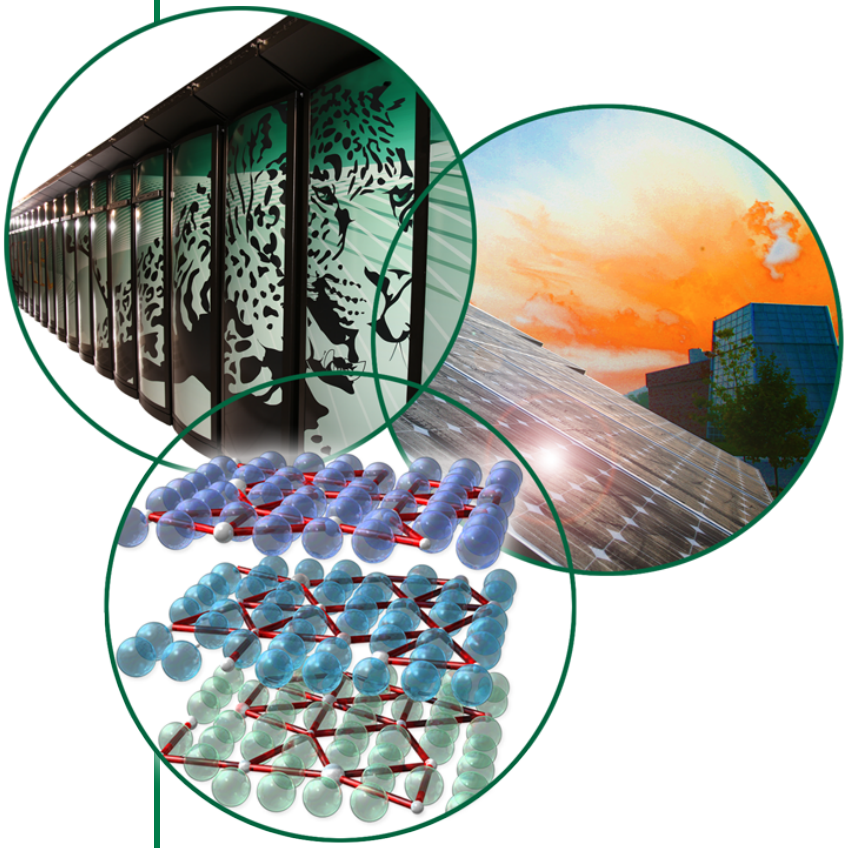
# Update on VisIt

Jeremy Meredith

DOECGF

April 13, 2010



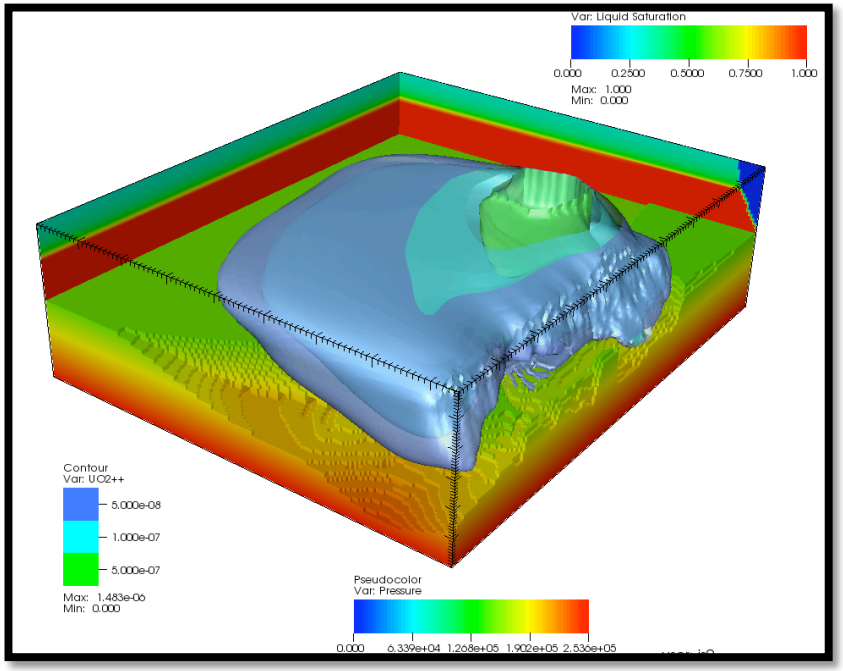
U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

# Overview

- Current Status
  - VisIt 1.12 released Summer 2009
  - VisIt 2.0 Beta released (last week)
- Outline
  - VisIt community
  - Software engineering
  - New features
  - Big changes
  - Research
  - Upcoming




PFLOTRAN groundwater simulation

# VisIt Community

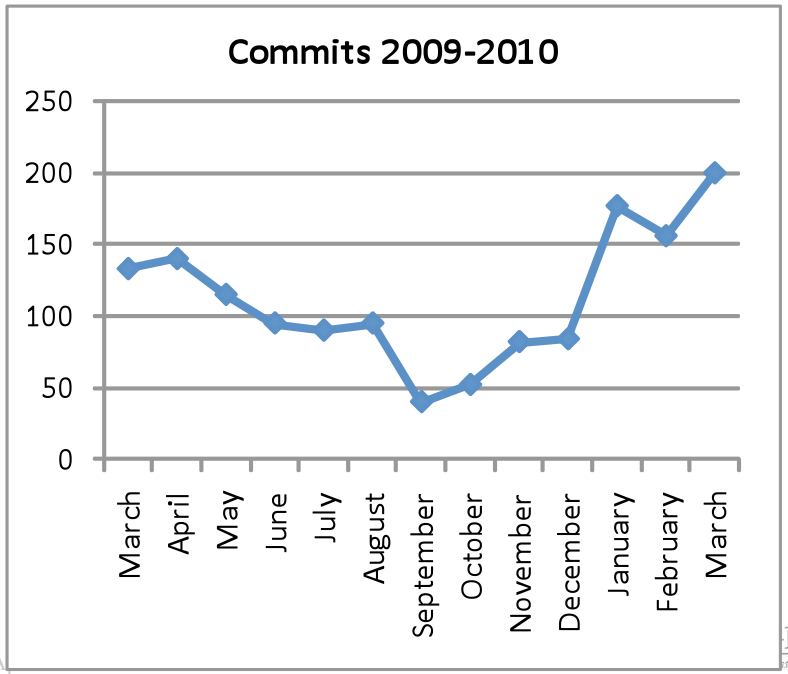
Commit emails

- Sean Ahern
- Kathleen Bonnell
- David Bremer
- Eric Brugger
- David Camp
- Hank Childs
- Rich Cook
- Marc Durant
- Cyrus Harrison
- Tom Fogal
- Gunther Weber
- Jeremy Meredith
- Mark Miller
- Paul Navratil
- Prabhat
- Dave Pugmire
- Oliver Ruebel
- Allen Sanderson
- Brad Whitlock

SVN accounts (NERSC SciDAC Outreach Center)

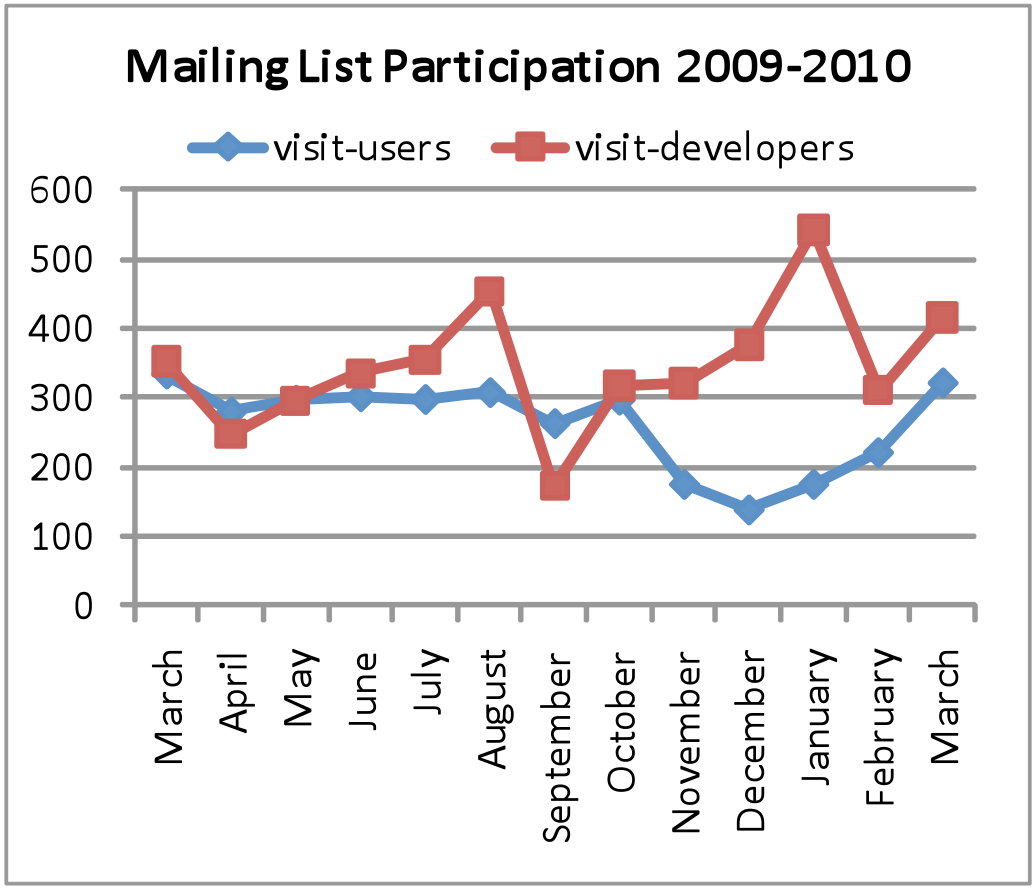
- US Nat'l Labs:
  - LLNL 7
  - ORNL 5
  - LBNL 4
  - ANL 2
- University:
  - Utah 4
  - UC Davis 3
  - UT/Austin 1
  - NCSA 1
  - CalTech 1
  - LSU 1
- Private industry:
  - Tech-X 1
- Foreign Labs:
  - AWE 2
  - CEA 1
  - Max Planck 1

- Both NSF XD centers deploying VisIt on their visualization resources and anticipating VisIt development to support their user communities
- And more!
  - Contributions from users
  - Embedded-viewer applications
  - Plug-ins (plots, readers, operators) maintained at external sites
  - Various projects in development right now



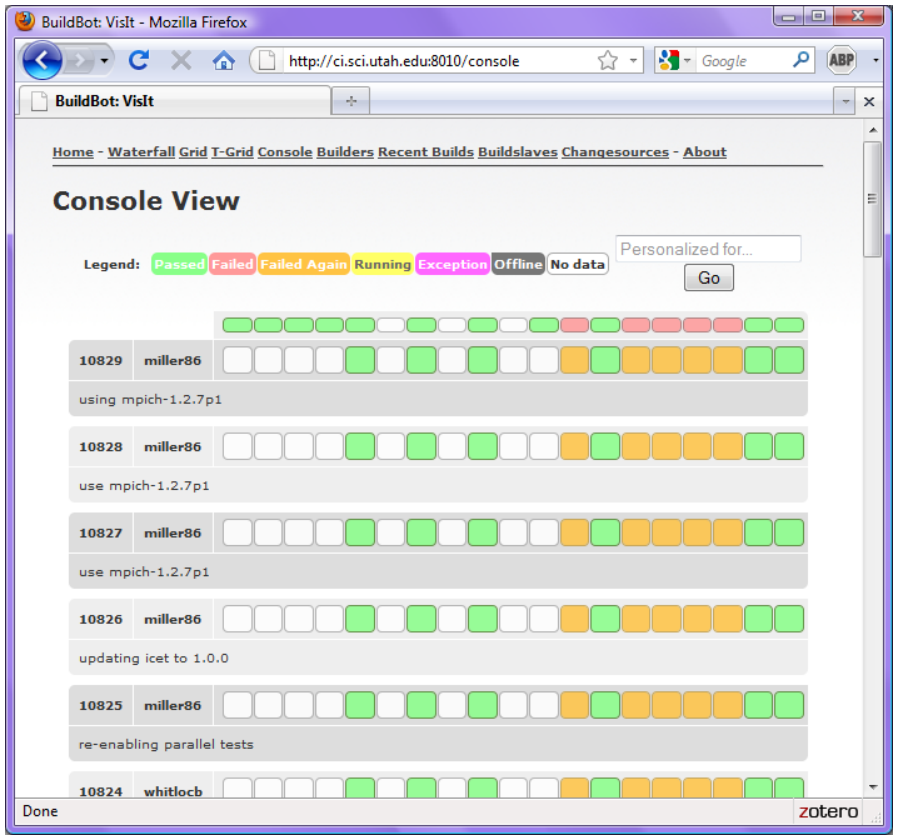
# VisIt Community

- http://email.ornl.gov
  email lists
  - visit-users
    - 417 members
  - visit-developers
    - 92 members
  - visit-help-funded
- http://visitusers.org
  - wiki
  - forum
- http://visit.llnl.gov



# Software Engineering

- Anonymous read access to subversion repository
- Shared issue tracker
  - as soon as hosting is ready
- Upstream commits
  - contributions to Mesa, GLEW
- New CMake build system
  - reduce porting effort
- BuildBot (as seen here ➔)
  - continuous integration



# Basic New Stuff

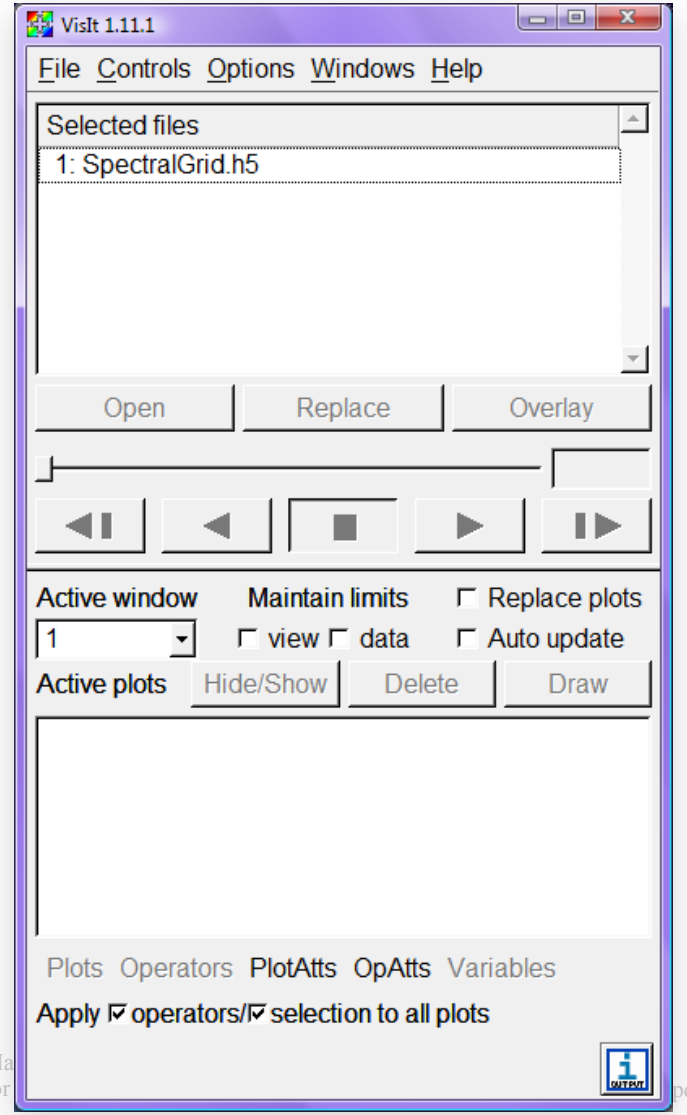
- Plot Plugins (now 23 of them)
  - Poincaré
- Operator Plugins (now 49 of them)
  - Edge
  - Delaunay
  - TriangulateRegularPoints
- Database Plugins (now 105 of them)
  - VisSchema
  - LAMMPS
  - PlasmaState
  - paraDIS
  - Velodyne
  - MatrixMarket
  - Adventure AdvIO
- Library upgrades
  - Python, Qt, Mesa, etc.
- Internationalization
  - new Dutch translation (+JP, FR)
- VisTrails support
  - automatically track provenance
- New machine support
  - BG/P, XT
- Analysis features
  - new expressions, queries
  - improvements
    - e.g. connected components
- Other improvements
  - plots, operators, file reader
  - too many to list!

# User interface updates

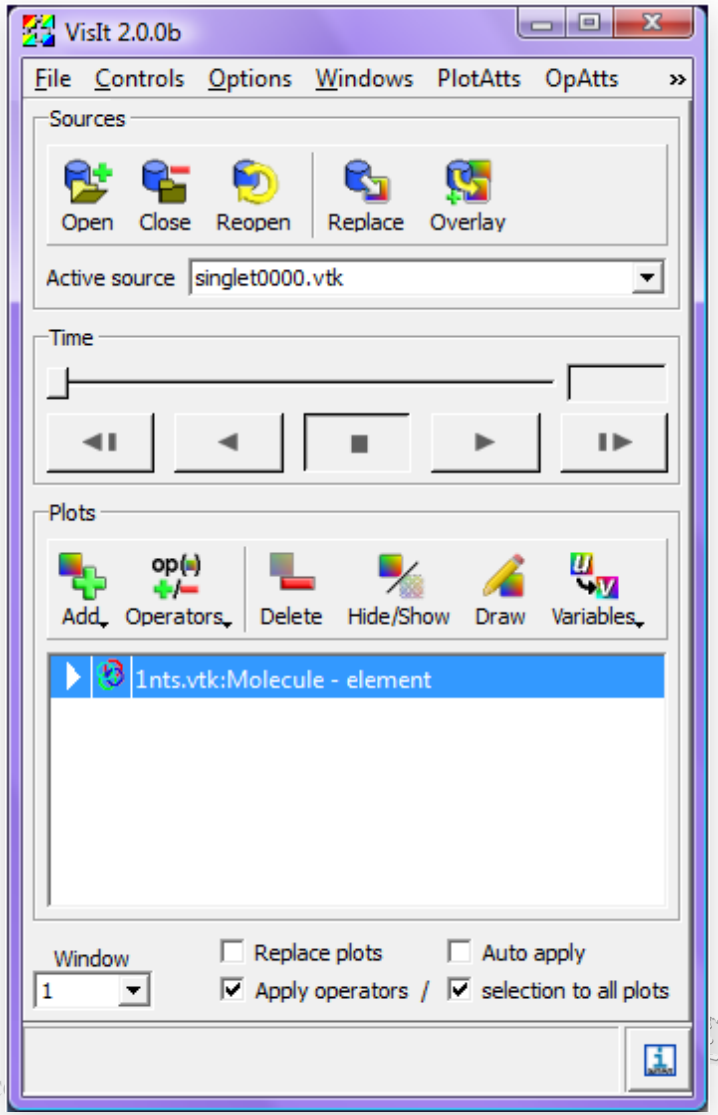
- Qt4 based GUI
- Selected files mode no longer the default
  - Many enhancements to make the new default mode highly usable
- Plot list enhancements
  - naming, re-ordering, cloning
- Operator categorization
- Unix/Linux/Windows/Mac OS X consistency
- Main window reorganized
  - fewer menu operations and mouse clicks
  - spatial proximity for common operations

# GUI main window updates

## Old (1.11)

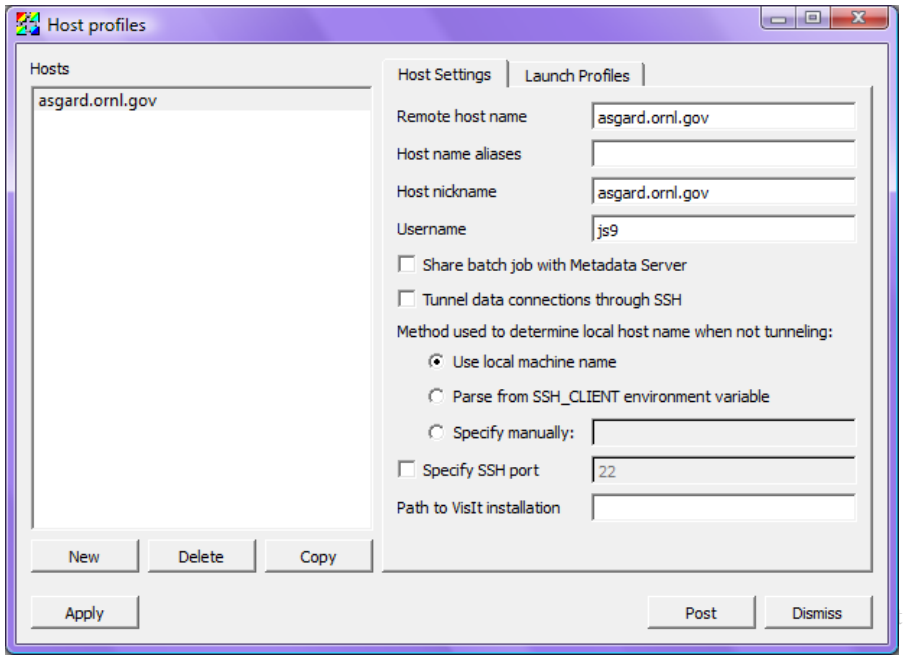


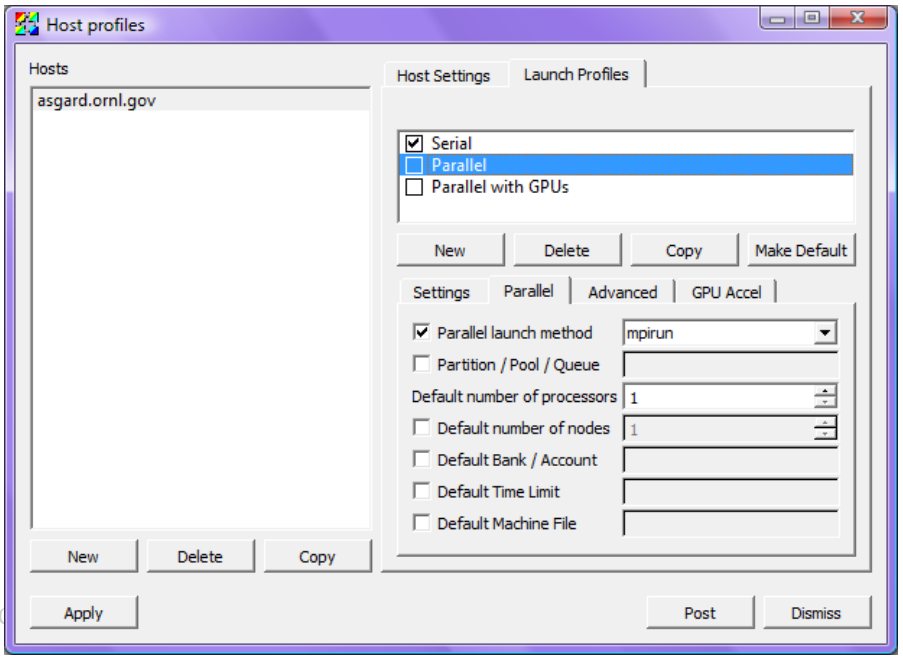

## New (2.0)



# Host profile updates

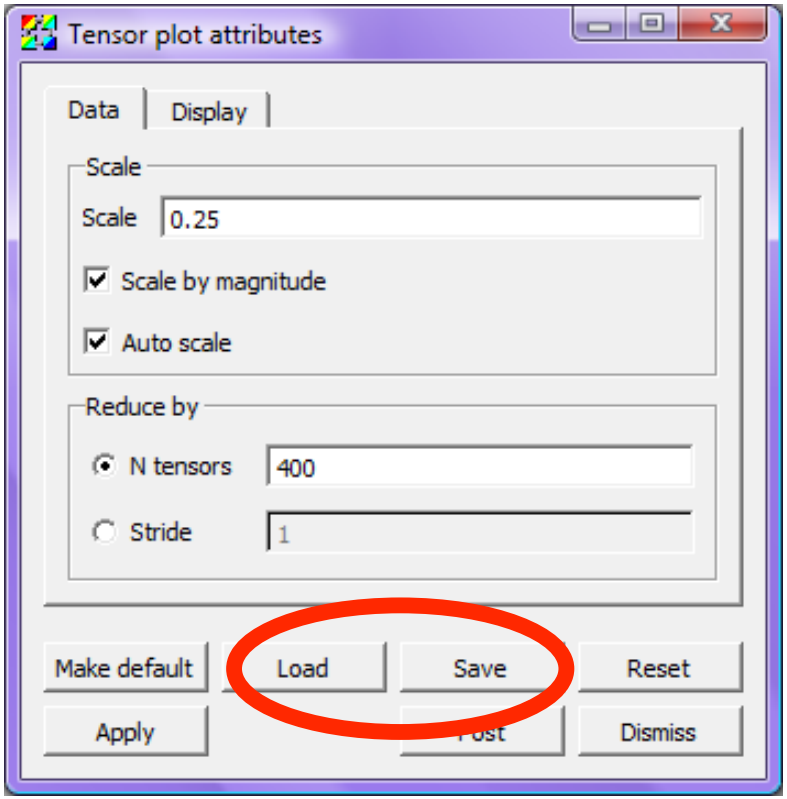
- Clearly define separation between two roles:
  - how to access machines (ssh, username, paths)
  - how to launch batch jobs (launcher, processors, GPUs)
- Extract from session/configuration management
  - Host accessing and launching procedures can change
  - Restoring an old session requires up-to-date host profiles
  - Sharing profiles for new or modified hosts should be trivial



# Load/Save individual attributes

- All plots and operators have Load and Save buttons
  - Only that plot/operator's settings
  - Standalone XML file
    - lighter weight than a whole session file
    - easy exchange between sessions, users
  - Examples:
    - transfer function of a volume plot
    - complex bond creation list



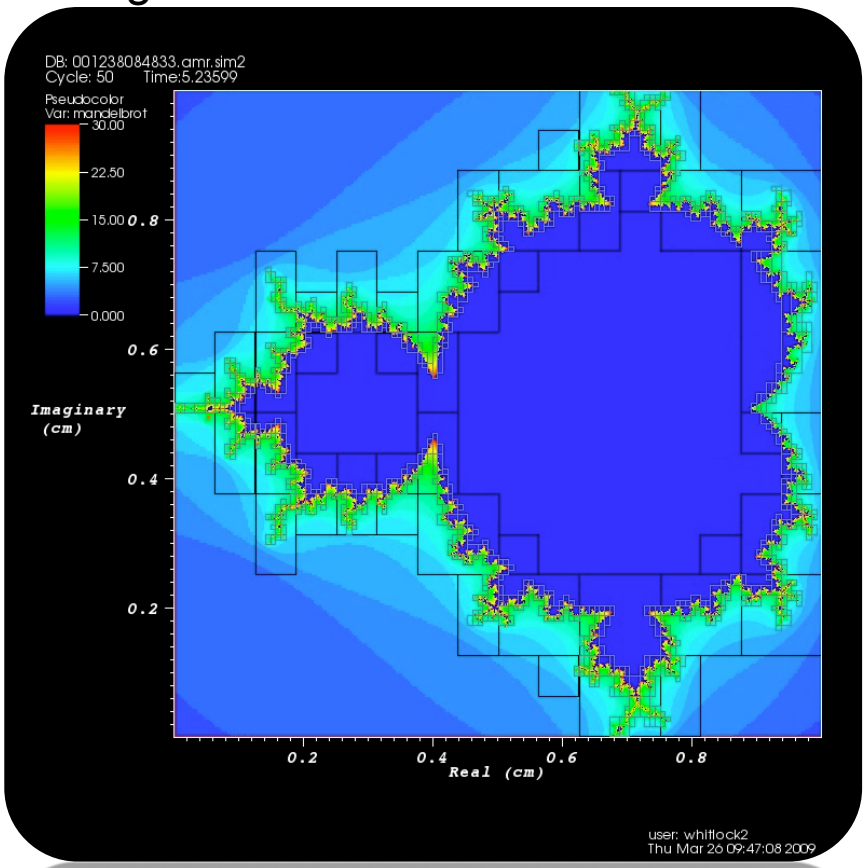
# File format detection improvements

- No longer restricted to a .extension for file pattern match
  - e.g. VASP takes “CHG*”, CEAUCD takes “U_#*_#*.inp”
- Setting preferred database plugins can be done in GUI
- Remember which plugin opened a file
  - Preserved across session files
- Many readers no longer blindly accept other types files
  - But forcing a specific plugin can relax some stricter checks
- Extra checks and warnings
  - If two plugins can open a file, warn the user we had to guess

# In-situ simulation interface V2

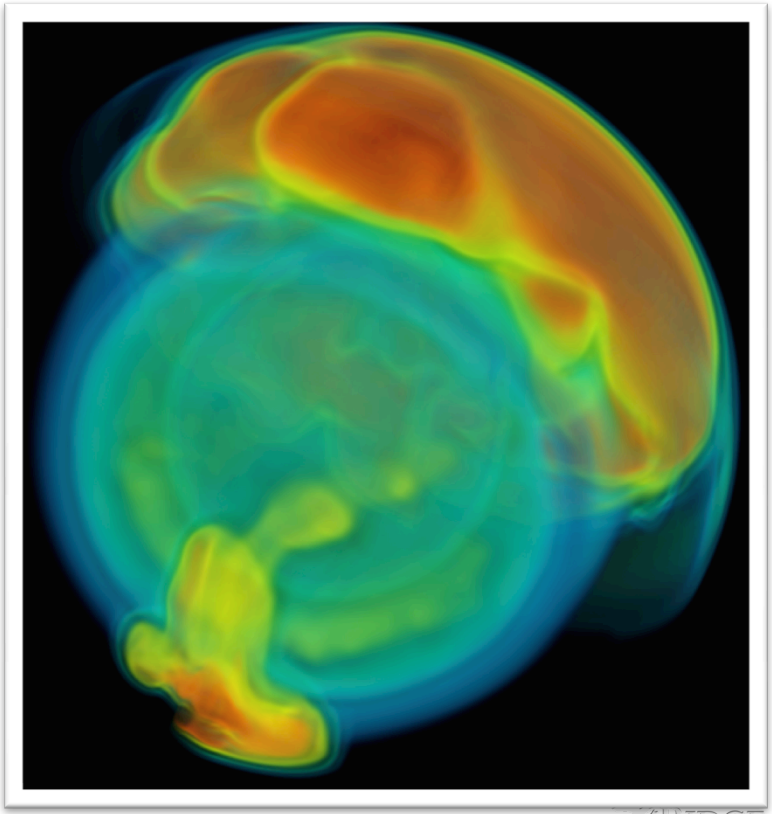
- **Additional data types**
  - Material species
  - CSG meshes
  - AMR meshes
  - Vector & Tensor data
- **New Functionality**
  - Save images directly from the simulation
  - Write files with trace information
  - Write information to VisIt's debug logs
- **Cleaner interface**
  - Prevent and detect errors
  - Improved robustness
  - C/Fortran APIs are now the same

Dynamic Mandelbrot set calculation using AMR and new simulation API

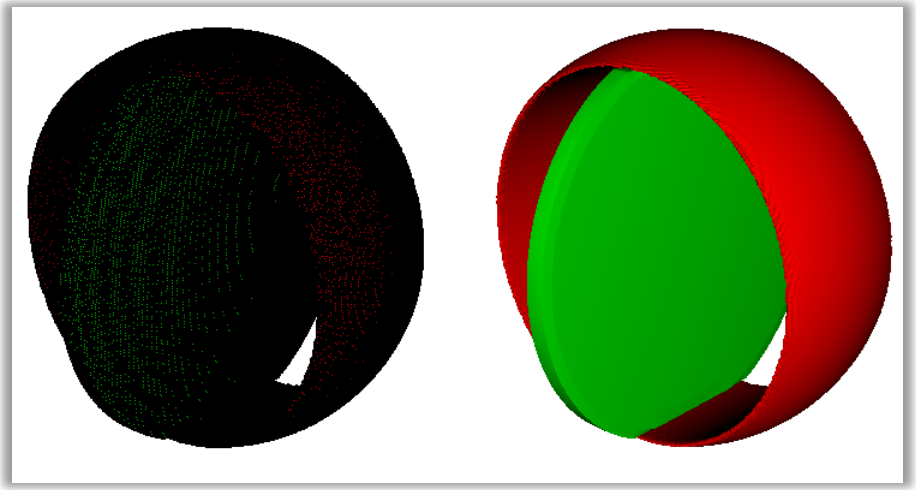


# Scaling Improvements

- Enable enormous runs: 64k cores, 8 trillion cells
- Vast numbers of domains, fewer collective MPI calls
- Memory improvements for happier <1GB/core runs
- Knock down bottlenecks
  - parallel startup costs
  - compositing (IceT now default)
  - adaptive mesh refinement data
  - analysis (connected components)
  - volume rendering
- First visual analysis code to be part of ASCR Joule metric

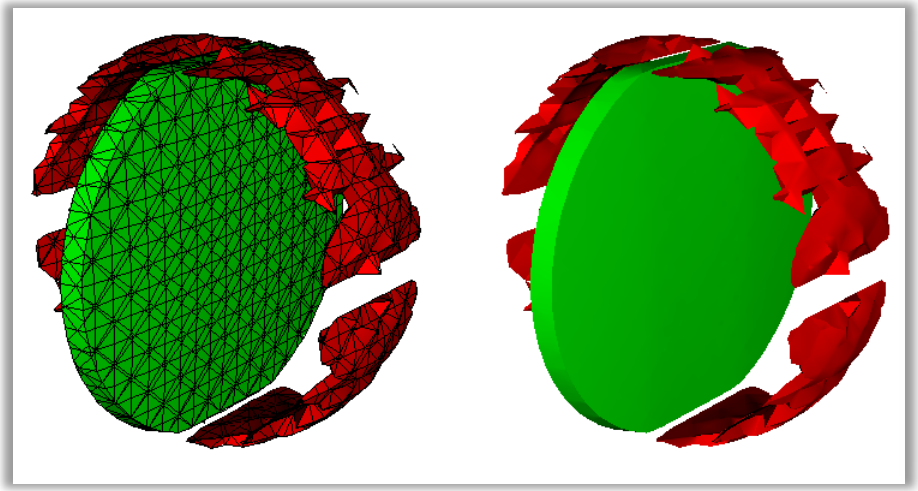


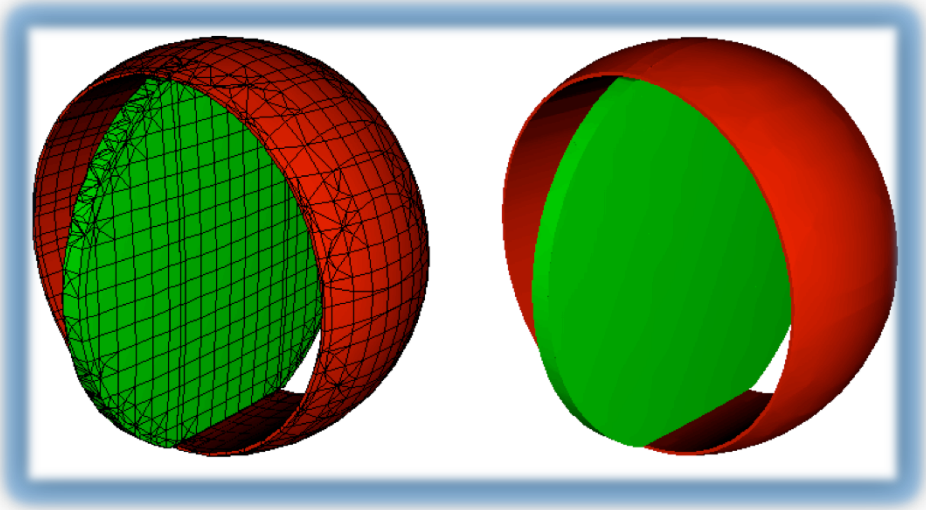
# Multi-pass CSG algorithm



- Repeated splitting + region tagging
- Achieves thin shells, sharp edges
- Slower than Uniform at same size
  - but allows much coarser mesh

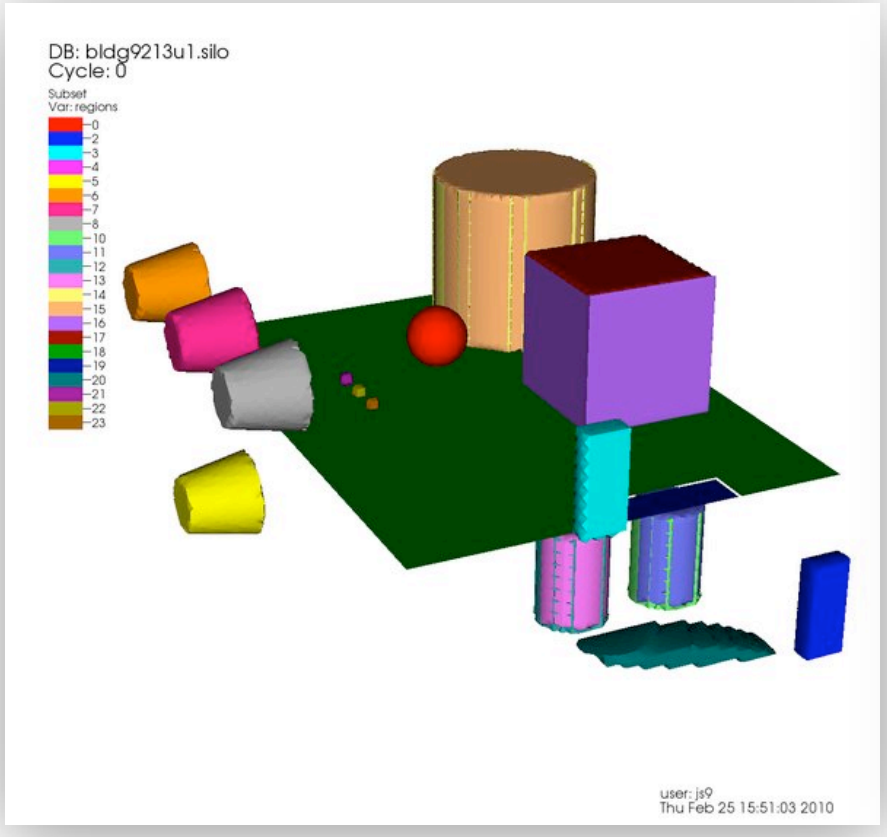
*Left:* *Uniform at 200^3*
*Lower-left:* *Uniform at 20^3*
*Below:* *Multi-pass 20^3*



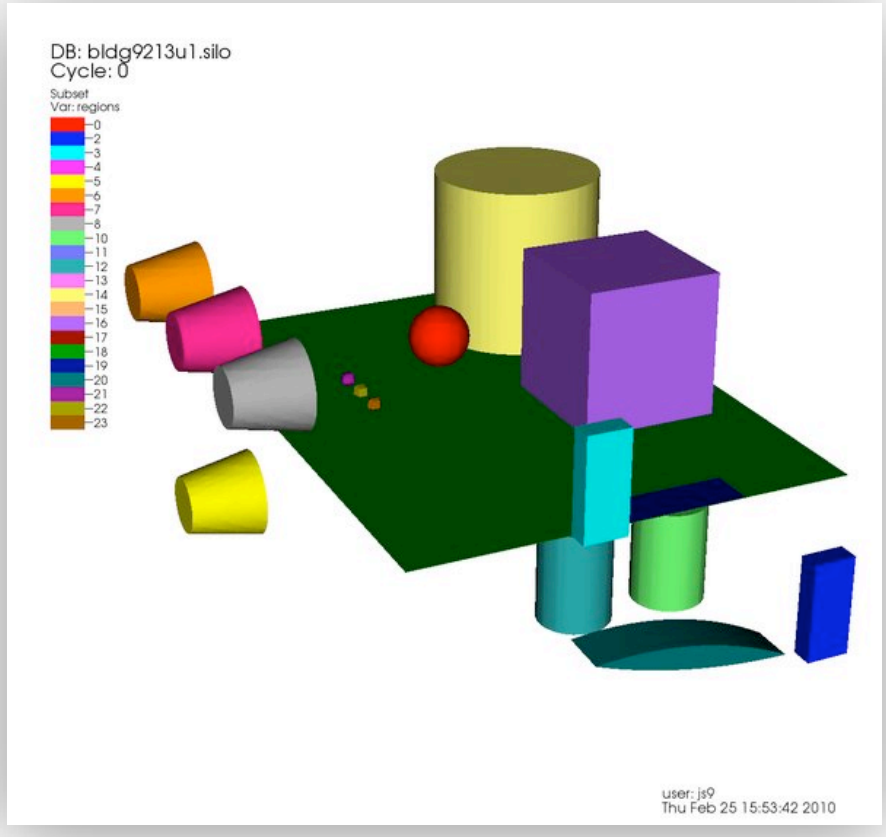


# CSG in reactor design

## uniform 100x100x100

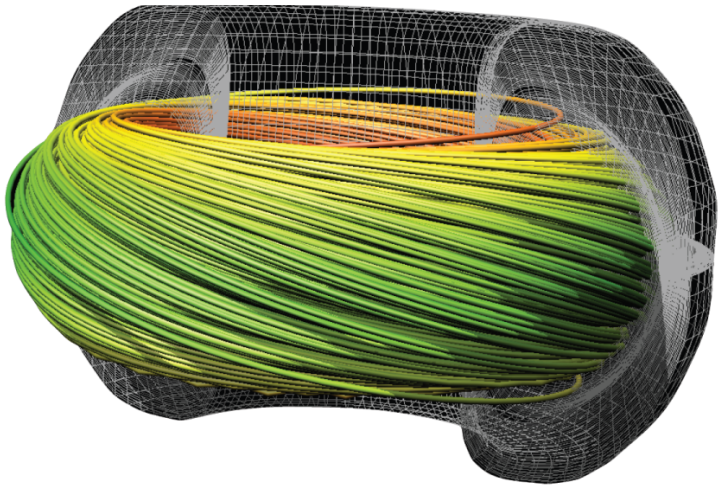


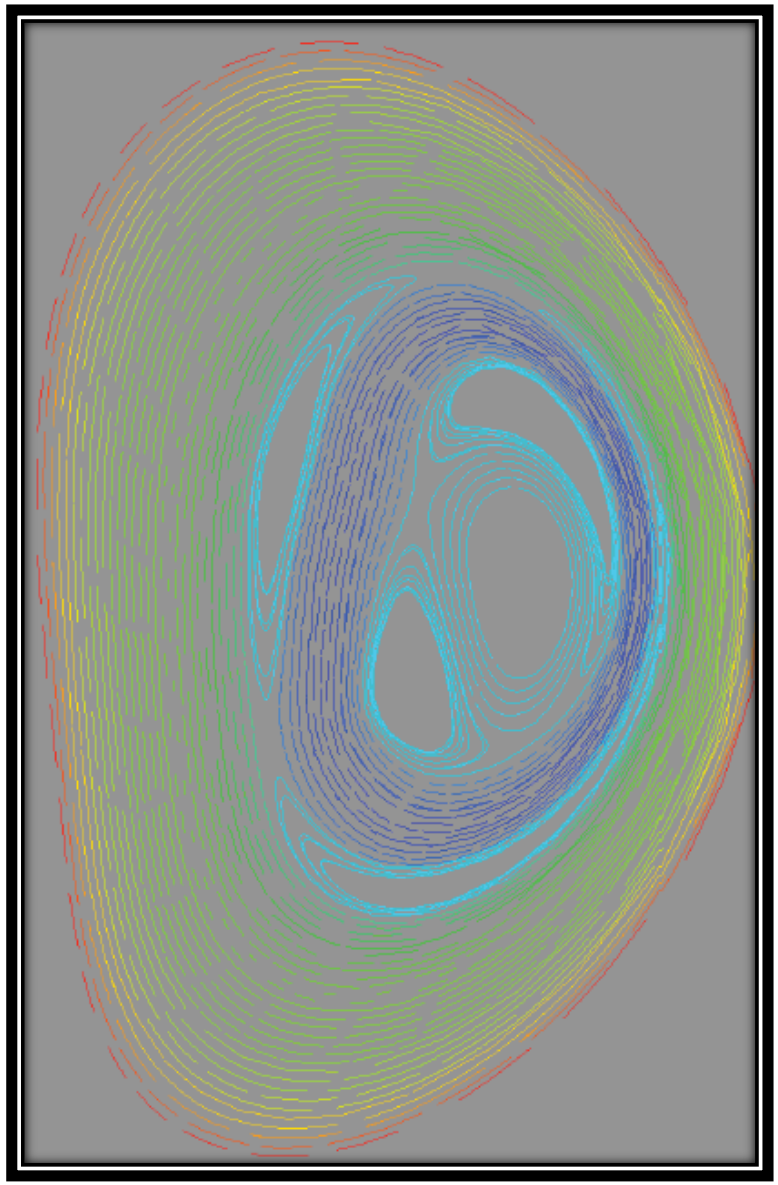

## multi-pass 100x100x100



# Poincaré plot

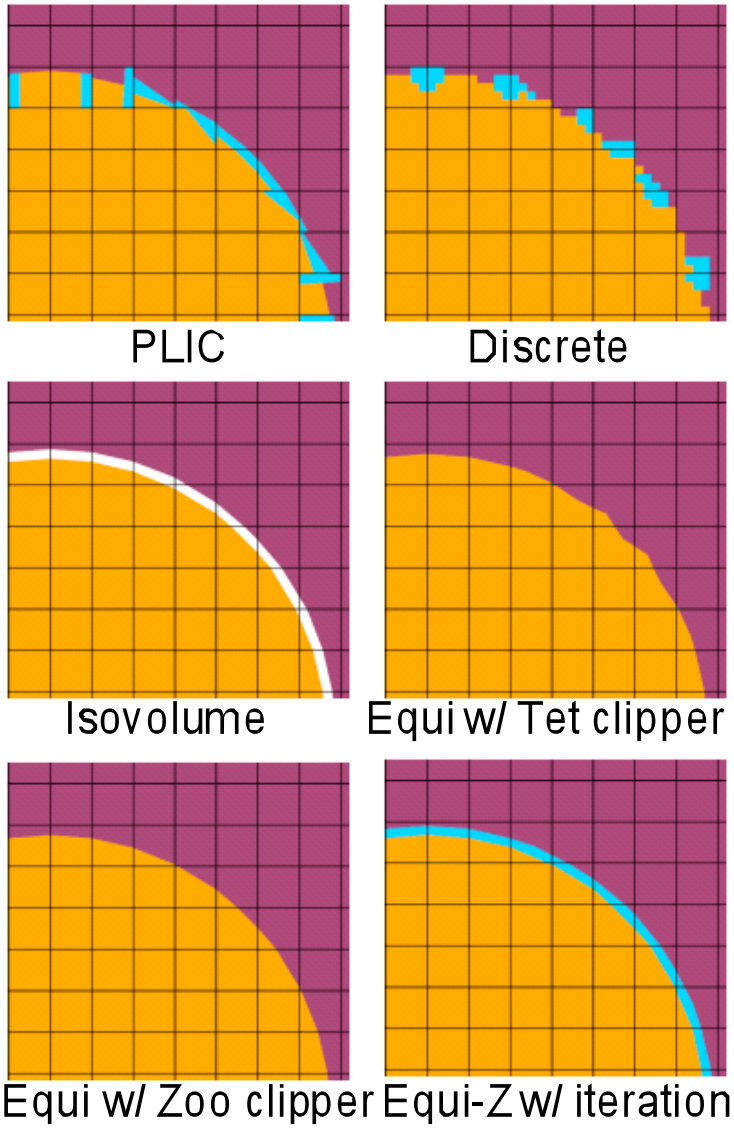
- Cross section of a curve integral
- Requires accurate streamlines
- Analyze topological structures
  - e.g. magnetic field in fusion reactor





# Material Interface Reconstruction

- New “Discrete” algorithm
  - contributed by John Anderson
  - fixed volume fraction error bounds
  - rectilinear grids
- New PLIC (Youngs) algorithm
  - contributed by Thierry Carrard
  - 100% accurate VF
  - discontinuous boundaries
- Iterative scheme
  - improves VF accuracy of VisIt’s default (“equi-surface”) algorithm
  - upcoming EuroVis paper



# MIR on CFD simulation

Iso-volume

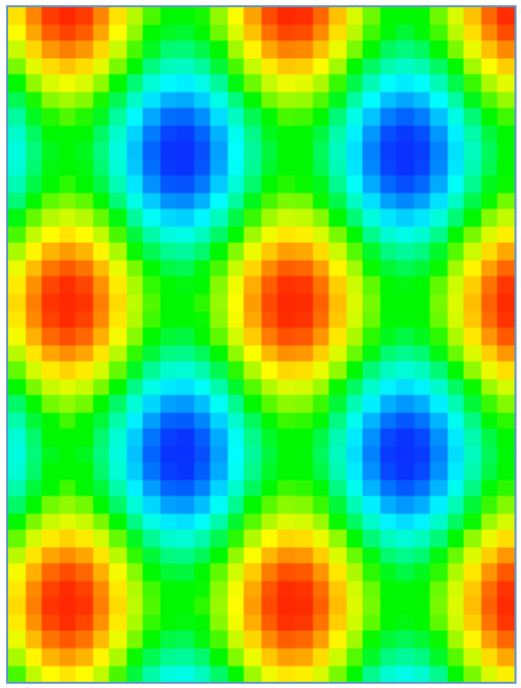
PLIC

Equi-surface

Equi- with iteration

# Python filters

- Plugins extend VisIt operations, but require C++ and a compiler
- Python filters allow rapid development of new expressions, queries
  - VTK-wrapped objects; can operate on individual cells, points

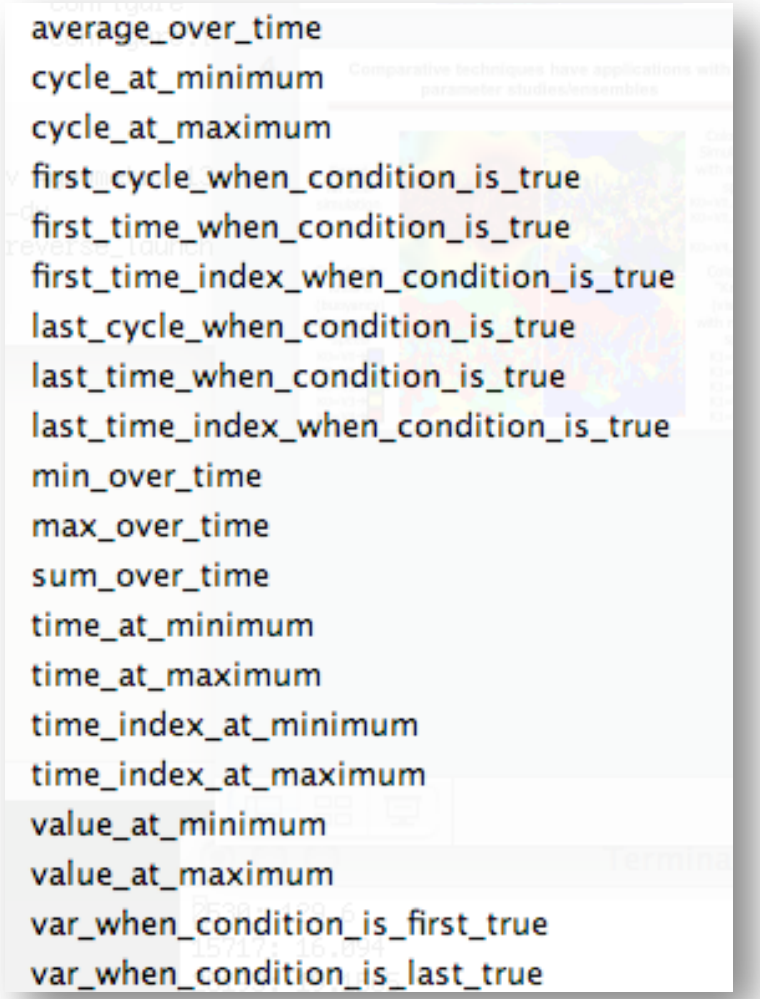


```
from math import sin, pi
class MyExpression(SimplePythonExpression):
    def __init__(self):
        SimplePythonExpression.__init__(self)
        self.point_var  = False
        self.output_dim = 1
    def derive_variable(self,ds_in,domain_id):
        # ds_in is a vtk dataset, we want
        # to create and return a new vtkDataArray
        # that contains a simple sine wave pattern
        ds_bounds = ds_in.GetBounds()
        x_ext = ds_bounds[1] - ds_bounds[0]
        y_ext = ds_bounds[3] - ds_bounds[2]
        ncells = ds_in.GetNumberOfCells()
        res = vtk.vtkFloatArray()
        res.SetNumberOfComponents(1)
        res.SetNumberOfTuples(ncells)
        for i in xrange(ncells):
            cell = ds_in.GetCell(i)
            bounds = cell.GetBounds()
            xv = bounds[0] + bounds[1] / 2.0
            yv = bounds[2] + bounds[3] / 2.0
            res.SetTuple1(i,.25 * (sin(xv*3*pi/x_ext) + sin(yv * 3*pi / y_ext)))
        return res

py_filter = MyExpression
```

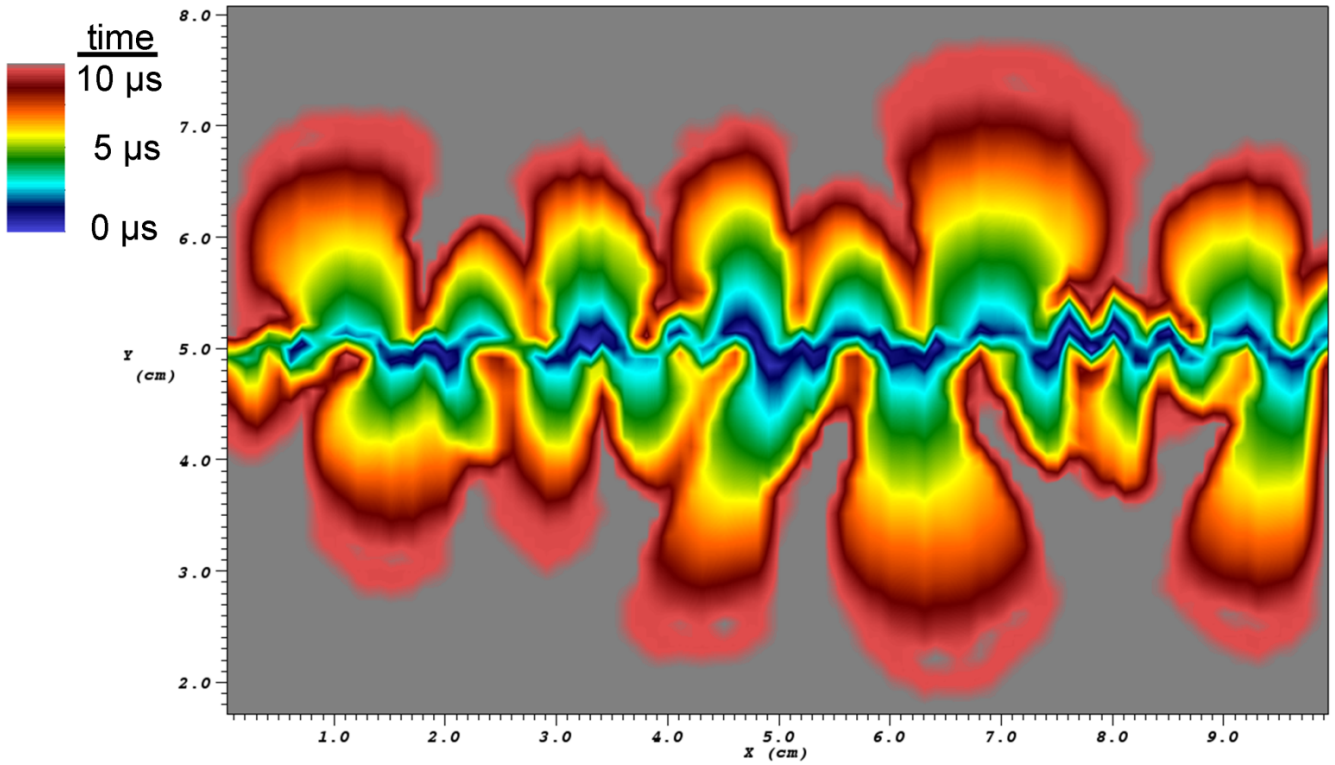
# Calculate derived fields over time

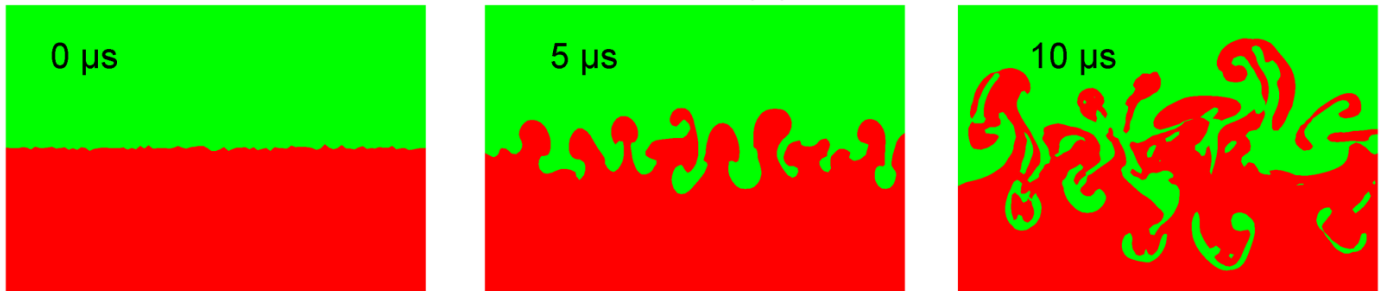
- Will read selected time slices and calculate a new field that combines the results of all time slices.
- Efficient in that:
  - there are no intermediate arrays
  - one time slice in memory at a time
- This was previously possible in VisIt … but hard & inefficient



# Calculate derived fields over time

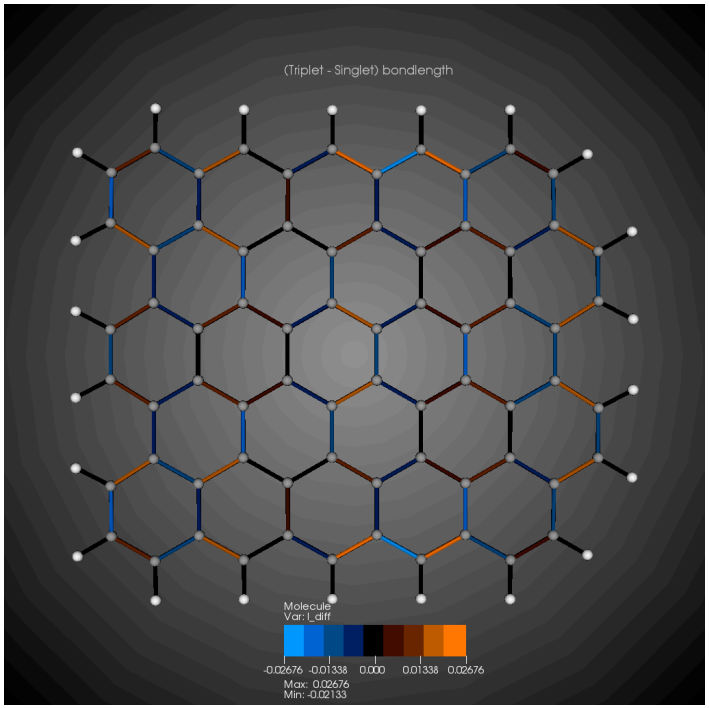
- **Expression #1: is_mixed = gt(nmats(material), 1)**
- **Expression #2: first_time_when_condition_is_true(is_mixed, …)**

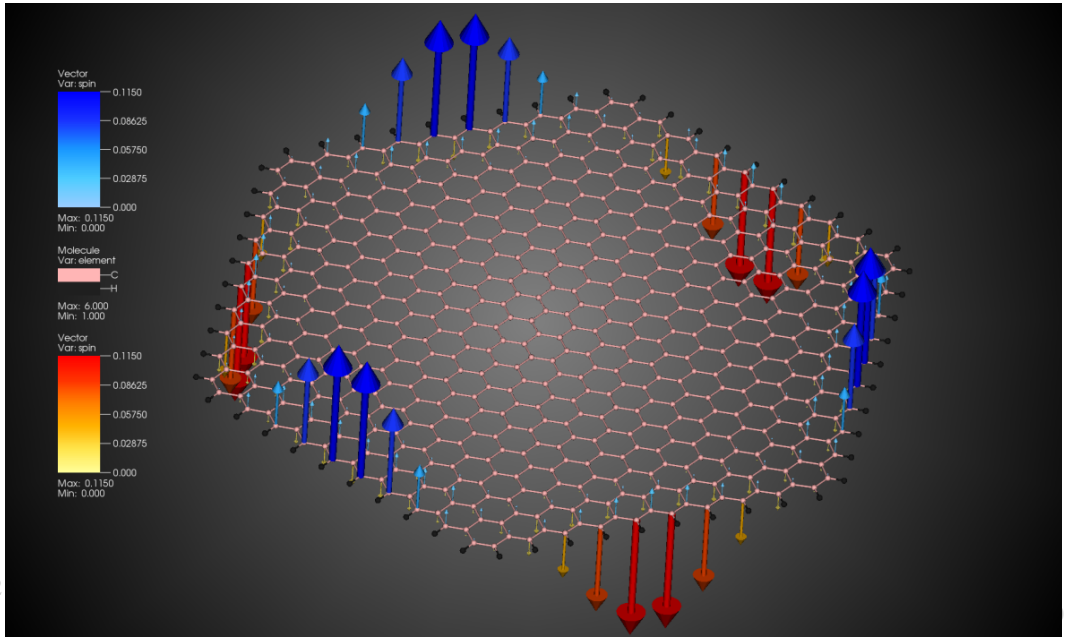


# Molecular visualization

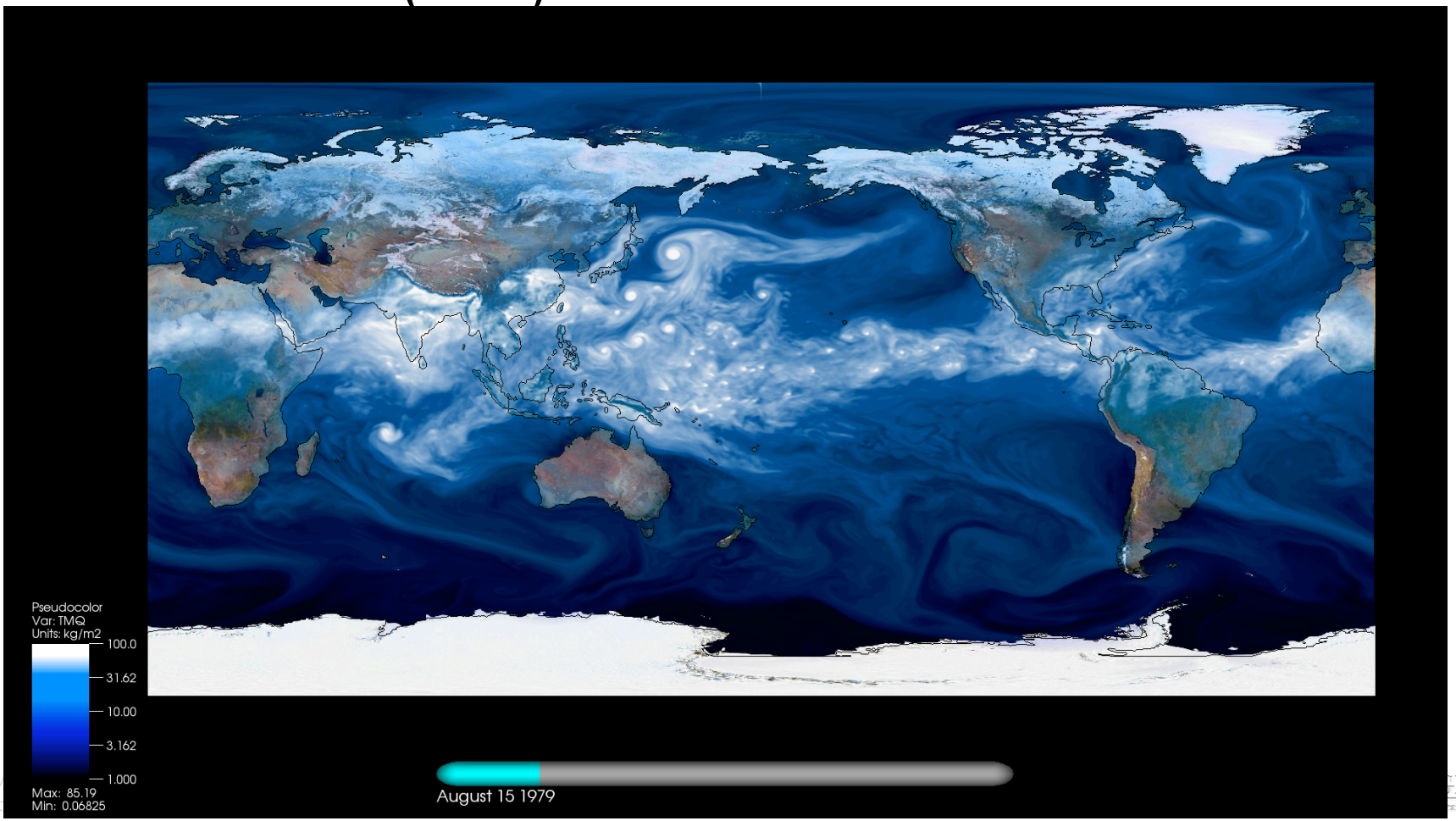
- Visualization of graphene patches
  - change in bondlength (right)
  - distribution of spin (below)
  - Michael Philpott, Sinisa Vukovic (UC Berkeley)



# Climate visualization

- Visualizations of CCSM simulations
  - Hurricane formation/evolution, climate, global warming
  - Michael Wehner (LBNL)



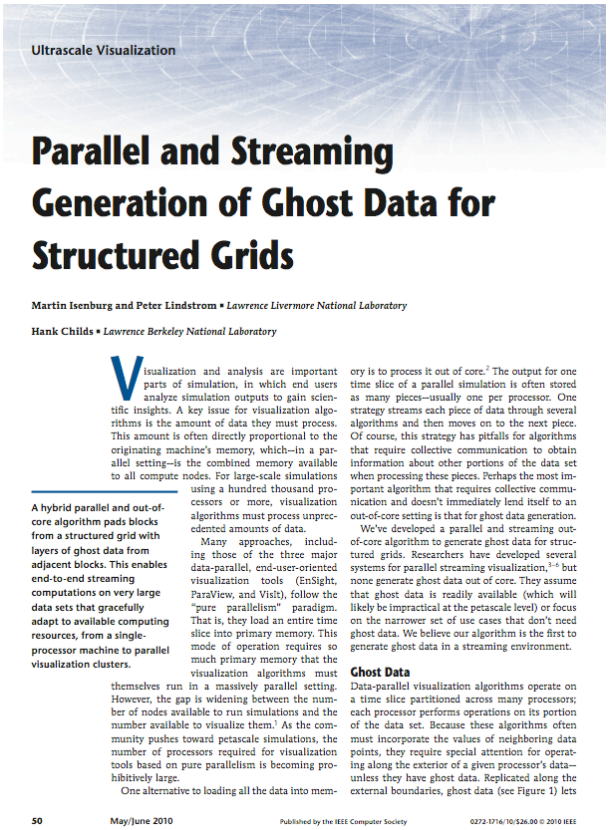
# Selected Publications

- J.S. Meredith and H. Childs, “Visualization and Analysis-Oriented Reconstruction of Material Interfaces”, EuroVis 2010, Bordeaux, France, June 2010.
- D. Pugmire, H. Childs, C. Garth, S. Ahern, G. Weber, "Scalable Computation of Streamlines on Very Large Datasets." SC09, Portland, OR, November 2009.
- C. Garth, E. Deines, K. Joy, E.W. Bethel, H. Childs, G. Weber, S. Ahern, D. Pugmire, A. Sanderson, C. Johnson. "Vector Field Visual Data Analysis Technologies for Petascale Computational Science", SciDAC Review, December 2009.
- H. Childs, D. Pugmire, S. Ahern, B. Whitlock, M. Howison, Prabhat, G. Weber, E.W. Bethel. “Extreme Scaling of Production Visualization Software on Diverse Architectures”, IEEE Computer Graphics and Applications, June 2010.
- M. Isenburg, P. Lindstrom, & H. Childs ,"Parallel and Streaming Generation of Ghost Data for Structured Grids“, Computer Graphics & Applications, special issue on Ultrascale Visualization, May/June 2010, p. 50-62.
- G.H. Weber, S. Ahern, E.W. Bethel, S. Borovikov, H.R. Childs, E. Deines, C. Garth, H. Hagen, B. Hamann, K.I. Joy, D. Martin, J.S. Meredith, Prabhat, D. Pugmire, O. Rubel, B. Van Straalen, K. Wu, "Recent advances in VisIt: AMR streamlines and query-driven visualization." Astronomical Society of the Pacific Conference Series, Numerical Modeling of Space Plasma Flows (Astronum) 2009.

VisIt on SciDAC Review Covers 2009-10

# Upcoming:
# Out-of-core ghost data generation

- Ghost data is normally generated when all of the data is in core.
  - (How VisIt has worked for the last ten years.)
- Hard to generate it out-of-core … once you've read in a block, you don't want to do it again.
- Need to add special hooks to your reader and add runtime flags to access this mode (truly beta).



Ultrascale Visualization

**Parallel and Streaming Generation of Ghost Data for Structured Grids**

Martin Isenburg and Peter Lindstrom • *Lawrence Livermore National Laboratory*

Hank Childs • *Lawrence Berkeley National Laboratory*

**A hybrid parallel and out-of-core algorithm pads blocks from a structured grid with layers of ghost data from adjacent blocks. This enables end-to-end streaming computations on very large data sets that gracefully adapt to available computing resources, from a single-processor machine to parallel visualization clusters.**

Visualization and analysis are important parts of simulation, in which end users analyze simulation outputs to gain scientific insights. A key issue for visualization algorithms is the amount of data they must process. This amount is often directly proportional to the originating machine's memory, which—in a parallel setting—is the combined memory available to all compute nodes. For large-scale simulations using a hundred thousand processors or more, visualization algorithms must process unprecedented amounts of data.

Many approaches, including those of the three major data-parallel, end-user-oriented visualization tools (EnSight, ParaView, and VisIt), follow the "pure parallelism" paradigm. That is, they load an entire time slice into primary memory. This mode of operation requires so much primary memory that the visualization algorithms must themselves run in a massively parallel setting. However, the gap is widening between the number of nodes available to run simulations and the number available to visualize them.[1] As the community pushes toward petascale simulations, the number of processors required for visualization tools based on pure parallelism is becoming prohibitively large.

One alternative to loading all the data into memory is to process it out of core.[2] The output for one time slice of a parallel simulation is often stored as many pieces—usually one per processor. One strategy streams each piece of data through several algorithms and then moves on to the next piece. Of course, this strategy has pitfalls for algorithms that require collective communication to obtain information about other portions of the data set when processing these pieces. Perhaps the most important algorithm that requires collective communication and doesn't immediately lend itself to an out-of-core setting is that for ghost data generation.

We've developed a parallel and streaming out-of-core algorithm to generate ghost data for structured grids. Researchers have developed several systems for parallel streaming visualization,[3–5] but none generate ghost data out of core. They assume that ghost data is readily available (which will likely be impractical at the petascale level) or focus on the narrower set of use cases that don't need ghost data. We believe our algorithm is the first to generate ghost data in a streaming environment.

**Ghost Data**

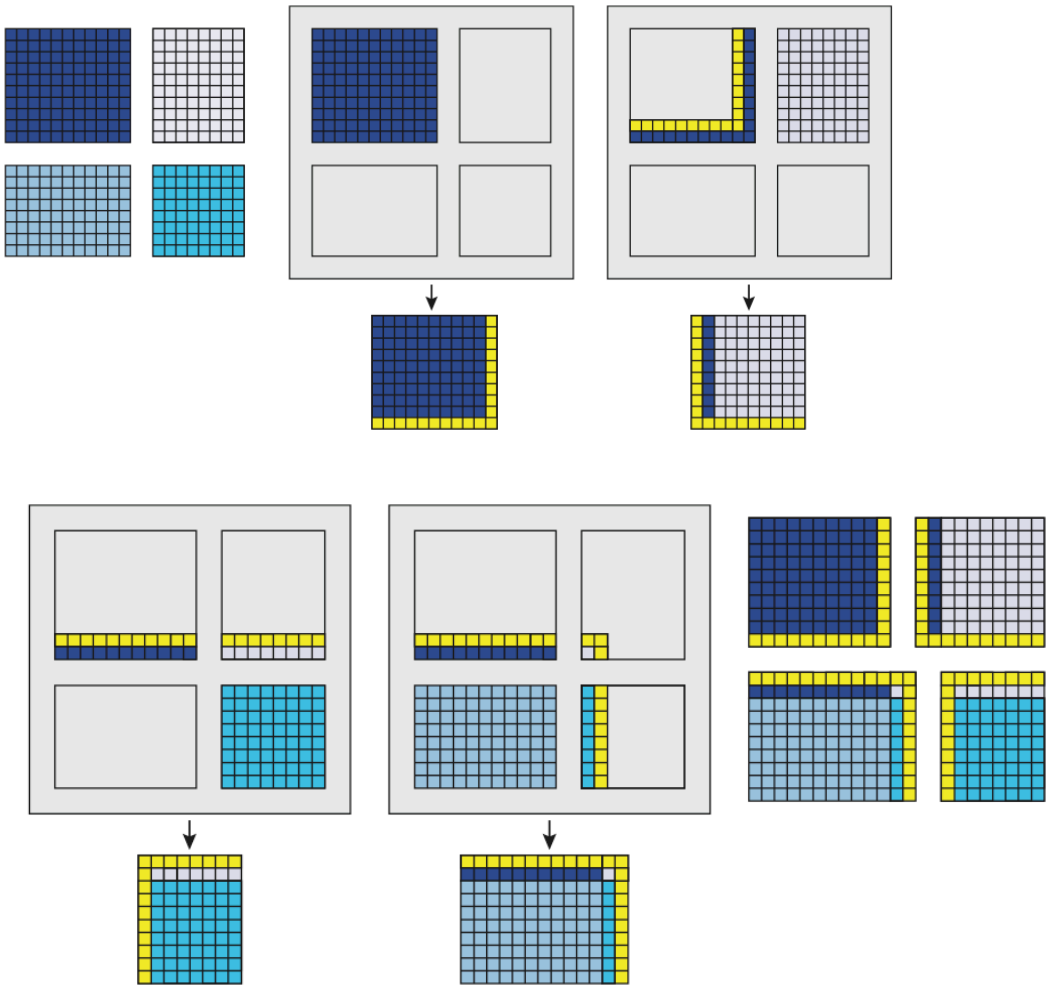
Data-parallel visualization algorithms operate on a time slice partitioned across many processors; each processor performs operations on its portion of the data set. Because these algorithms often must incorporate the values of neighboring data points, they require special attention for operating along the exterior of a given processor's data—unless they have ghost data. Replicated along the external boundaries, ghost data (see Figure 1) lets

50 May/June 2010 Published by the IEEE Computer Society 0272-1716/10/$26.00 © 2010 IEEE

Upcoming CG&A article on Ultrascale Visualization.
Authors: Isenburg, Lindstrom, & Childs

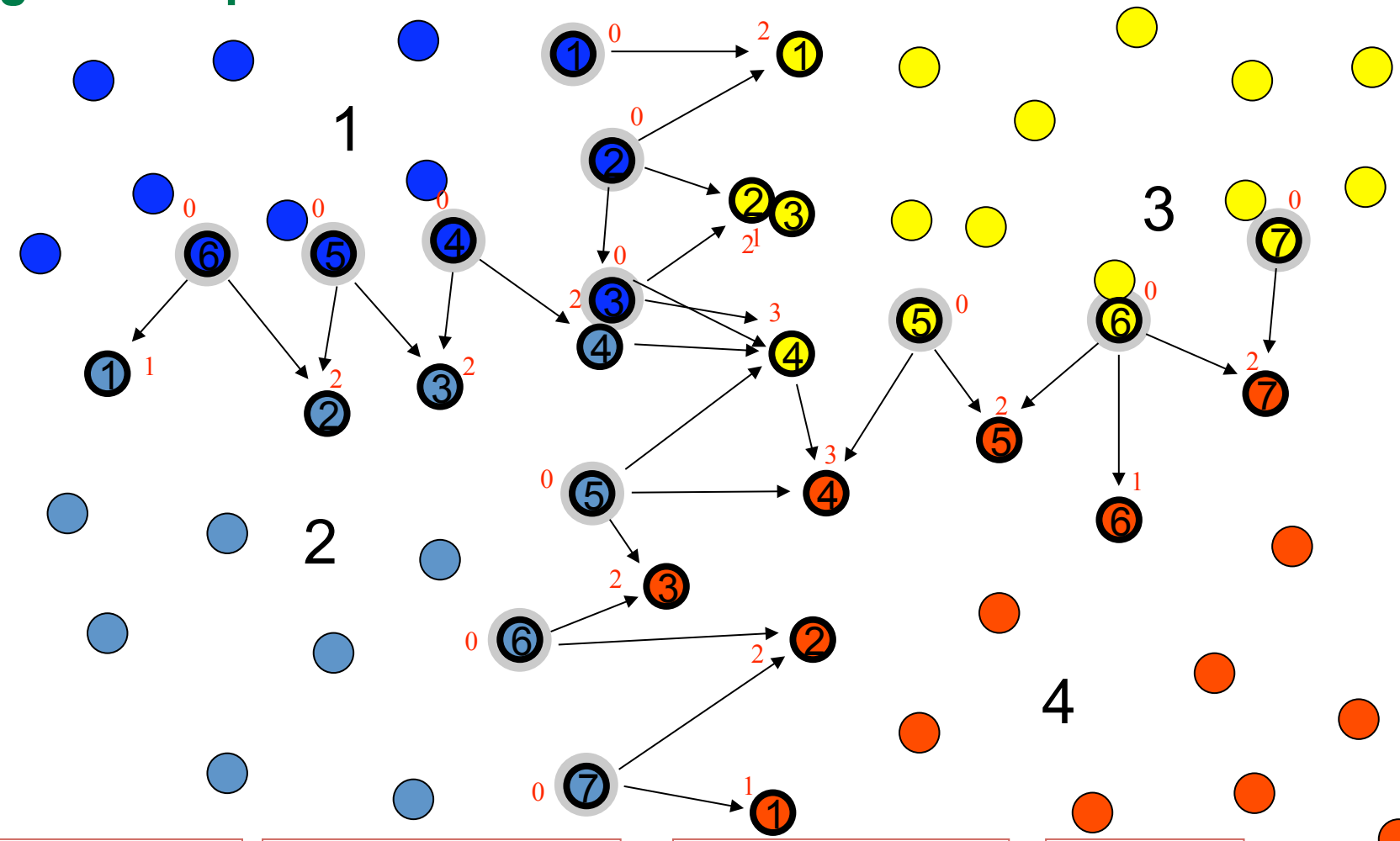
# Out-of-core ghost data generation

- First block is accessed
  - Mark real cells as ghost
  - Save two layers for later
    - one real and one ghost
- When its neighbor is accessed
  - Some shared data becomes part of this block
  - And others become ghosts
- Each block
  - receives from predecessors
  - sends only to successors
  - is accessed only once



# Out-of-core ghost data generation

- Significant parallel ramifications.




process 1
send 1,2,3 to 3 wait 1,2,3,4 from 3
send 3,4,5,6 to 2 wait 1,2,3,4 from 2

process 2
wait 3,4,5,6 from 1 send 1,2,3,4 to 1
send 4,5 to 3 wait 4 from 3
send 5,6,7 to 4 wait 1,2,3,4 from 4

process 3
wait 1,2,3 from 1 send 1,2,3,4 to 1
send 4 to 4
send 5,6,7 to 4 wait 1,2,3,4 from 2

process 4
wait 4,5,6,7 from 3
wait 5,6,7 from 2
Smooth interior

# Future plans

- Finish testing and release version 2.0
- Customer driven focus
  - Many ongoing projects
  - Focus on the science
- Prepare for exa-scale
  - smarter visualization algorithms
  - streaming, out-of-core, multi-resolution, *in situ*