The Scalable Data Management, Analysis, and Visualization Institute http://sdav-scidac.org
Visualization Technologies and Applications

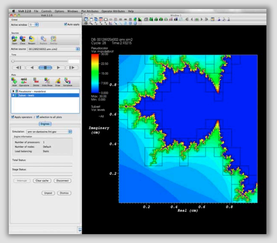
Production *In Situ* Software Infrastructure

VisIt libsim

Library interface to VisIt's compute engine, providing simulations with access to a full suite of visualization and analysis capabilities.

ParaView Catalyst

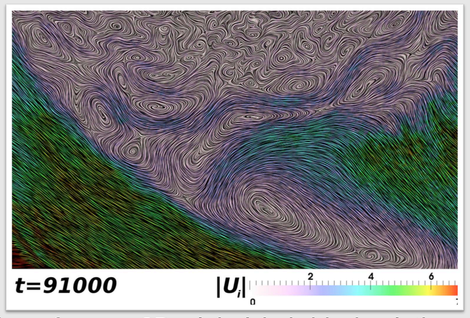
A simplified library providing scalable visualization resources, having a minimal impact on the simulation.



Application: *In Situ* and Space Weather Modeling

H. Karimabadi (UCSD), P. Oleary, B. Geveki (Kitware), B. Loring (LBNL), A. Majumdar, M. Tatineni (SDSC).

In situ visualization for global hybrid (electron fluid, kinetic ions) simulations used to study the interaction of the solar wind with planetary magnetospheres such as the Earth and Mercury. ParaView Catalyst is designed to be tightly coupled with simulation codes, and was directly embedded into UH3D to perform scalable in-situ analysis at run time. The adaptor code translates the internal UH3D data structures to the VTK- based data structures. This translation was performed using zero-copy mechanisms, something that the existing I/O capabilities of UH3D fail to achieve.



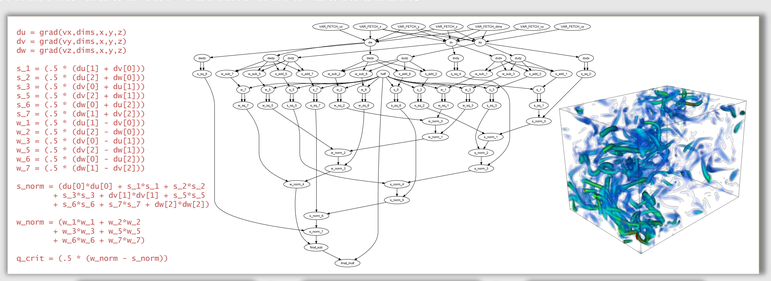
(Left) Temperature isosurface from a 3D global hybrid simulation of the Earth's magnetosphere. (Right) Line-integral convolution showing a zoomed-in view of the flow field, generated with a new parallel- and multi-block capable LIC algorithm.

H. Karimabadi, P. O'Leary, M. Tatineni, B. Loring, A. Majumdar, and B. Geveci, "In-situ Visualization for Global Hybrid Simulations," XSEDE '13 Proceedings (to appear).

OpenCL Derived Quantity Framework

E. Brugger (LLNL)

We developed a Python-based OpenCL framework that supports multiple *Execution Strategies* to implement processing and execution of dynamic user expressions. An expression, defining a new mesh field, is parsed at runtime and translated into a data flow network specification, which in turn is used to dynamically generate a single OpenCL kernel for the operation. This illustration shows the OpenCL GPU expression framework with a user-defined Q-criterion expression, which is used for vortex core detection.



Energy Efficiency and Urban Light Modeling

G. Weber, H. Childs, P. Colella, G. Graves, H. Johansen, T. Ligocki, D. Martin. B. van Straalen (LBNL), E. Brugger (LLNL)

Problem:

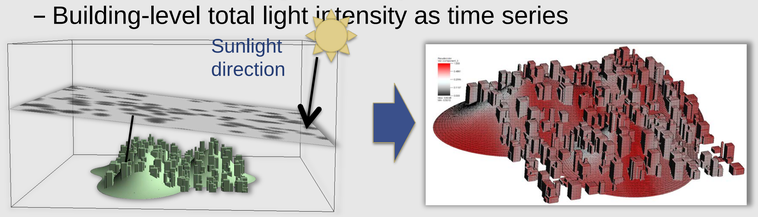
- Optimize energy consumption of urban area over time
- Calculate light intensity derived from time-series of opacity, direction
- To be used for high-resolution (space, time) solar radiation calculation

Overall Approach:

- Combine a CFD simulation of moving "clouds" over complex geometry
- Light angle, building shadows, opaque cloud layer all interact
- Use visual data analysis (VisIt) to compute light intensity transfer through clouds and map it to individual buildings
- Use results as part of time series model for building energy simulation

Current Work:

- End-to-end "proof of concept" implementation of all parts of pipleline
  - Chombo urban region example model
  - VisIt for computing light intensity transfer based on direction
  - Building-level total light intensity as time series



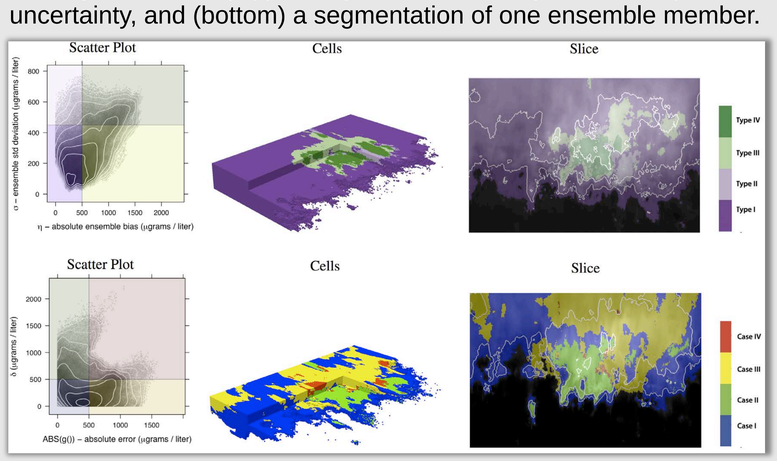
Characterizing and Visualizing Predictive Uncertainty through Bayesian Model Averaging

K. Bensema, H. Obermaier, K. Joy (UCD), H. Childs (LBNL/UO), L. Gosink, T. Pulsipher, M. Henry (PNNL)

We have developed new approaches to analyze the predictive uncertainty in simulation ensemble data based on ground truth data. Here, we characterize predictive uncertainty from contaminant plume model outputs, which consists of 12 simulations where model parameters vary to explore different geologic scenarios.

Our strategy employs Bayesian Model Averaging (BMA) to first construct a statistical aggregate from the ensemble, producing a continuous ground truth model from the given (sparse) data. With this BMA aggregate, we are able to analyze predictive uncertainty anywhere in the domain, following our novel two-step visual analysis process. This approach provides the means for robust visual analysis of predictive uncertainty, and to quanitify the contribution of individual members to the overall predictive accuracy.

The figure shows (top) a segmentation of high and low predictive uncertainty, and (bottom) a segmentation of one ensemble member.

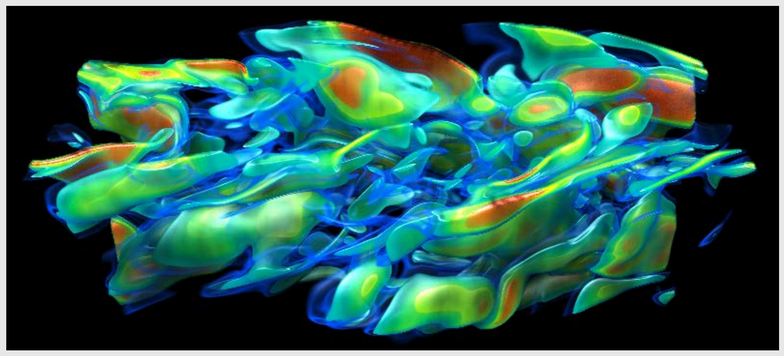


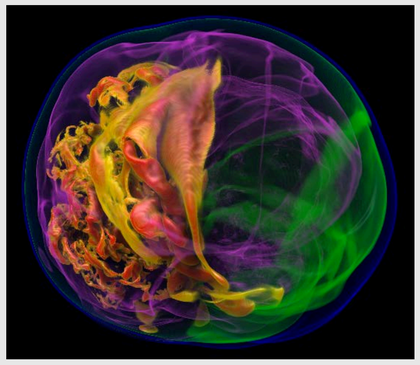
Luke Gosink, Kevin Bensema, Trenton Pulsipher, Harald Obermaier, Michael Henry, Hank Childs, Kenneth Joy. Characterizing and Visualizing Predictive Uncertainty in Numerical Ensembles Through Bayesian Model Averaging IEEE TVCG, Proc. of IEEE Vis2013.

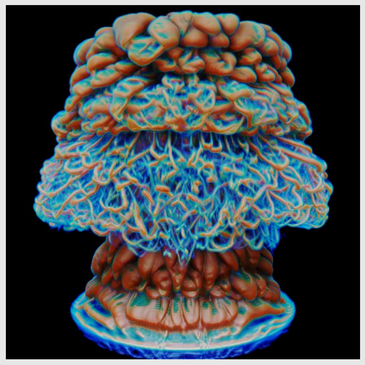
Advanced Illumination for Visualizing 3D Flow and Structures

Y. Zhang (UCD) and K.-L. Ma (UCD)

- Interactive visualization incorporating global illumination
- Enhancing the perception of the depth, thickness, and spatial structure of flow features
- High quality and GPU accelerated
- Automatic lighting design for volume data visualization







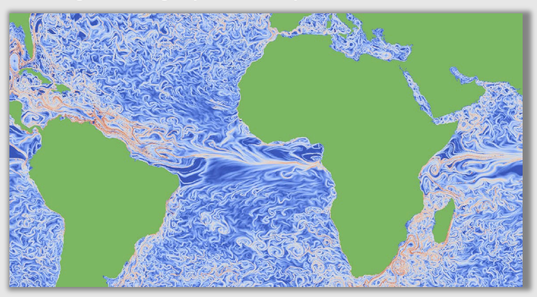
Y. Zhang and K.-L. Ma. *Fast Global Illumination for Interactive Volume Visualization*. In Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D '13), pp. 55-62.

Y. Zhang and K.-L. Ma. Lighting Design for Globally Illuminated Volume Rendering. IEEE TVCG (also Proceedings of SciVis 2013).

Scalable FTLE Computation for Time-Varying Vector Fields

B. Nouanesengsy, T.-Y. Lee, K. Lu, H.-W. Shen (OSU), T. Peterka (ANL)

We developed a scalable FTLE computation algorithm using a pipelining model. In the algorithm, we break the compute processes into different time groups, where each group independently processes different time interval. Particles are passed between different time groups as they travel in space and time to achieve computation efficiency and reduce I/O overhead. Our algorithm is able to demonstrate strong scaling up to 16K processors.



FTLE of an ocean data set output from an eddy resolving simulation. Each time step of the data has a resolution of 3600 x 2400 x 40. The data set was provided by Mathew Maltrud of Los Alamos National Laboratory.

B. Nouanesengsy, T.-Y. Lee, K. Lu, H.-W. Shen, and T. Peterka. Parallel particle advection and FTLE computation for time-varying flow fields. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC '12). IEEE Computer Society Press, Los Alamitos, CA, USA.

SDAV
Scalable Data Management, Analysis, and Visualization

The Scalable Data Management, Analysis, and Visualization Institute http://sdav-scidac.org
Visualization Technologies and Applications

Delivering Visual Data Analysis Technology to the SciDAC Community

Our goal is two-fold: to provide technical solutions in data management, analysis, and visualization that are broadly used by the computational science community, and to actively work with science teams to help them apply these technologies to achieve scientific knowledge discovery.

Our approach is to develop and deploy software infrastructure at major DOE supercomputing centers and partner with science teams to help them achieve their objectives.

This poster contains information about the major themes in the SDAV Visualization Focus areas:

- Software tools: applications (VisIt, ParaView) and libraries (DIY)
- Evolving these tools to a changing computational landscape in the near term (multi-threading, hybrid parallelism) and longer term (Dax, EAVL, PISTON).
- Select vignettes of science applications and ongoing R&D projects.

VisIt

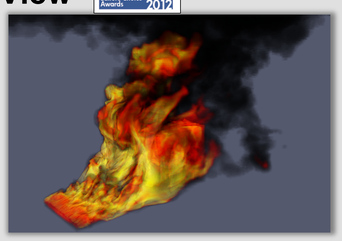
The VisIt project has dual foci on (1) visualizing and analyzing the world's largest data sets and (2) providing a robust product to end users. See http://www.llnl.gov/visit.

SciDAC Review covers with VisIt visualizations

ParaView

HPC Wire Editors' Choice Awards 2012

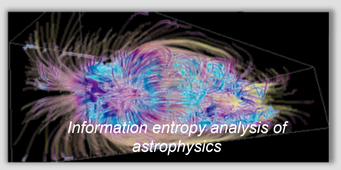
ParaView is an open-source, multi-platform data analysis and visualization application. It was designed from the ground-up to analyze extremely large datasets using distributed memory computing resources. http://www.paraview.org/.



Unstructured volume rendering of the output from a fire simulation performed at Sandia National Labs.

DIY

DIY is a library that provides visualization and analysis capabilities for *in situ*, co-processing or post-processing use on modern HPC platforms. It uses a form of MPI-hybrid parallelism, and is for scientists, vis researchers, or tool builders. http://mcs.anl.gov/project/diy-do-it-yourself-analysis/.



James Ahrens, Chris Sewell, John Patchett/LANL, E. Wes Bethel, Hank Childs, Gunther Weber, Hari Krishnan, Oliver Ruebel/LBNL, Eric Brugger/LLNL, Berk Geveci/Kitware, Kwan-Liu Ma, Ken Joy, Harald Obermaier/UC Davis, Ken Moreland/SNL- NM, Valerio Pascucci, Kristi Potter, Chuck Hansen, Allen Sanderson/Utah, Han-Wei Shen/Ohio State, Tom Peterka/ANL

Evolving the Parallelism in VisIt and ParaView

One of SDAV's objectives it to ensure that core visualization and analysis technologies operate well on major SC platforms over the course of the five-year lifetime of the project. To that end, we are undertaking a bifurcated strategy with short- and long-term objectives.

In the short term, efforts are underway for both VisIt and ParaView to follow a quick path towards hybrid parallelism, and thus be able to leverage shared-memory parallelism on multi-core platforms that complements their existing distributed-memory parallel architectures. Those efforts are summarized below.

In the long term, our team includes three projects (Dax, EAVL, PISTON), which aim to provide the means for robustly mapping algorithms onto either multi-core CPU or many-core GPU platforms.

Transitioning VisIt to Multi-core Hybrid Parallelism

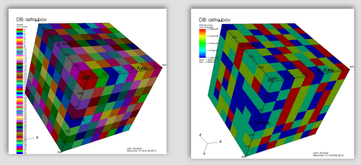
D. Camp (LBNL), H. Childs (LBNL/UO)

Our approach is to add data-parallel execution to VisIt through the use of POSIX threads. We added multi-threading capability to VisIt by:

- A new execution manager, which maps work to resources.
- Adding code to key base classes to engage the execution manager to round-robin data *pieces* (subsets) over threads.
- Updating code that was previously thread-unsafe, and adding new elements to VisIt's regression tests to report on thread unsafety
- Updating internal infrastructure for timing measurements.

Results of a recent study, submitted for publication, show this new infrastructure runs more quickly and uses less many than a purely MPI-only based approach. This work, which is a Year 1 SDAV deliverable, is merged into the VisIt trunk, and will be released to the community in version 2.7.0 of VisIt.

The images below show a 512-block astrophysics dataset colored by block ID (left) and by thread ID (right).



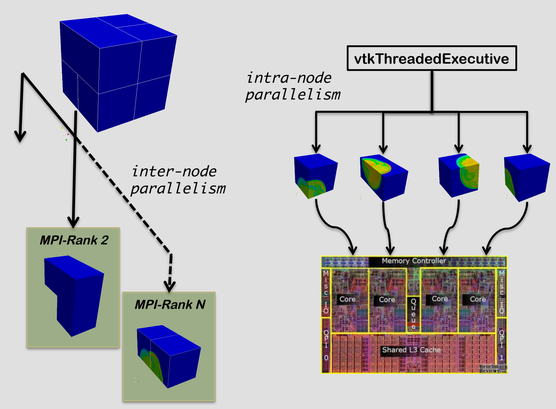
Transitioning ParaView to Multi-core Hybrid Parallelism

B. Geveci (Kitware)

Transitioning ParaView to take advantage of multi-core nodes is following examining parallelism in two paths, coarse- and fine-grained parallelism.

Coarse-grained parallelism leverages domain decomposition by processing partitions in parallel. This approach is a natural extension of distributed memory parallelism. Many algorithms require no logic change, but only need to become thread safe. Fine-grained parallelism parallelizes over cells or vertices. It works for all data types, and does not require domain decomposition within an MPI rank, but does require logic changes to algorithms.

We are presently working: infrastructure for developing parallel infrastructure and algorithms in VTK using Intel TBB and Inria KAAPI libraries; improvements to the VTK API and algorithms to promote thread-safe code development.

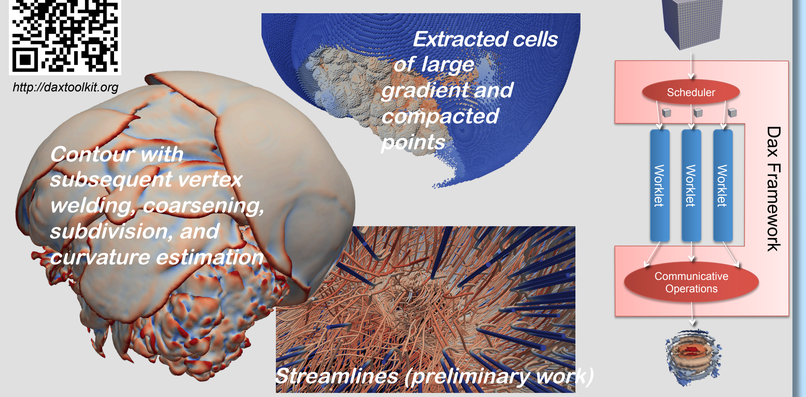


Dax: A Toolkit for Analysis and Visualization at Extreme Scale

K. Moreland (SNL-NM)

- The primitives necessary to design finely-threaded algorithms
- "Worklets" ease design in serial, scheduled in parallel
- Basic visualization design objects (think VTK for many-core)
- Communicative operations provide neighborhood-wide operations without exposing read/write hazards.

http://daxtoolkit.org



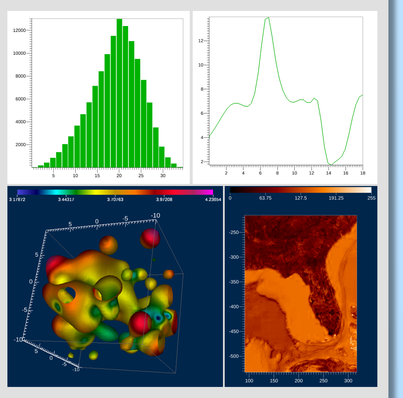
EAVL: Extreme-scale Analysis and Visualization Library

J. Meredith (ORNL)

EAVL is a library with infrastructure and algorithms designed to facilitate analysis and visualization algorithms efficient use of advanced computational architectures. See http://ft.ornl.gov/eavl.

Targets approaching hardware/ software ecosystem:

- Update the traditional data model to handle modern simulation codes and a wider range of data.
- Investigate how an updated data and execution model can improve computational, I/O, and memory efficiency.
- Enable algorithm developers to achieve these efficiency gains and better support exascale architectures.
- Explore means for simulation applications to enable *in situ* analysis and visualization in coming HPC environments.

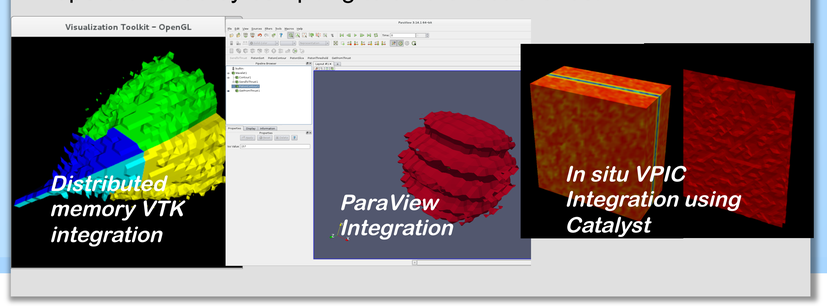


PISTON: A Portable Data-Parallel Framework

C. Sewell, L.-T. Lo, J. Ahrens (LANL)

Goal: Portability and performance for visualization and analysis operators on current and next-generation parallel architectures

- Main idea: Write operators using only data-parallel primitives (scan, reduce, transform, etc.).
- Requires architecture-specific optimizations for only for the small set of primitives
- PISTON is built on top of NVIDIA's Thrust library.
- We can run algorithms on different multi and many core architectures (e.g, GPUs, multi-core CPUs) using the exact same operator code by compiling to different backends.



SDAV
Scalable Data Management, Analysis, and Visualization