. N. Sahasrabudhe, J. E. West, R. Machiraju, and M. Janus. *Structured spatial domain image and data comparison metrics*. In Proceedings of IEEE Visualization '99 Conference, October 1999, pp. 97–104.
7. C. Wang, H. Yu, and K.-L. Ma. *Importance-driven time-varying data visualization. IEEE Transactions on Visualization and Computer Graphics,* Volume 14, Number 6, October 2008, pp. 1547-1554.
8. T. Tu, H. Yu, L. Ramirez-Guzman, J. Bielak, O. Ghattas, K.-L. Ma, and D. O'Hallaron. *From Mesh Genertion to Scientific Visualization: An end-to-end approach to parallel supercomputing.* ACM/IEEE Supercomputing 2006 Conference, November 2006.
9. M. Glatter, J. Huang, S. Ahern, J. Daniel and A. Lu. *Visualizing Temporal Patterns in Large Multivariate Data using Textual Pattern Matching*. IEEE Transactions on Visualization and Computer Graphics, Volume 14, Number 6, October 2008, pp. 1467-1474.

**Kwan-Liu Ma** is a professor of computer science at the University of California, Davis. He directs the DOE SciDAC Institute for Ultrascale Visualization. Contact him at ma@cs.ucdavis.edu