SCaLeS ~ Science Case for Large-scale Simulation
Visualization Breakout Session

"Scalability Challenges Facing Visualization R&D"

Position Statement ~ James Kohl, Oak Ridge National Laboratory

Introduction

As scientific simulations grow increasingly large and complex, the computational resources required for their execution become ever more massive and unwieldy. Due to the immense amount of data produced by these simulations, it is crucial to have efficient data analysis and visualization capabilities for processing these gigantic data to glean scientific insight.

Unfortunately, significant technological and sociological barriers stand in the way of such scalable visualization systems. There are currently insufficient computational, network and storage resources available, let alone efficient parallel algorithmic foundations, to adequately address scalable visualization.

I/O and Data Movement

The input/output requirements of large-scale computer systems are seemingly of the lowest design priority; hence it is surprisingly difficult to get large amounts of data in and out of a big machine. This basic fact can render an extremely powerful compute engine primarily *idle*, waiting for input data or output results to be shuffled slowly to/from external storage media. For example, the Japanese Earth Simulator, for all its might and high efficiency, must periodically stop execution all together to clear data into a mass storage bottleneck.

Most massively parallel computer systems, consisting of 100s or 1000s of processing units, are designed with a limited number of network connections to the outside world. Large datasets must be funneled through these few externally visible nodes, and often must be integrated even more tightly for archiving on a few large disk or tape arrays. Even for systems with huge parallel file systems, it is likely that data must traverse several hops to get onto disk. Yet the multi-pipe parallel nature of such disk systems does not necessarily help with external transfers, unless the data is read back into the original computer system for post-processing. In fact, for many of these large systems the data analyses and visualization computation must be done *in situ* to avoid significant delays, so either way the performance of the great machine is held captive. And often the scientist must still save the full *original* data to disk for archival purposes and future analyses, thereby further complicating matters.

Dedicated Resources for Visualization

There is a discrepancy in the resources available for data analysis and visualization; an equivalent amount of infrastructure funding should be spent for "matching" analysis and rendering hardware. Rather than "playing an expensive stereo through a cheap paper cone speaker" the visualization engine must be of comparable quality to the scientific computational engine, and must be capable of handling the required data load.

Research supported by the Mathematics, Information and Computational Sciences Office, Office of Advanced Scientific Computing Research, U. S. Department of Energy, under contract No. DE-AC05-00OR22725 with UT-Battelle, LLC.

Alas, many political and sociological issues prevent this delicate balance from fruition. Shortsighted funding decisions often seek the allure of "big iron" computational systems without guaranteeing the practicality of the accompanying analysis and visualization hardware that would make these "big iron" systems actually useful to the scientist! A philosophical paradigm shift is needed to elevate visualization resources to a proper priority, both at funding agencies and from the perspective of the domain scientists and principal investigators who propose the use of large computer systems.

Interactivity vs. Batch Mode

Visualization and visual exploration of large-scale datasets is a naturally interactive venture that encourages scientists to intuitively navigate the reams of data, looking for subtle patterns and dynamic behavior. While statistical analysis techniques can identify key regions of interest in a large dataset, such approaches require good heuristics to locate the desired features. Often, scientists do not know *a priori* what it is they are looking for, and so some type of freeform exploration is ultimately necessary. It is therefore not practical to run most visualization computations in "batch" mode, where jobs are submitted and scheduled for execution at a later time. Interactive exploration requires online computational resources, that are "on call" as needed to process the data and supply it to the front-end visualization, based on the instantaneous path of traversal by the scientist. Similar to issues above on dedicated visualization hardware, computer center policies must be expanded to make allowances for this kind of interactive resource allocation and usage.

Data Storage ~ Size and Bandwidth

An obvious issue with respect to large-scale computation is that the amount of data involved is fundamentally large as well. The manner in which scientific simulations utilize and generate large datasets is based on the efficiency of the specific access patterns exhibited by the computational algorithm. Yet this data organization is rarely compatible with that needed for efficient data analysis and visualization. This data access, often a streaming of data through the analysis and rendering pipeline toward a front-end visual display device, will constitute its own unique signature of data access patterns and networking requirements, as needed to move and manipulate the data. As interactive requests from the front-end user scientists propagate back down the pipeline, a variety of bandwidth and latency requirements must be met to produce a smooth and fluid visualization. The organization of the data for this purpose must optimize search and lookup of desired data subregions, and the algorithms and storage methodologies must be designed to provide sufficiently fast access time to optimize network throughput.

Degree of Parallelism and Algorithm Scalability

Perhaps the most insidious of challenges with respect to visualizing datasets from large-scale simulations involves the scalability of the software algorithms currently applied to process and render the data. A majority of statistical data analysis algorithms are inherently serial, and many parallel rendering algorithms do not scale well above 10s of processors. Therefore, simply acquiring massive computational resources for data analysis and visualization will not in itself be sufficient for providing scalable solutions; fundamental research is required to efficiently parallelize these algorithms and develop new approaches that minimize latency and hide communication.