SDAV Technologies for Scientific Visualization

http://sdav-scidac.org

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

A major challenge faced by scientists today is the increasing rate of data generation by simulations and experimental devices. In this poster, we describe how SDAV tools, libraries, and algorithms can scalably and efficiently process these massive results. An additional major challenge is that the complexity of data generated from multi-parameter, multi-physics, multi-scale simulations are also increasing at a significant rate. Features in these datasets often manifest in a wide range of spatial and temporal scales, and a complete development of a phenomenon often traverses these scales. We highlight the need for exploratory visualization techniques that support understanding ensembles of results, methods for quantifying uncertainty, and tools for visually exploring and understanding features.

Delivering to the SciDAC Community with Production Visualization Tools

The goal of SDAV is twofold: to actively work with application teams to assist them in achieving break-through science; and to provide technical solutions in the data management, analysis, and visualization regimes that are broadly used by the computational science community.

- These tools are installed at the LCFs and we are actively engaging with application teams to address problems of interests

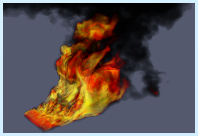
VisIt

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.

SciDAC Review covers with VisIt visualizations

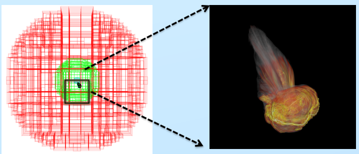
ParaView

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.



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

AMR Visualization

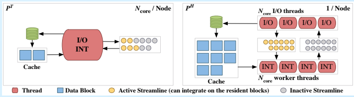


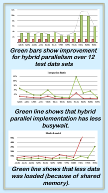
Many simulation codes, such this astrophysics example, are adopting adaptive mesh refinement (AMR) to locally increase domain resolution where needed to accommodate more complex physics. AMR presents special challenges to visualization. SDAV technologies include production-quality, petascale-ready software tools for visualization and analysis of AMR data.

Preparing for the near future with next-generation visualization technology

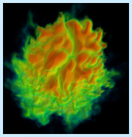
Efficient Performance with Hybrid Parallelism in VisIt and ParaView

- Hybrid Parallelism refers to a blend of shared- and distributed-memory parallel programming techniques within a program.
- Benefits come from many factors, including: (1) Reduced overhead from MPI, (2) Smaller memory footprint and improved caching effects from shared memory within a node





A hybrid-parallel streamline design (see image above), implemented by David Camp (LBL), ran up to ten times faster than distributed memory parallelism using the same resources. Benefits came from improved load balancing and from sharing of resources within a node (see charts on left).



This volume rendering image comes from a study on hybrid parallelism by Howison, Bethel, and Childs (LBL). They found, for a 216K core Jaguar run, that hybrid parallelism halved runtime (by reducing communication) and reduced MPI memory overhead by a factor of twelve.

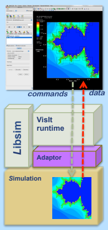
In-situ visualization and analysis frameworks

VisIt In Situ with Libsim

Connect VisIt to your simulation for in situ processing.

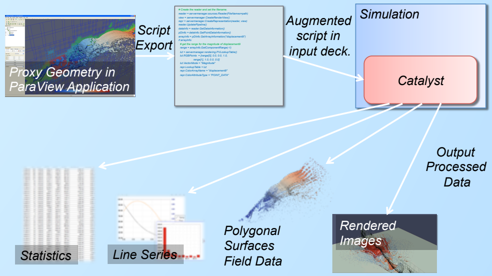
1. *Write adaptor code*
2. *Pass arrays to VisIt* (including zero copy)
3. *Visualize the results*

- *Perform any VisIt operation*
- *Watch plots update while advancing simulation*



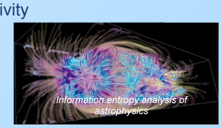
ParaView In Situ with Catalyst

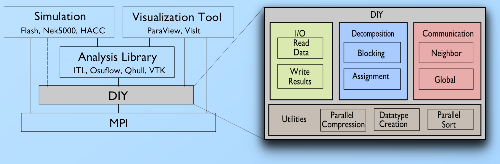
- A simplified library providing scalable visualization resources to simulation.
- A ParaView-enabled infrastructure provides familiar interactive tools to a variety of post-processing capabilities with minimal impact on simulation.



DIY (Do-It-Yourself Library)

- Main Ideas and Objectives
  - Large-scale parallel analysis (visual and numerical) on HPC machines
  - For scientists, visualization researchers, tool builders
  - In situ, coprocessing, postprocessing
  - Data-parallel problem decomposition
  - MPI + threads hybrid parallelism
  - Scalable data movement algorithms
  - Runs on Unix-like platforms, from laptop to supercomputer (including all IBM and Cray HPC leadership machines)
- Benefits
  - Researchers can focus on their own work, not on parallel infrastructure
  - Analysis applications can be custom
  - Reuse core components and algorithms for performance and productivity





DIY usage and library organization