Petascale visual data analysis in a production computing environment

Sean Ahern

National Center for Computational Sciences, Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008 MS-6016, Oak Ridge, TN 37831, USA

E-mail: `ahern@ornl.gov`

Abstract. Supporting the visualization and analysis needs of the users of the Department of Energy's premiere high-performance computing centers requires a careful engineering of software and hardware system architectures to provide maximum capability and algorithmic breadth. Data set growth follows an inverse power law that has implications for the platforms that are deployed for analysis and visualization; central storage and coupled analysis platforms are critical for petascale post-production. Software architectures like VisIt – which exploit parallel platforms, as well as provide remote capability, extensibility, and optimization – are fruitful ground for delivering new analysis capabilities for petascale applications. Finally, direct interaction with customers is key to deploying successful results.

1. Introduction

At the Department of Energy's (DOE's) high-performance computing (HPC) centers, and within the SciDAC Visualization and Analysis Center for Enabling Technologies (VACET) in particular, we are attempting to provide data analysis solutions that go beyond traditional "visualization." Post-processing and analysis go beyond simple generation of imagery and include data exploration, visual code debugging, comparative analysis, quantitative analysis, and presentation graphics. Our goal is to deploy infrastructure for all of these uses for full analysis, code comparison, and verification and validation. To do so, we will need the assistance of application scientists, as physics experience will be required. Our codes will act as facilitators, providing parallel infrastructure, releases on many platforms, documentation, and support.

The National Center for Computational Sciences (NCCS) at Oak Ridge National Laboratory (ORNL) is home to the world's largest supercomputer for open science research, the Cray XT3/4 "Jaguar" system, with a current peak computational rate of 119 teraFLOPS and 46 TB of aggregate memory. In addition, NCCS provides the world's largest vector supercomputer, the Cray X1E "Phoenix" system, with a peak computational rate of 18.5 teraFLOPS and 1 TB of aggregate memory. Through the DOE INCITE program [1], 28 different science application projects, including notable SciDAC applications, have access to these computational systems in 2007. These projects cover a wide range of application areas including astrophysics, material science, climate dynamics, fusion, and turbulent combustion (see Figures 7, 8, and 9 at the end of this paper). Members of the NCCS visualization team, many of whom are shared with VACET, are working with application teams to provide direct visualization support.

The National Energy Research Scientific Computing (NERSC) Center, an open computing facility located at the Oakland Scientific Facility and part of Lawrence Berkeley National

Laboratory, provides compute and storage resources as well a portfolio of services that support state-of-the-art computational science research. The NERSC facilities include several computing platforms: (1) a Cray XT4 with 19,000 dual-core AMD Opteron processors and 39.5 TB of aggregate memory (Franklin); (2) an 888 CPU IBM Power5 system with approximately 3.5 TB of RAM (Bassi); (3) a 320 node, dual CPU Opteron cluster (Jacquard); (4) a 6080 CPU IBM Power3 system (Seaborg); and (5) a 32 CPU SGI Altix with 192 GB of RAM used primarily for interactive visual data analysis (DaVinci). The NERSC Analytics Team, some of whom are shared with VACET, provides solutions spanning the domains of visualization, analysis, data, and workflow management to the challenging data understanding problems of the NERSC user community. (See Figures 1 and 2.)

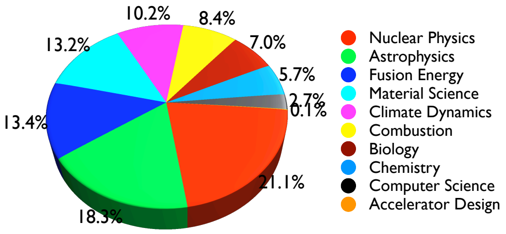


Figure 1. LBL/NERSC usage by scientific discipline in 2006. *Source: https://www.nersc.gov/about/users.php*

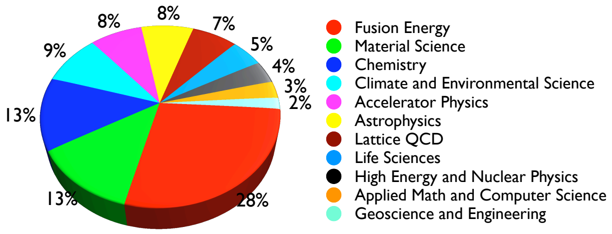


Figure 2. ORNL/NCCS usage by scientific discipline in 2007.

Each of these diverse application areas has unique data analysis requirements driven by its respective domain science, and the requirements for data analysis systems are similarly specialized. Among the significant data understanding challenges to be faced in the next 5 years are aggregating ensembles and parameter studies, understanding coupled multi-physics simulation output, and understanding simulation features that span many temporal scales. Commonality may be exploited to deliver common solutions, but direct application impact is crucial.

As mentioned above, VACET includes staff members at DOE's large open computing facilities at NERSC and NCCS. Including facility staff in VACET is critical for deploying analysis and visualization tools and resources to the users of these facilities. It is becoming increasingly important for production-quality post-processing capabilities to be located close to the data. Visualization and analysis of large-scale scientific data is a rich field of research. However, all too often, research results remain in the realm of academia and never reach the hands of the application scientists for whom the research was originally undertaken. Fully deploying research results to application scientists is a complex process that requires careful attention to data set characteristics, user interfaces, system engineering, storage systems, institutional architectures, remote image delivery, and user support infrastructures. As data set sizes increase to the petascale in the near term and to the exascale in the next 10-15 years, it will become increasingly important to develop critical analysis capabilities through research and deliver them to end users to help them understand their data.

2. Deployment to end users

Successful deployment of the infrastructure architectures discussed in Section 5 requires both full access to the compute platforms and direct interaction with end users. Members of VACET have an ERCAP allocation at NERSC for both cycles and storage. At ORNL, VACET members and the SciDAC Institute for UltraScale Visualization (IUSV) members have a Director's

Discretionary INCITE allocation for cycles and storage. This grants us the access we need to have a direct impact upon SciDAC customers.

In addition, we have been the most successful in our efforts when we have been invited to embed visualization team members within application groups themselves. Notably, this approach has enabled us to achieve demonstrable success with the SciDAC Terascale Supernova Initiative and ORNL's climate dynamics group. This close relationship between visualization specialists and domain experts provides the best route to deploying successful analysis capabilities to those actually performing the science. We expect to replicate these successes at DOE's premiere HPC centers. This strategy often requires co-location and can be manpower-intensive.

The visualization and analysis teams at NCCS and at NERSC are well established and have ongoing customer collaborations that have been fruitful. The combination of established institutional support structures at both laboratories and the resources provided by VACET are a pattern for success in delivering capability to the users of the DOE HPC centers.

3. Collaborations

In addition to the strong deployment base provided by NCCS and NERSC, VACET is partnering with many other collaborators to help deliver successful solutions. No analysis and visualization deployment effort is likely to be successful in delivering capability to all users without cross-disciplinary collaboration. VACET members are connected to many visualization research efforts, including notable efforts at Lawrence Livermore National Laboratory, UC Davis, the University of Tennessee, and the University of Utah. In addition, VACET is partnering with the IUSV to assist in the deployment of novel visualization research.

One ongoing effort is the integration of the IUSV's multidimensional parallel indexing method [2] with the VisIt visualization system. This will provide users with the ability to do rapid index queries on petascale data sets from their desktop systems. In a similar vein, VACET is partnering with the SciDAC Scientific Data Management (SDM) Center to help users improve the I/O performance of their codes and develop community-centric data models and formats. Another joint VACET/SDM effort aims to integrate bitmap index queries into VisIt, providing the capabilities of FastBit for large query-driven visualization.

4. Data set growth and analysis platforms

As the maximum data set size grows toward the petascale and the exascale, the range of data set sizes also continues to grow. For example, current fusion data sets range from hundreds of terabytes down to a few megabytes. Climate data sets often have similar ranges, depending on the scope of the problem. Gordon Bell demonstrates [3] that data set sizes of scientific computational problems obey an inverse power law whereby the data set size and the number of such data sets are inversely proportional – there are a small number of huge data sets and a huge number of small data sets (see Figure 3). As has always been the case, the largest "hero" size data sets are the least frequently generated and the most difficult to work with. On the other end of the scale, many scientific data sets are small enough to be processed on local desktops. The challenge is in providing analysis capabilities that span the entire size range.

Because of the difficulties in moving and processing extremely large data sets, the options for analysis decrease as the size increases. For small data sets that can fit on local desktops, the range of analysis techniques is large. Given a modern desktop with 4 GB of memory, a 3-dimensional uniform rectilinear mesh with 5 variables (8 bytes per variable) can generally reach spatial resolutions of approximately 420^3 grid cells if "normal" analyses such as extracting isosurfaces and volumetric rendering are undertaken. As data set sizes grow beyond that limit, or if more complex analysis capabilities (temporal) are needed, other computational platforms are required. In large data post-processing, the limiting factor is no longer processor speed but instead aggregate I/O bandwidth and total memory footprint.

The next step up the chain in platforms is the medium-to-large symmetric multiprocessor (SMP), one that provides a scalable resource of up to a few terabytes of memory and I/O capability and processors to match. Such platforms have proven very useful at NERSC and NCCS and are expected to continue their utility into the petascale and exascale eras.

Once data set sizes become large enough to outstrip the capabilities provided by large SMPs, the only feasible solution currently available is employing distributed parallelism. This technique has been used by only a few visualization and analysis tools (e.g., VisIt [4], ParaView [5], EnSight [6]) because of the complexities of providing a complete data-parallel solution. However, the technique has proved capable of providing the highest degree of scalability to date with large scientific data sets. Childs et al. [7] have demonstrated using VisIt in parallel to produce visualizations for Rayleigh-Taylor instability problems as large as 27 billion computational elements.

The final rung on the data processing ladder is taking advantage of the original computational platform itself. As data sets reach petabyte and exabyte scales, the cost of providing a separate visualization and analysis cluster or SMP will become prohibitively high. Current price estimates for petascale analysis clusters are in the $10-30 million range. Consequently, analysis must move "closer" to the data source, the original computational platform upon which the scientific simulation code runs. This presents many challenges: coupling directly to the simulation code, sharing computation and I/O time with the simulation code, and the restrictions of limited execution kernels on the largest computational platforms (IBM's Blue Gene [8], Cray XT Catamount [9]). See Section 7 for more discussion of large computational platforms.

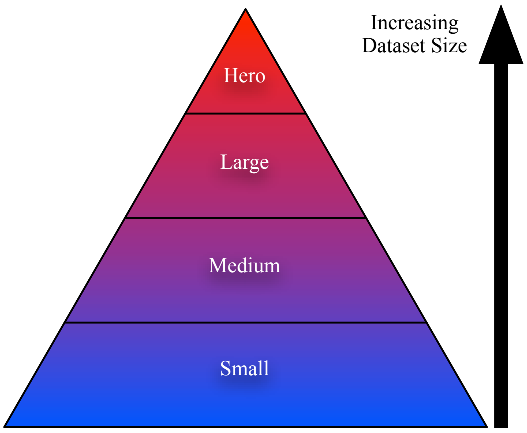


Figure 3. The relationship between data set size and frequency. "Hero" size runs occur less frequently than small data sets.

5. Institutional infrastructures and remote visualization

The thresholds that Bell outlines [3] show the need for co-locating analysis capability with the generated data once the petaFLOPS level is reached. These thresholds, in addition to the platforms in Section 4, argue for an institutional approach to delivering analysis capability. Requiring users to move all data to local sites for analysis places two undue burdens upon them: the time and space cost of storing the generated data, and the monetary cost of purchasing and maintaining analysis systems. The centrally located analysis systems we have deployed at NCCS and NERSC provide a level of analysis capability that caters to the remote nature of our customers.

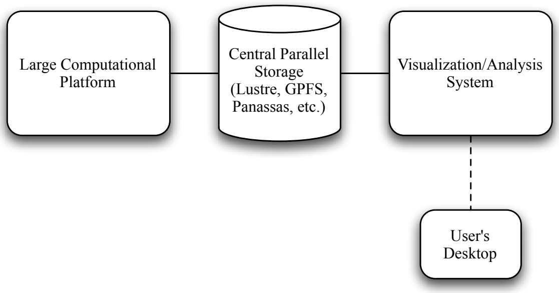

Large Computational Platform
Central Parallel Storage (Lustre, GPFS, Panassas, etc.)
Visualization/Analysis System
User's Desktop

Figure 4. Institutional infrastructure for scalable remote visualization.

5.1. Central parallel I/O system

One key element of successful institutional infrastructures for visualization and analysis is the shared parallel file system. Such file systems as CFS Lustre, Panassas, and IBM GPFS [10] have proved themselves to be scalable solutions for the large-scale I/O needs of the computational systems. In a similar way, the I/O needs of analysis and visualization scale with the I/O needs of the simulations. Architecting a central parallel file store that shares bandwidth with the analysis systems provides a "no copy" solution for large-scale data (see Figure 4). Simulation results may be left in place on the main parallel file store and directly accessed from the analysis systems. The I/O parallelism enjoyed by the computational system may be similarly employed for visualization and analysis.

5.2. Remote visualization

The final step to providing remote capability is the separation of the data processing from final rendering. Two major methods have been employed in production visualization tools, both with a moderate level of success: remote geometry delivery and remote image delivery.

5.2.1. Remote geometry delivery. Several visualization applications (VisIt, ParaView, EnSight) employ a client/server architecture (see Figure 5) that separates data processing from the final destination by transmitting visualization geometry over the network for display on a local workstation. When geometry size does not saturate network bandwidth, this architecture allows the user to interact with a visualization at high frame rates, dependent only upon the capabilities of the local graphics card. Since many visualizations can distill the simulation data down to a relatively small number of geometric primitives, this framework can scale for certain generated data sets. However, as this method is very dependent upon a reasonable network bandwidth to transmit the generated geometry, users must be extremely careful not to saturate the network pathways.

5.2.2. Remote image delivery. This same client/server architecture may be employed to stream imagery to the user as well. Once generated geometry sizes become too large to send wholesale over the network, it becomes more efficient to render the visualization using the remote parallel visualization/analysis system and send the resulting image across the network to the user. The cost of sending image data is independent of the size of the input geometry and remains fixed for a given resolution and depth. While this method entails greater latency, especially for long networks, it provides a scalable solution for the largest data set sizes because client-side performance is a function of available network bandwidth. In addition, it allows the deployment of centralized resources for parallel rendering and compositing of imagery.

More advanced infrastructures such as ICE-T [11] and CEI's Distributed Rendering architectures take this yet another step further, decoupling data processing from parallel image rendering and compositing. The highest frame rates to date have been delivered through architectures such as these. Display-based systems such as Chromium RenderServer [12] and VirtualGL [13] provide remote visualization capabilities by extending the display system itself over the network. The major advantage that these systems have over the client/server architectures described is that they do not require modifications to the visualization application. Similar in concept to VNC [14], they provide architectures that allow centralized systems to provide

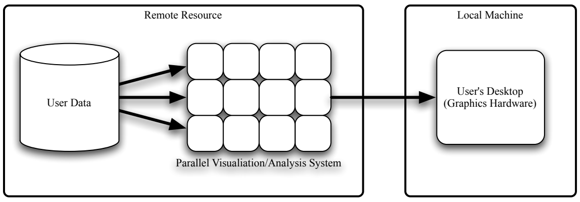


Figure 5. Scalable client/server visualization architecture.

entire X11/OpenGL rendering systems to be deployed to remote users. Chromium Render-Server even allows for a degree of parallelism to be employed, providing delivery to remote tiled displays.

6. VisIt architecture

One of the primary software systems we have chosen for application delivery in VACET, particularly at NCCS and NERSC, is the VisIt visualization and analysis package. This system holds much promise for the deployment of production-ready parallel visualization and analysis capability at the petascale for SciDAC customers. In this section, we outline some of the architectural decisions that have enabled the successes of this system for current production HPC analysis needs.

6.1. Cross-platform deployment

VisIt was architected with maximum portability in mind. The client runs on all major flavors of Microsoft Windows, Apple's Mac OS X (both PPC and Intel), and 32- and 64-bit Linux. The data processing engine runs on all of these, as well as achieving scalability on every major parallel computational platform deployed, including Cray's Catamount operating system. Dependencies upon VTK, Qt, and POSIX ensure consistent behavior across platforms. In addition, VisIt has native support for network tunneling and submission to HPC system batch queues, providing simple deployment in production parallel environments. All of these allow easy deployment of VisIt across all ranges of platforms and data set sizes generated by SciDAC customers.

6.2. Contract-based data pipelines

The concept of pipeline-based data flow networks is very common in the field of visual data analysis (OpenDX, VTK, AVS/Express, etc.). In the pipeline model, independent data manipulation operations are chained together to extract data of relevance for the user. This model allows extremely flexible analysis routines to be designed on the fly for processing scientific data. In [15], Childs et al. extend the traditional data flow network with the notion of a contract, a form of intercommunication between the independent data manipulation operators. This intercommunication allows for performing a high degree of optimization on the network, such as I/O optimization, ghost data computation, and rectilinear subgrid generation. The contracts are extendible, allowing deployment of other optimizations for particular pipeline configurations. As the data sets generated by SciDAC customers vary greatly in size and character, this extendible optimization system provides for a relatively simple method for prototyping and deploying new optimization techniques to take best advantage of I/O architectures.

6.3. Plug-in architecture

The bulk of VisIt's functionality is delivered in the form of plug-ins, code modules that extend VisIt's data processing capability in directions unanticipated in the original design. There are three areas of functionality that may be extended using plug-ins: databases, operators, and plots. The native capability to extend the system by way of the plug-in interface provides an extremely rich infrastructure for deployment of custom capability for SciDAC customers, as new capability may be prototyped and deployed to individual customers without requiring wholesale code modification. In addition, successful plug-ins may easily be incorporated into the general code base, providing capability to the entire VisIt community.

6.3.1. Databases. VisIt may be easily extended to read new kinds of data. The internal data model of the system is very rich; it encompasses many different mesh types and representations, including particles, unstructured meshes, and AMR hierarchies; scalar, vector, and tensor variables; subgrid volumetric material information; and mass fraction-based species. The work of reading in the data generated by a given simulation code consists of writing the mapping between the simulation's data model and VisIt's internal data model. More than 50 database readers have been deployed to date, and writing new readers is a relatively simple operation, depending on how complex the mapping is. Some database readers can be deployed in as little as 30 minutes.

One of the first software efforts undertaken by VACET was the development of custom database readers for various SciDAC partners that fully expose the underlying data model. This has provided a much greater production data analysis capability than has been seen before in SciDAC applications.

6.3.2. Operators. VisIt's fundamental method for transforming data is an operator. These code modules convert the data in some fashion, such as transforming between coordinate spaces, extracting slices and isosurfaces, thresholding data, or extruding new meshes. Similar to databases, operators may be extended through the plug-in model to provide new data transformation capabilities. For domain-specific analysis, such as magnetic field line tracking in fusion simulation codes, VACET members have been working to deploy operators that encompass the algorithms that directly impact simulation efforts.

6.3.3. Plots. The method of displaying data to the screen in the form of polygons and images, as well as the interaction behavior of the display, is embodied in VisIt through a plot. These encompass such features as pseudocolor displays, streamline generation, parallel coordinates, and simple mesh displays. Plots may be extended through the plug-in mechanism as well.

6.4. Derived quantities and expressions

VisIt contains a rich mathematical data manipulation language that allows scientists to derive new quantities in arbitrary ways from variables stored in the simulation data. A major goal of the expression language is to provide a facility in which creation of new derived quantities is so intuitive that there is little to no learning curve for the most common operations. For example, if users want to plot the logarithm of density, or the average of two pressure fields, they create simple expressions like $log(density)$ or $(pressure_a + pressure_b)/2$. Support exists in the language for manipulating scalars, vectors, tensors, strings, lists, meshes, materials, arrays, and other subsets of elements. For accessing other files, either within the same time sequence or in different sequences, the language supports references by cycle, absolute and relative time index, and filename. Accessing other files has been found to be extremely useful in mapping data from one mesh onto an entirely different mesh, possibly generated by a separate simulation. The language may be extended to provide new capability and mathematical operations.

6.5. Queries and quantitative analysis

VisIt data analysis capabilities are not restricted to deriving new mesh variables through expressions. VisIt also contains a rich library of analysis capabilities called “queries” which allow the user to perform arbitrary analyses, including weighted sums, integrations, shape characterizations, and calculations of moments. In addition, we have added statistical analysis capability in the form of equivalence class functions to facilitiate analysis of parameter studies, variable binning, and uncertainty characterization. Delivered in a production visualization and analysis tool, these capabilities are well-poised to provide useful scientific analyses for large-scale simulation efforts within the SciDAC community.

7. Future work

Ideally, simulation codes would be able to output all interesting data at every time step. Such a data resource would allow deployment of a rich set of analysis algorithms on a co-located analysis machine. However, the limitations of I/O bandwidth and storage space are prohibitive, and many simulation codes output only a small subset of the data of interest, sometimes as small as 0.1%. Thus it is important for future analysis capabilities to move all the way to the source of the data, the computational platform and the simulation itself. Delivering analysis capability in the form of parallel libraries that can be leveraged by the application codes directly is an active area of research.

One institutional architecture that holds particular promise is the coupling of three separate systems: the computational system, a large analysis system with rendering capabilities, and the user's desktop (see Figure 6). Such a three-tiered arrangement would deploy the full range of analysis capabilities while still optimizing I/O bandwidth and distributed resources. In this model, the simulation code would provide raw simulation data to a processing library that would do an initial analysis (e.g., probability density function generation, cluster analysis, data reordering). That library would stream data in parallel to an analysis cluster for processing and geometry generation. The geometry would be rendered (and optionally composited) in parallel and then delivered to a user's desktop, possibly in a remote location.

One primary challenge in delivering such an ambitious system is the exploitation of the restricted execution model of current HPC compute platforms. Operating systems such as Catamount and that deployed on IBM BlueGene systems pose significant challenges for analysis models based on plug-ins, sockets, and threads. Fortunately, progress is being made on Cray systems by way of library and socket emulation layers and production deployment of Compute Node Linux.

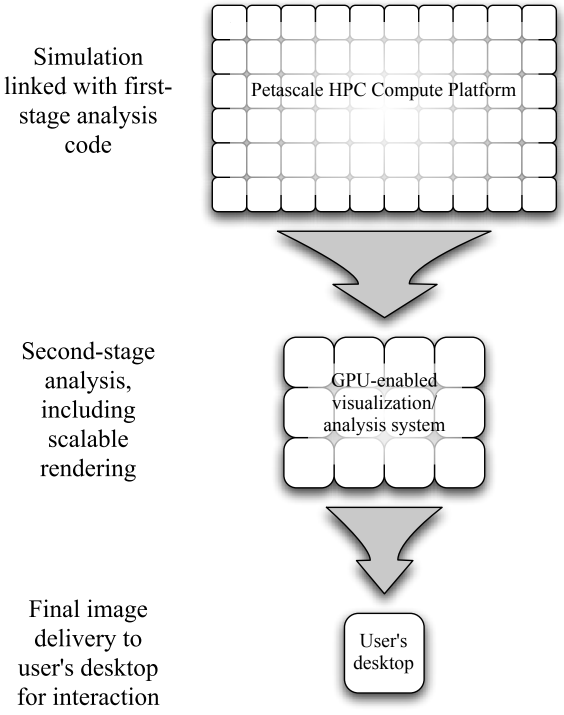

Simulation linked with first-stage analysis code
Petascale HPC Compute Platform
Second-stage analysis, including scalable rendering
GPU-enabled visualization/ analysis system
Final image delivery to user's desktop for interaction
User's desktop

Figure 6. A three-stage architecture for delivering interactive visualization and analysis capability at the extreme end of the data set size.

Finally, the complexity of managing the submission of large compute tasks, marshalling of data, backup to archival storage, and offline generation of analysis results is becoming sufficiently large that automated methods for managing the workflow are becoming necessary. Similarly, the use of higher-level management utilities can facilitate the management of complex data flow networks for post-processing. In this vein, tools such as Kepler [16] and VisTrails [17] hold the promise of providing "end-to-end" services for the users while still allowing fine-grained control where necessary.

8. Conclusion

The DOE HPC centers, augmented with the capabilities created by SciDAC's Visualization Center and Institute, have been able to deploy terascale data analysis to current scientific application users. However, the challenges posed by multiphysics codes and by increases in data set sizes with the advent of petascale computing will challenge the existing methods for deployment and customer delivery. With the adoption of coupled hardware systems, along with scalable infrastructures for data parallel visualization and appropriate management tools, petascale data analysis and visualization is within reach.

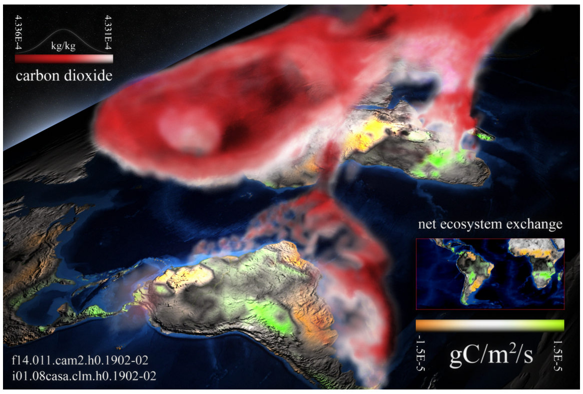


Figure 7. Coupled climate simulation, detailing net ecosystem exchange and elements of the carbon cycle within the CCSM climate simulation code.

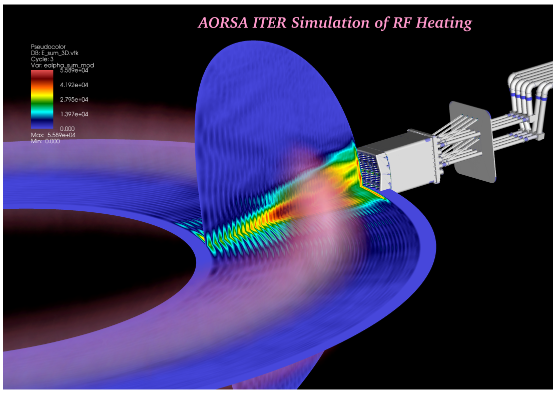
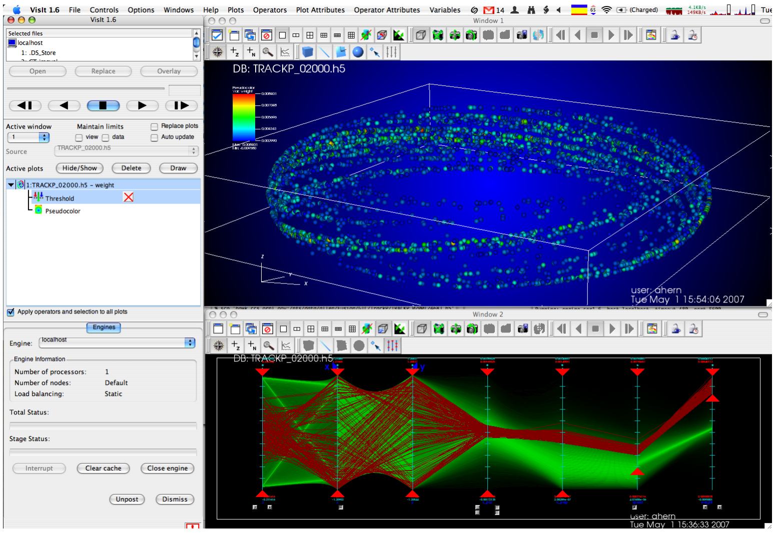


Figure 8. Simulation of ITER fusion reactor with radio frequency antenna providing heating to tokamak plasma from the AORSA simulation code.



Active window: 1
Maintain limits: □ view □ data
□ Replace plots
□ Auto update
Source: TRACKP_02000.h5
Active plots:
1:TRACKP_02000.h5 - weight
Threshold
Pseudocolor
☑ Apply operators and selection to all plots
Engine: localhost
Engine Information
Number of processors: 1
Number of nodes: Default
Load balancing: Static
Total Status:
Stage Status:

Figure 9. Particle extraction from the GTC fusion simulation code by way of parallel coordinates and multidimensional filtering deployed in the VisIt visualization system.

References

[1] DOE INCITE Program web site: http://hpc.science.doe.gov/

[2] Glatter M, Mollenhour C, Huang J and Gao J, 2006, Scalable Data Servers for Large Multivariate Volume Visualization, *IEEE Transactions on Visualization and Computer Graphics*, **Vol. 12, No. 5**, pp. 1291–1299

[3] Bell G, Gray J and Szalay A, 2006, Petascale computational systems, *IEEE Computer*, **39(1)**:pp. 110–112.

[4] VisIt web site: http://www.llnl.gov/visit

[5] ParaView web site: http://www.paraview.org/

[6] CEI EnSight web site: http://www.ensight.com/ensight-gold.html

[7] Childs H, Duchaineau M and Ma K-L, 2006, A Scalable, Hybrid Scheme for Volume Rendering Massive Data Sets, *Eurographics Symp. on Parallel Graphics and Visualization*

[8] Almási G, Bellofatto R, Brunheroto J, Cascaval C, Castaños J, Ceze L, Crumley P, Erway C, Gagliano J, Lieber D, Martorell X, Moreira J, Sanomiya A and Strauss K, 2003, An overview of the Blue Gene/L system software organization, *Euro-Par*, pp. 543–555

[9] Kelly S and Brightwell R, May 2005, Software architecture of the light weight kernel, Catamount. *Cray User Group*, Albuquerque, NM.

[10] Schmuck F and Haskin R, 2002, GPFS: A Shared-Disk File System for Large Computing Clusters, *In Proc. of the First Conference on File and Storage Technologies (FAST)*

[11] Moreland K, Wylie B and Pavlakos C, 2001, Sort-last parallel rendering for viewing extremely large data sets on tile displays. *In Proc. of the IEEE 2001 Symp. on Parallel and Large-Data Visualization and Graphics*, pp. 85–92.

[12] Chromium Renderserver Overview: http://vncproxy.sourceforge.net/

[13] VirtualGL web site: http://www.virtualgl.org/

[14] Virtual Network Computing description page: http://en.wikipedia.org/wiki/VNC

[15] Childs H, Brugger E, Bonnell K, Meredith J, Miller M, Whitlock and Max N, 2005, A Contract-Based System for Large Data Visualization, *In Proc. IEEE Visualization 2005*, pp. 190–198

[16] Kepler web site: http://kepler-project.org/

[17] VisTrails web site: http://www.sci.utah.edu/stories/2006/vistrails.html