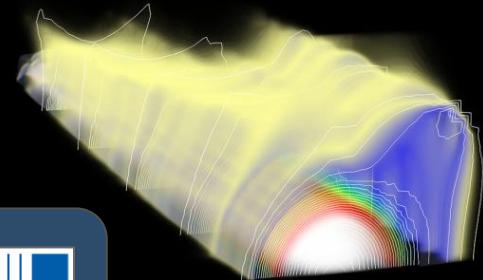


# Visualization and Analysis of HPC Simulation Data with VisIt

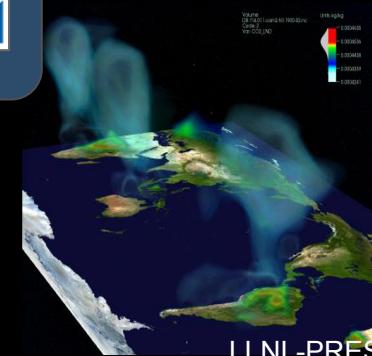
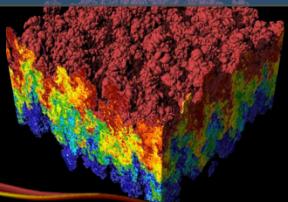
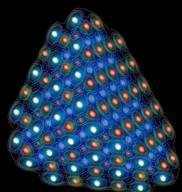
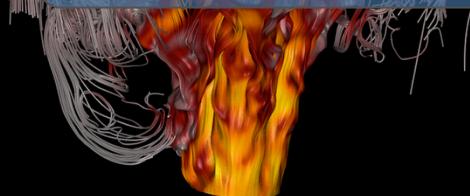
## ATPESC 2016

Argonne Training Program on Extreme-Scale Computing

Wednesday August 10, 2016



Cyrus Harrison  
Lawrence Livermore National Laboratory  
[cyrush@llnl.gov](mailto:cyrush@llnl.gov)



This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

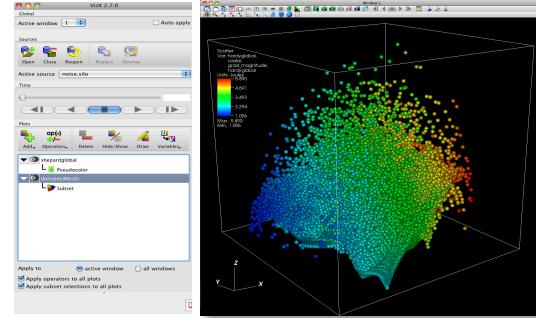
LLNL-PRES-699924



# ATPESC 2016 Outline

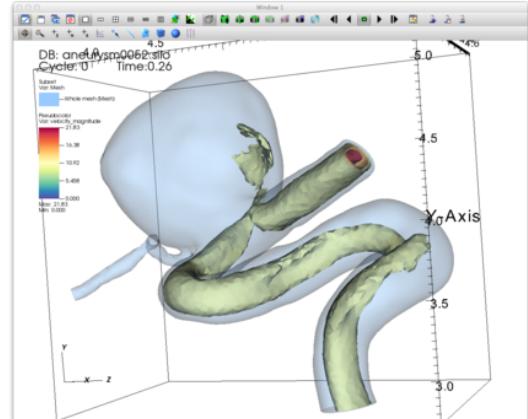
- **Talk**

- VisIt Project Introduction (10 min)
- Guided Tour of VisIt (20 min)
- Exploration of an Aneurysm Simulation (30 min)



- **Additional Hands-on Materials**

- Exploration of a Water Flow Simulation
- Volume Rendering
- Advanced Movie Making





# Tutorial Resources

---

- **Tutorial Prep**
  - [http://visitusers.org/index.php?title=Tutorial\\_Preparation](http://visitusers.org/index.php?title=Tutorial_Preparation)
- **Example Datasets**
  - [http://visitusers.org/index.php?title=Tutorial\\_Data](http://visitusers.org/index.php?title=Tutorial_Data)
- **Aneurysm Simulation Exploration**
  - [http://visitusers.org/index.php?title=Blood\\_Flow\\_Aneurysm\\_Tutorial](http://visitusers.org/index.php?title=Blood_Flow_Aneurysm_Tutorial)
- **Additional Tutorial Materials**
  - [http://visitusers.org/index.php?title=VisIt\\_Tutorial](http://visitusers.org/index.php?title=VisIt_Tutorial)
- **Cyrus' Email:** [cyrush@llnl.gov](mailto:cyrush@llnl.gov)



# Tutorial Data Acknowledgements

---

## Aneurysm Simulation Dataset

Simulated using the LifeV (<http://www.lifev.org/>) finite element solver.

### Available thanks to:

- Gilles Fourestey and Jean Favre  
Swiss National Supercomputing Centre (<http://www.cscs.ch/>)

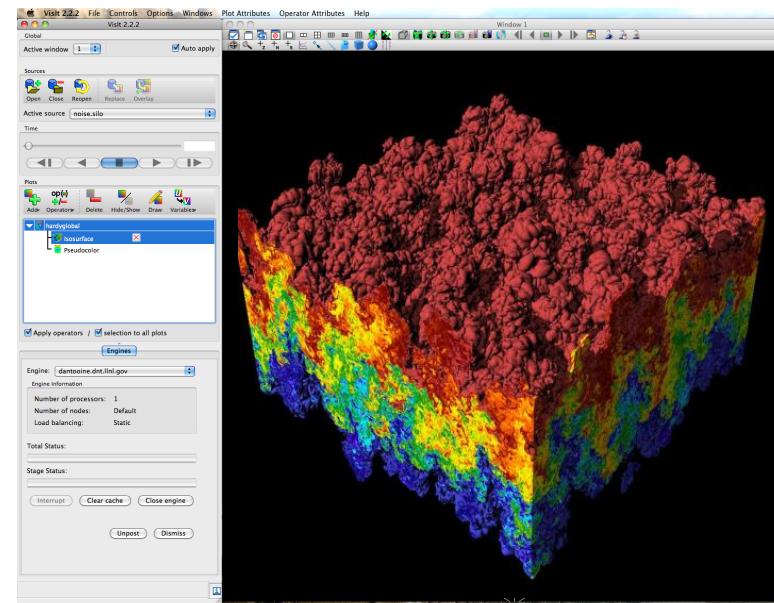


# VisIt Project Introduction



# VisIt is an open source, turnkey application for data analysis and visualization of mesh-based data.

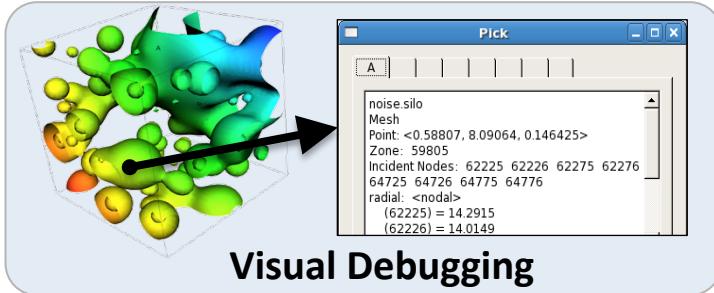
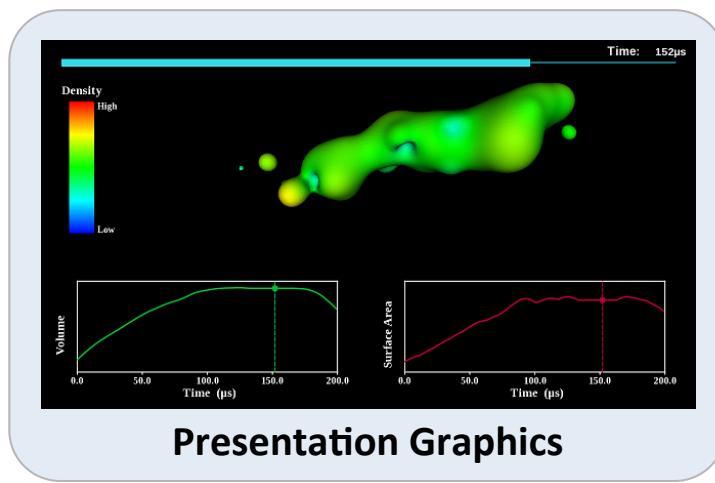
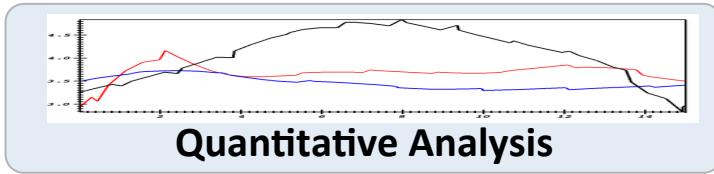
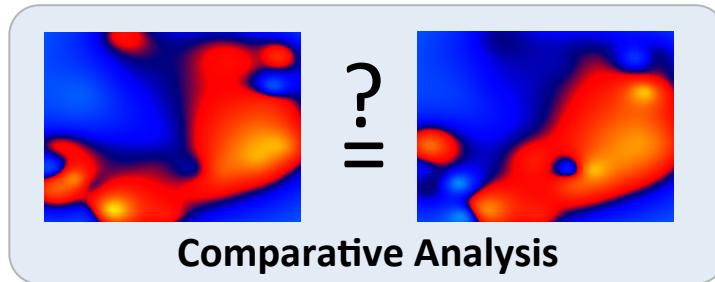
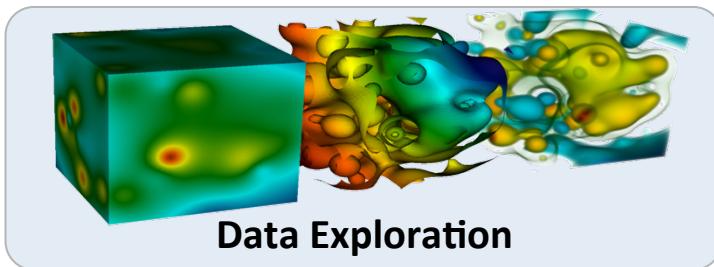
- Production end-user tool supporting scientific and engineering applications.
- Provides an infrastructure for parallel post-processing that scales from desktops to massive HPC clusters.
- Source released under a BSD style license.



**Pseudocolor plot of Density  
(27 billion element dataset)**

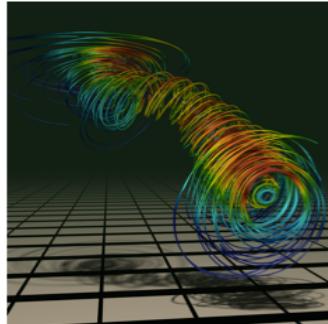


# VisIt supports a wide range of use cases.

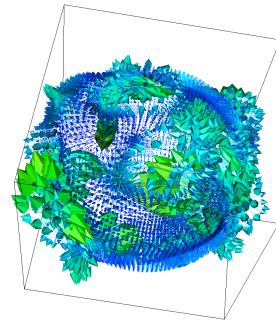




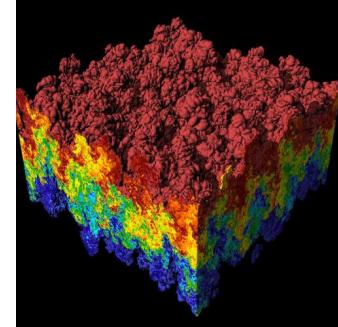
# VisIt provides a wide range of plotting features for simulation data across many scientific domains.



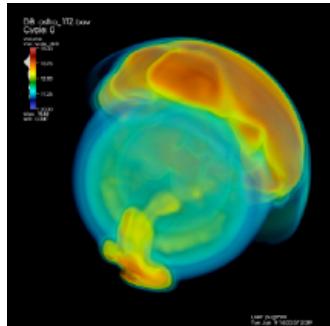
Streamlines / Pathlines



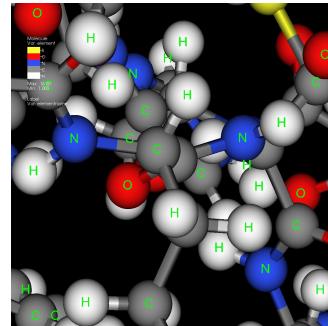
Vector / Tensor Glyphs



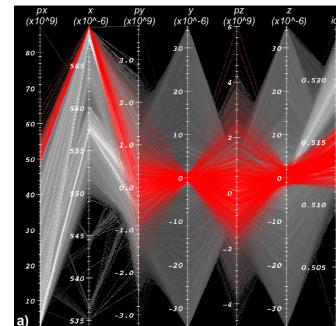
Pseudocolor Rendering



Volume Rendering



Molecular Visualization

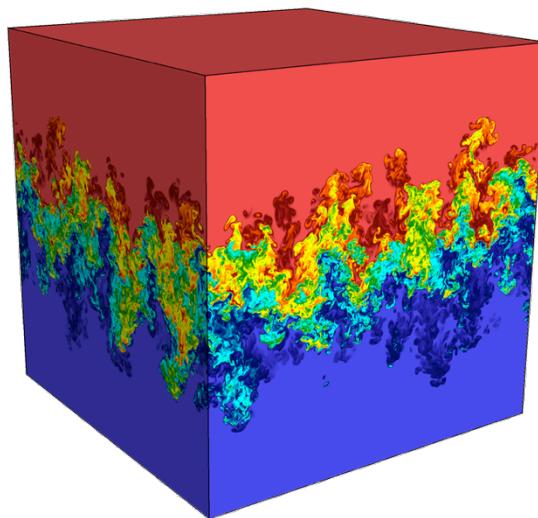


Parallel Coordinates

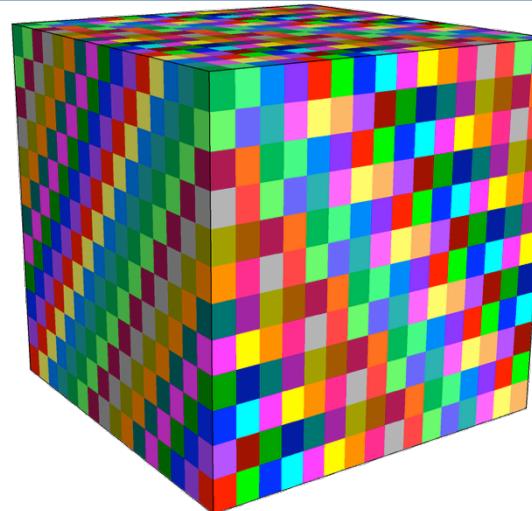




# VisIt uses MPI for distributed-memory parallelism on HPC clusters.



**Full Dataset**  
(27 billion total elements)



**3072 sub-grids**  
(each 192x129x256 cells)

We are enhancing VisIt's pipeline infrastructure to support threaded processing and many-core architectures.



# VisIt is a vibrant project with many participants.

- The VisIt project started in 2000 to support LLNL's large scale ASC physics codes.
- The project grew beyond LLNL and ASC with research and development from DOE SciDAC and other efforts.
- VisIt is now supported by multiple organizations:
  - LLNL, LBNL, ORNL, UC Davis, Univ of Utah, Intelligent Light, ...
- Over 75 person years of effort, 1.5+ million lines of code.

Project Started



2000

LLNL ASC users  
transitioned to  
Visit



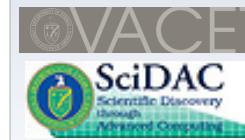
2003

2005 R&D 100



2005

VACET Funded



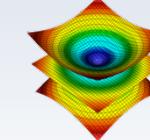
2006

Transition to  
Public SW repo



2008

VisIt 2.0 Release



2010

OSDAV



Intelligent Light

2012 - 2017



Lawrence Livermore National Laboratory

LLNL-PRES-699924



10



# The VisIt team focuses on making a robust, usable product for end users.

## ■ Regular Releases (~ 6 / year)

- Binaries for all major platforms
- End-to-end build process script ``build\_visit''

## ■ User Support and Training

- [visitusers.org](http://visitusers.org), wiki for users and developers
- Email lists: visit-users, visit-developers
- Beginner and advanced tutorials
- VisIt class with detailed exercises

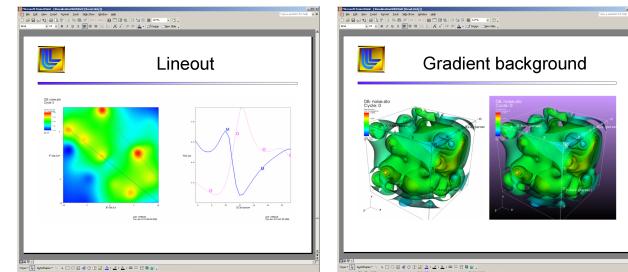
## ■ Documentation

- Getting Data Into VisIt Manual
- Python Interface Manual
- Users Reference Manual

**3D Texturing**  
Like the first volume rendering method, the second method, hardware-accelerated 3D Texturing, resamples the entire database onto a small rectilinear grid. Once the data has been resampled, it is converted into a 3D texture using the Volume plot's volume-transfer function and gets loaded into the video card's texture memory. The Volume plot then draws a set of planes that are perpendicular to the volume's axes. Each plane gets the pre-loaded texture mapped onto it. The resulting image is very crisp and captures details not evident when the splatting method is used.  
1. Change the Rendering Method to 3D Texturing. Click Apply  
2. Change the Number of slices to 800. Click Apply

**Ray Casting**  
The third volume-rendering technique, called ray-casting, used by the Volume plot is not hardware accelerated. In ray-casting, a ray is followed in reverse from the computer screen into the dataset. As a ray progresses through the dataset, sample points are taken and their sample values are used to determine a color and opacity for each point. Each sample point along a ray is composed to form a final color for the ray. The sampling process goes from closest to farthest to allow for early ray termination which stops the sampling process when the pixel opacity gets above a certain threshold. This method of volume-rendering yields superior pictures at the cost of speed and memory use.  
This method requires recalculating each time the view is changed, so you probably want to use Splatting or 3D Texturing to get the view just-right before changing to this method. You will also want to use a smaller window with Ray cast: compositing.

Tutorials on [visitusers.org](http://visitusers.org)



VisIt class materials



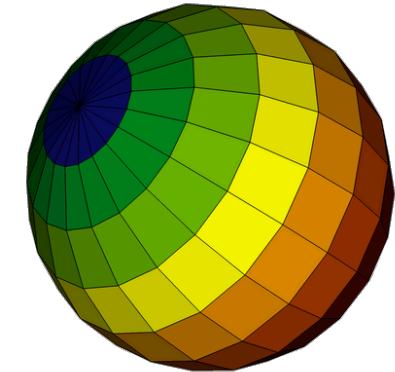
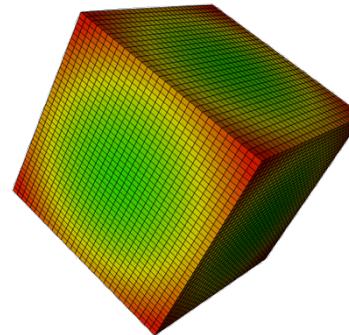
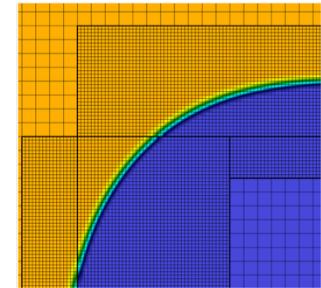
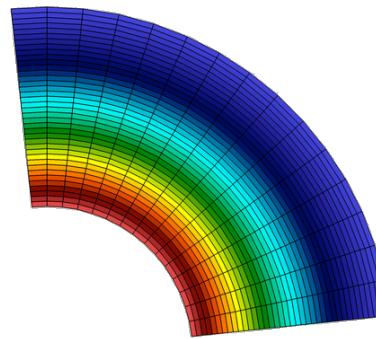
# VisIt provides a flexible data model, suitable for many application domains.

- **Mesh Types**

- Point, Curve, 2D/3D Rectilinear, Curvilinear, Unstructured
- Domain Decomposed, AMR
- Time Varying
- Primarily linear element support, limited quadratic element support

- **Field Types**

- Scalar, Vector, Tensor, Material Volume Fractions, Species





# VisIt supports more than 110 file formats.

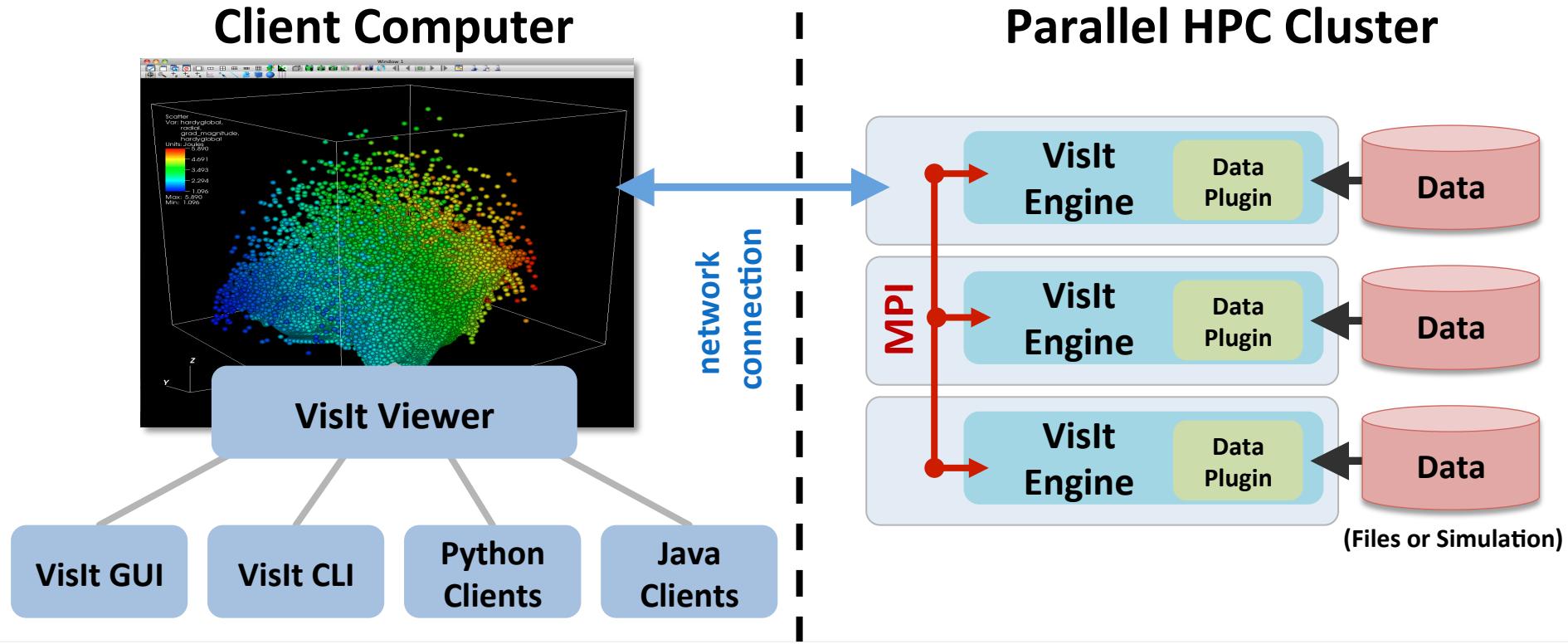
---

## “How do I get my data into VisIt?”

- Experiment with the *visit\_writer* utility:
  - <http://visitusers.org/index.php?title=VisitWriter>
- Write to a commonly used format:
  - VTK, Silo, Xdmf, PVTK
- Consult the [Getting Data Into VisIt Manual](#) and its associated [source code examples](#).

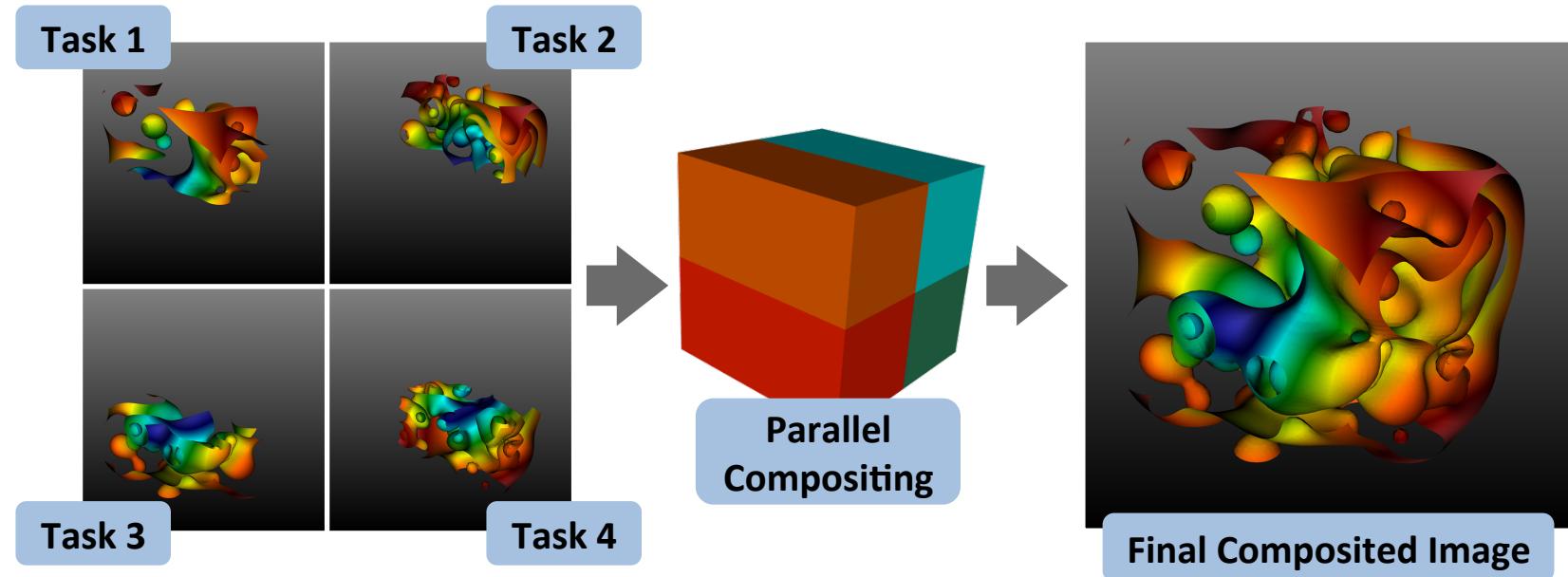


# VisIt employs a parallelized client-server architecture.





# VisIt automatically switches to a scalable rendering mode when plotting large data sets on HPC clusters.



In addition to scalable surface rendering, VisIt also provides scalable volume rendering.



# VisIt's infrastructure provides a flexible platform for custom workflows.

## ■ C++ Plugin Architecture

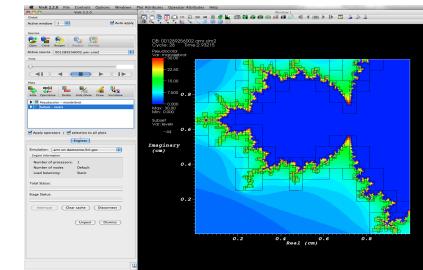
- Custom File formats, Plots, Operators
- Interface for custom GUIs in Python, C++ and Java

## ■ Python Interfaces

- Python scripting and batch processing
- Data analysis via Python Expressions and Queries

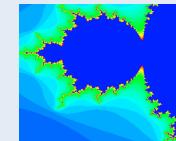
## ■ In-Situ Coupling

- VisIt's *Libsim* library allows simulation codes to link in VisIt's engine for in situ visualization



VisIt

Simulation



*Libsim*  
Adaptor



# VisIt is used as a platform to deploy visualization research.

## ■ DOE ASCR Research Collaborations:



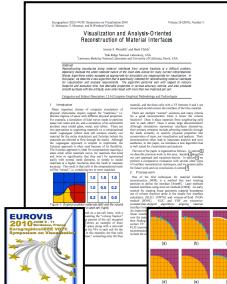
- **Research Focus:**
  - Light weight In Situ Processing
  - Node Level Parallelism
  - Distributed Memory Parallel Algorithms



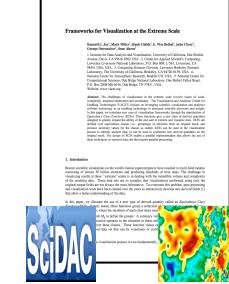
**Scaling research**  
Scaling to 10Ks of cores and trillions of cells.



**Algorithms research**  
How to efficiently calculate particle paths in parallel.



**Algorithms research:**  
Reconstructing material interfaces for visualization



**Methods research:**  
How to incorporate statistics into visualization.





# VisIt is a robust, usable tool, that provides a broad set of visualization capabilities for HPC simulation data.

- **Provides Features that span the “power of visualization”**
  - Data Exploration
  - Confirmation
  - Communication
- **Provides Features for different kinds of users**
  - Visualization Experts
  - Code Developers
  - Code Consumers

VisIt is actively developed and has vibrant developer and user communities.



# Guided Tour of VisIt

- **Materials from:**
  - <http://visitusers.org/index.php?title=VisIt-tutorial-basics>
  - <http://visitusers.org/index.php?title=VisIt-tutorial-data-analysis>
  - <http://visitusers.org/index.php?title=VisIt-tutorial-Python-scripting>



# VisIt's interface is built around five core abstractions.

---

- **Databases:** Read data
- **Plots:** Render data
- **Operators:** Manipulate data
- **Expressions:** Generate derived quantities
- **Queries:** Summarize data



# Aneurysm Simulation Exploration

[http://visitusers.org/index.php?title=Blood\\_Flow\\_Aneurysm\\_Tutorial](http://visitusers.org/index.php?title=Blood_Flow_Aneurysm_Tutorial)



# Additional Hands-on Materials

- **Water Flow Simulation Exploration**
  - [http://visitusers.org/index.php?title=Water\\_Flow\\_Tutorial](http://visitusers.org/index.php?title=Water_Flow_Tutorial)
- **Volume Rendering**
  - <http://visitusers.org/index.php?title=Visit-tutorial-Volume-Rendering>
- **Advanced Movie Making**
  - <http://visitusers.org/index.php?title=Visit-tutorial-Advanced-movie-making>



# Resources

---

- **Presenter Contact Info:**

- Cyrus Harrison: [cyrush@llnl.gov](mailto:cyrush@llnl.gov)

- **User resources:**

- Main website: <http://www.llnl.gov/visit>
  - Wiki: <http://www.visitusers.org>
  - Email: [visitusers@ornl.gov](mailto:visitusers@ornl.gov)

- **Developer resources:**

- Email: [visit-developers@ornl.gov](mailto:visit-developers@ornl.gov)
  - SVN: <http://visitilight.com/svn/visit/>





**Lawrence Livermore  
National Laboratory**