DOE's Open Source Leadership in Scientific Visualization

2018 Salishan Conference on High-Speed Computing

Thursday April 26, 2018

Cyrus Harrison
Computer Scientist
Weapons Simulation and Computing, LLNL

LLNL-PRES-749717
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

Lawrence Livermore National Laboratory

Abstract

To meet the challenge of digesting and understanding massive data sets from HPC physics simulations, DOE's Advanced Simulation and Computing Initiative invested heavily in scientific visualization. These investments gave rise to two widely used and vibrant open source projects: VisIt and ParaView, each with large user bases and impact beyond DOE. This talk will provide perspectives of the trials and successes of the HPC scientific visualization community and how this community is working together to address the challenges from the evolving HPC landscape. It will also dive into how perceptions and use of open source solutions have evolved over the lifetime of these tools.

Thanks!

Thanks to the following for feedback, blessings, and advice:

- Kathleen Biagas (LLNL)
- Eric Brugger (LLNL)
- Hank Childs (University of Oregon)
- Berk Geveci (Kitware)
- Mark Miller (LLNL)
- Ken Moreland (SNL)

A little about me (to anchor my perspectives) …

- I work for LLNL’s Weapons Simulation and Computing (WSC) program supporting scientific visualization and data analysis, most visibly on the VisIt Project

- How did I land here?
  - I joined LLNL in 2005 as a computer scientist and the VisIt team in 2007, initially in a unique role: Completely embedded to support data analysis of large scale multi-physics simulations

    A crude way to describe this: *“Data analysis first and software development second”*

  - Building from this, I grew into more general role on the VisIt team, but I still have the strongest affinity to DOE NNSA mission areas

  - Inherited the VisIt “Architect” role from Hank Childs in 2013

ASCI invested heavily to create scientific visualization capabilities in support of Stockpile Stewardship Science

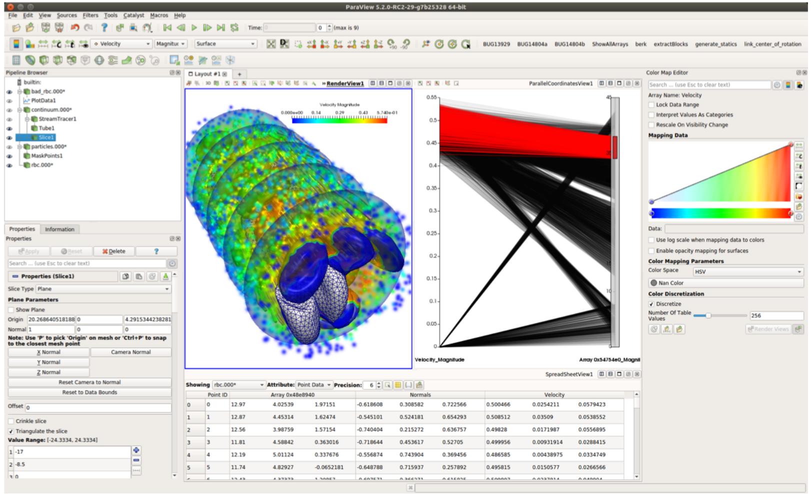
- Display Technologies
  - High Resolution Screens
  - Tiled Displays
- Visualization Clusters
  - Leveraging GPUs for rendering
  - Connections to Displays
- Software
  - In-house, commercial and open source visualization tools


Images from: “A New World of Seeing”
LLNL Science & Technology Review, October 2000
https://str.llnl.gov/str/Quinn.html

This talk focuses on DOE’s success with two open source visualization tools

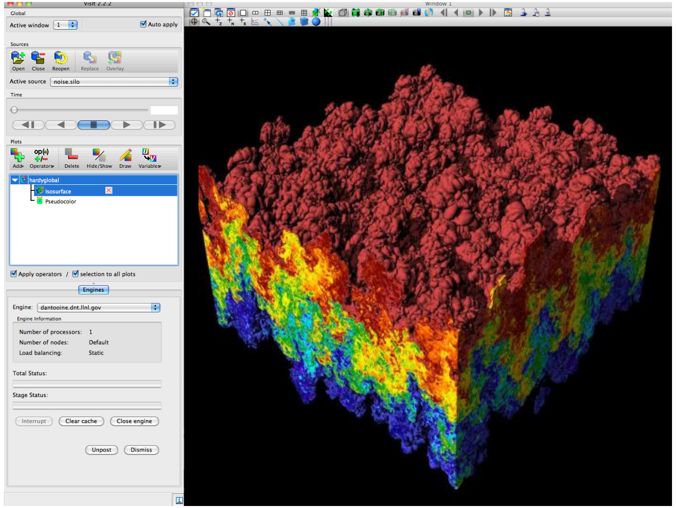
ParaView and VisIt are successful open source scientific visualization tools developed with DOE support

ParaView



https://www.paraview.org

VisIt



https://visit.llnl.gov

These tools have widespread usage in DOE and over a million downloads worldwide

Program documents and retrospectives hint at how the use of open source visualization tools evolved (1996)

Accelerated Strategic Computing Initiative

Program Plan

September 1996

U.S. Department of Energy Defense Programs
Lawrence Livermore National Laboratory
Los Alamos National Laboratory
Sandia National Laboratories

Ensuring That Simulation Results Are Accessible and Understandable

Given the massive amounts of data involved, they will depend on graphically oriented data comprehension applications. These highly flexible applications must (1) allow the designers to directly examine all aspects of the simulation results, (2) provide powerful analytical capabilities with customizable and extensible human-oriented graphical interfaces, and (3) handle the massive amounts of data that will be generated by the ASCI code-platform combination. Such tools do not exist today. Their development will involve efforts in the following areas:

- High-performance visualization systems for data comprehension (including hardware)
- Graphical representation of large, multidimensional data sets

1996 ASCI Program Plan identifies visualization and data analysis gap: *"Such tools do not exist today"*

https://asc.llnl.gov/alliances/alliances_archive/pdf/asci-plan.pdf

Program documents and retrospectives hint at how the use of open source visualization tools evolved (2000)

Accelerated Strategic Computing Initiative (ASCI) Program Plan

Department of Energy

Defense Programs

January 2000

U.S. Department of Energy Defense Programs
Los Alamos National Laboratory
Sandia National Laboratories
Lawrence Livermore National Laboratory

VIEWS:

- Deploy state of the art Data and Visualization Corridors that provide high-resolution powerwall displays (5 million pixels or larger) in shared laboratory environments and high bandwidth data and visualization services in designer offices.
- Deploy, support and enhance common visualization tools (CEI/Ensight, IBM Data Explorer, AVS/Express, MeshTV) as appropriate.
- Develop, deploy and support scalable visualization and rendering tools that allow high performance interaction with extremely large ASCI data sets.
- Develop and deploy a common data model and file format for ASCI scientific data that allows interchange of data between code teams and between laboratories.
- Develop and deploy meta-data management tools that allow weapons designers to create, edit and search meta-data describing created ASCI data sets.

January 2000 ASCI Program plan for VIEWS outlines investments in visualization tools

https://www.osti.gov/servlets/purl/768266

Program documents and retrospectives hint at how the use of open source visualization tools evolved (2000)

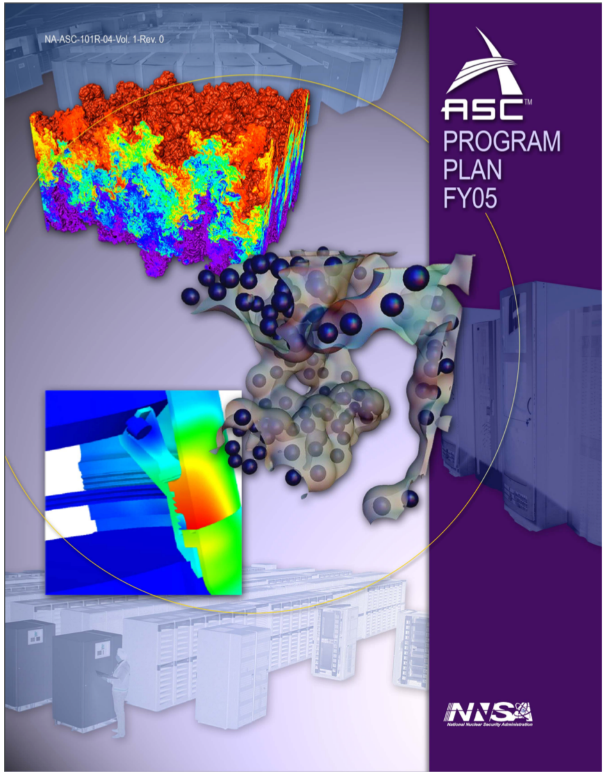
VIEWS:

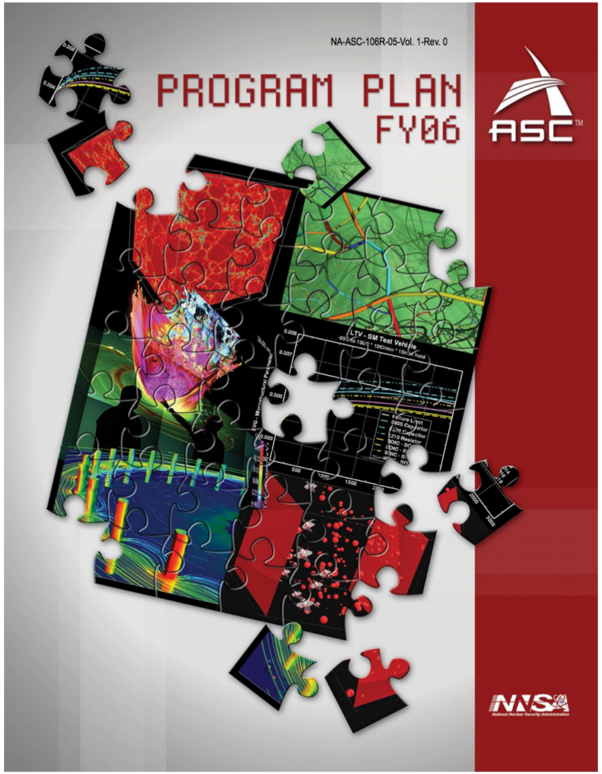
- Deploy state of the art Data and Visualization Corridors that provide high-resolution powerwall displays (5 million pixels or larger) in shared laboratory environments and high bandwidth data and visualization services in designer offices.
- Deploy, support and enhance common visualization tools (CEI/Ensight, IBM Data Explorer, AVS/Express, MeshTV) as appropriate.
- Develop, deploy and support scalable visualization and rendering tools that allow high performance interaction with extremely large ASCI data sets.
- Develop and deploy a common data model and file format for ASCI scientific data that allows interchange of data between code teams and between laboratories.
- Develop and deploy meta-data management tools that allow weapons designers to create, edit and search meta-data describing created ASCI data sets.

January 2000 ASCI Program plan for VIEWS outlines investments in visualization tools

https://www.osti.gov/servlets/purl/768266

Program documents and retrospectives hint at how the use of open source visualization tools evolved (2005, 2006)





Program documents and retrospectives hint at how the use of open source visualization tools evolved (2009)

Delivering Insight

The History of the Accelerated Strategic Computing Initiative (ASCI)

Prepared by: Alex R. Larzelere II

For

Lawrence Livermore National Laboratory

Under Sub-Contract B545072

New technology developed in the commercial marketplace, along with inventions spearheaded by the National Laboratories have significantly changed the world of data handling and visualization over the last decade. ASCI projects were fundamental in enabling these changes.

Three of the projects focused on software for the manipulation and visualization of large datasets. ASCI contracted with a company called CEI (Computational Engineering International) to develop Ensight Gold. This software enables users to manipulate ASCI-scale datasets, render visual images, and then interact with those images in a variety of ways. Ensight was particularly useful for simulations run remotely by LANL staff on the ASCI White computer located at LLNL. Ensight converted the data into many small files in Livermore and then rapidly moved them over secure networks to Los Alamos, where scientists used Ensight Gold and SGI graphic systems to render and display images from the files.

Two other ASCI software tools projects, VisIt and ParaView, also provided important data manipulation capabilities. Both used open-source software as building blocks, such as the Visualization Tool Kit (VTK, from a company named Kitware) to visualize and interact with data. The use of open-source software in these tools was important because it fostered greater engagement with the external community and provided a more robust tool-development effort. Both VisIt and ParaView provide important data manipulation capabilities for ASCI users, and in recognition of its importance, VisIt was awarded an R&D 100 award in 2005.

***Delivering Insight* (a LLNL commissioned history of ASCI) highlights investment and success with:**

Ensight Gold, ParaView and VisIt

https://asc.llnl.gov/asc_history/Delivering_Insight_ASCI.pdf

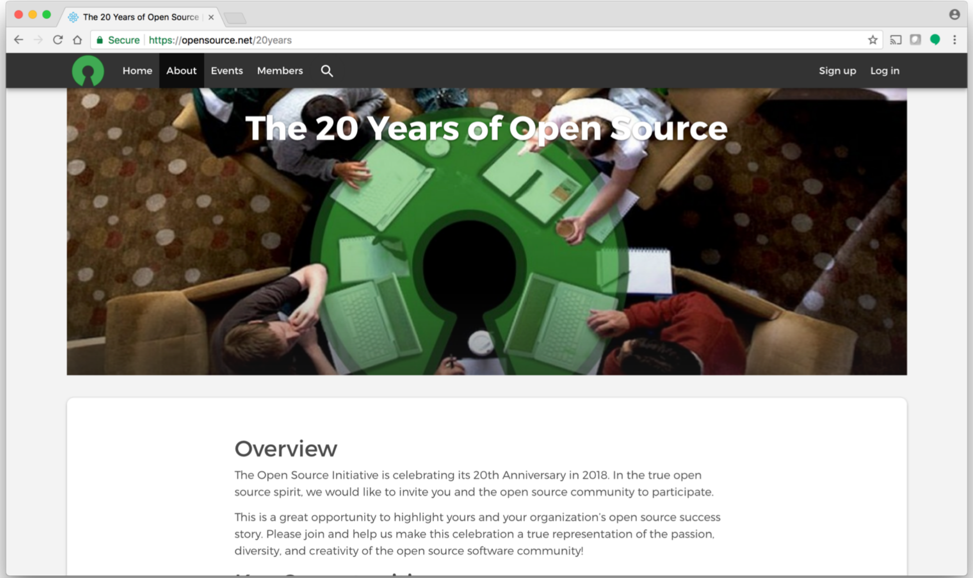
Program documents and retrospectives hint at how the use of open source visualization tools evolved (2009)

Two other ASCI software tools projects, VisIt and ParaView, also provided important data manipulation capabilities. Both used open-source software as building blocks, such as the Visualization Tool Kit (VTK, from a company named Kitware) to visualize and interact with data. The use of open-source software in these tools was important because it fostered greater engagement with the external community and provided a more robust tool-development effort. Both VisIt and ParaView provide important data manipulation capabilities for ASCI users, and in recognition of its importance, VisIt was awarded an R&D 100 award in 2005.

***Delivering Insight* (a LLNL commissioned history of ASCI) highlights investment and success with open source in ParaView and VisIt**

https://asc.llnl.gov/asc_history/Delivering_Insight_ASCI.pdf

The term “Open Source” was not even part of the nomenclature when ASCI started …



The term “Open Source” was coined in 1998

https://opensource.org/history
https://opensource.net/20years

DOE ASC(I) and ASCR have explicitly embraced and promoted open source software development since 2002

Benefits outlined in joint 2003 ASCI and ASCR Memo:

- OSS provides HPC sites the opportunity to identify and fix bugs quickly
- The OSS model yields important contributions to the global state of the art, thus providing significant leverage of Government investments
- The OSS model provides a hedge against “change in support” status for software required to execute the missions of these Programs. This yields protection for the investments made in the software
- Access to source code of OSS can enhance cyber security by facilitating rapid identification and repair of security vulnerabilities

https://science.energy.gov/~/media/ascr/pdf/research/docs/Doe_lab_developed_software_policy.pdf

Open source and open development enable sharing

Sharing amplifies:

- Collaboration
  - Lowers barriers to exchange ideas and code
  - Amortizes development effort
  - Helps establish a developer community

- Adoption
  - More users leads to a better understanding of your software
  - Helps establish a user community

- Competition
  - Reuse and leverage of existing work, but with access to source as stopgap to remedy issues
  - Brings transparency and avoids "secret sauce" myths

Open source and open development enable sharing

Sharing also can exacerbate:

- Priority Inversion
  - Visibility leads to many competing requests and directions

- Squeaky wheel scenarios
  - Open dialog about bugs and requests can asymmetrically overwhelm the kind hearted

Teams need to invest to grow user and developer communities to scale gracefully!

What are these tools used for?

Visualization and analysis tools are key part of the application ecosystem supporting HPC multi-physics simulations

Multi-physics Simulation Applications

CS Infrastructure
- Input Parsing
- Steering
- Communication
- Parallelism Abstractions
- I/O
- In Situ Coupling

Physics Packages
- Hydrodynamics
- Chemistry
- Thermal radiation
- *{and many more …}*

Physics Libraries
- Material Properties
- Material Models

Numerical Libraries
- Linear Algebra
- Finite Elements

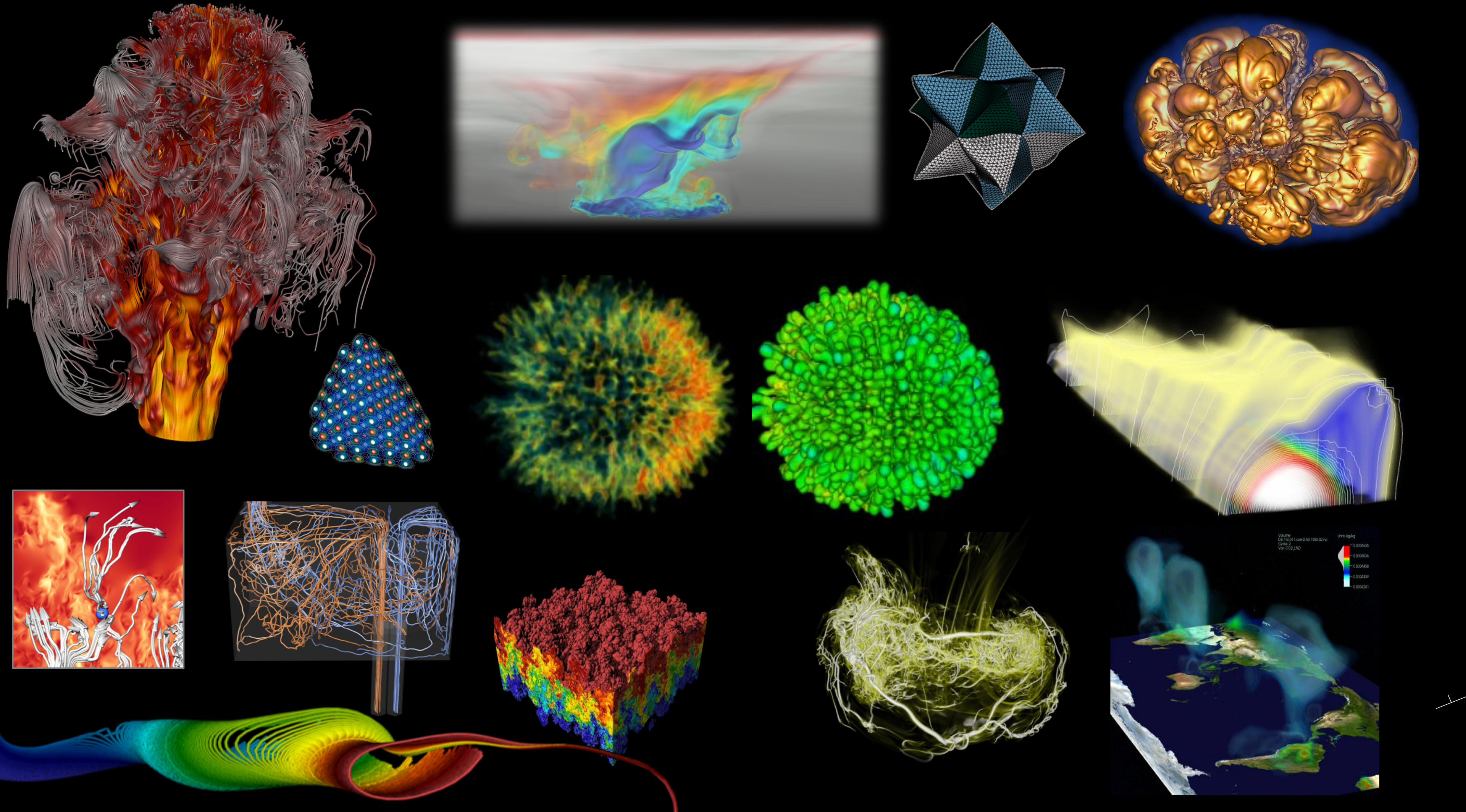
Workflow Applications

Problem Setup
- Computational Geometry
- Mesh Generation
- Mesh Decomposition

Visualization and Analysis
- Mesh Rendering
- Feature Extraction
- Simulated Diagnostics

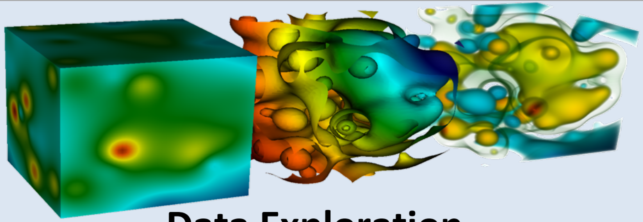
Uncertainty Quantification
- Ensemble Generation
- Parametric Studies
- Statistical Models

Simulation & Data Management
- Workflow Capture
- Data Organization
- Provenance

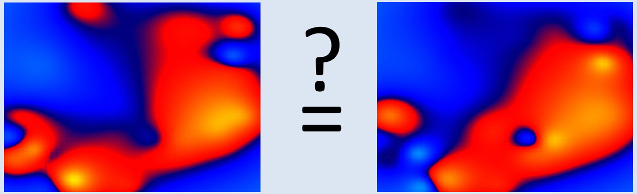


Volume
Cycle: 2
Var: CO2_LND
Units: kg/kg

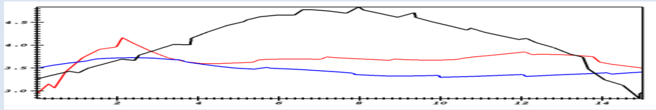
Scientific visualization tools support a wide range of use cases



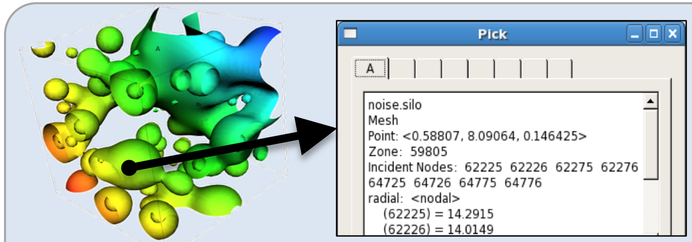
Data Exploration



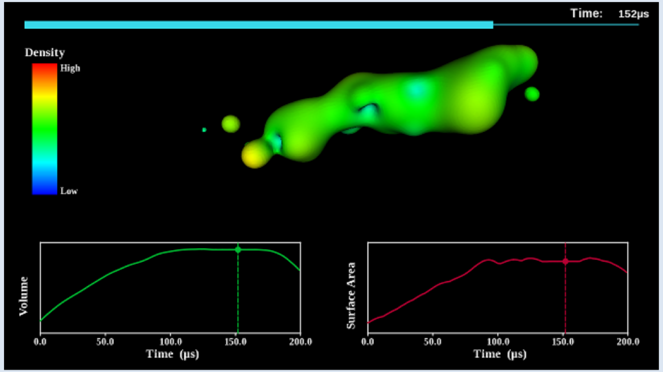
Comparative Analysis



Quantitative Analysis

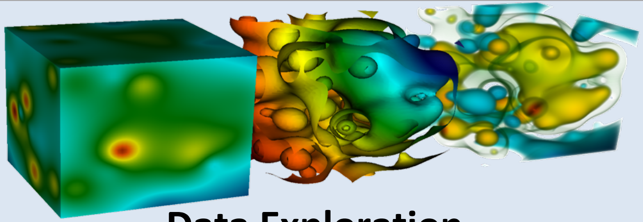


Visual Debugging

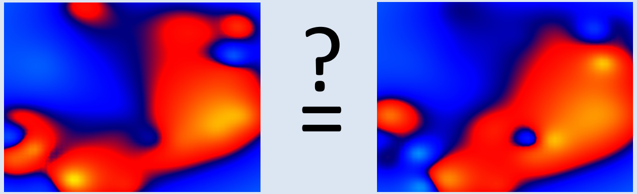


Presentation Graphics

Scientific visualization tools support a wide range of use cases

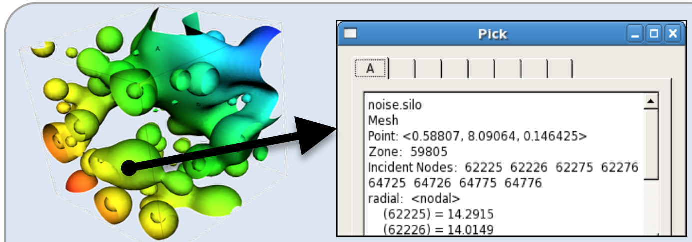


Data Exploration

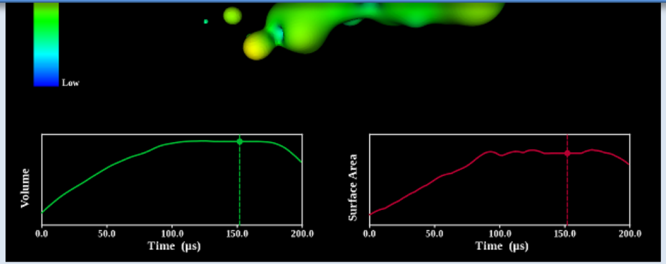


Comparative Analysis

These tools are an important part of the daily workflow used to digest simulation data



Visual Debugging



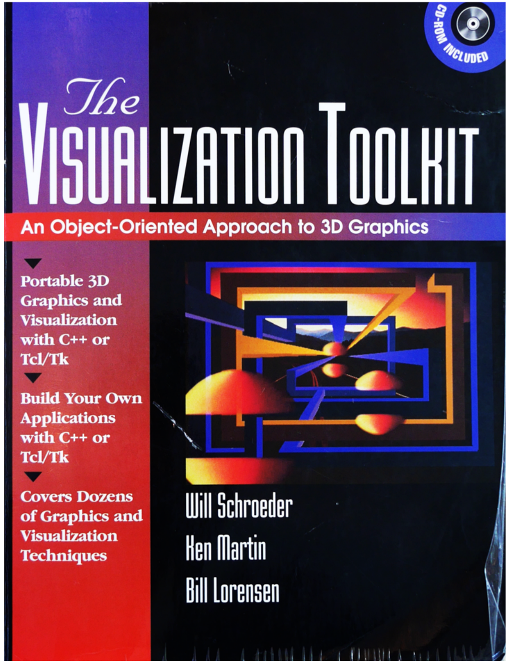
Presentation Graphics

How were these tools created?

VTK was a key substrate that enabled rapid adoption of scientific visualization capabilities

- The software started in support of a book focused on how to do scientific visualization with C++ in 1993

- The success of VTK lead to the founding of Kitware in 1998



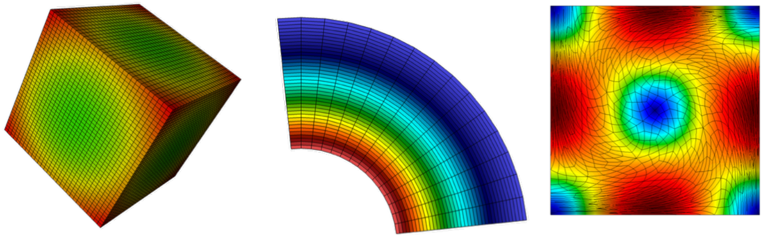
The Visualization Toolkit: An Object-Oriented Approach to 3-D Graphics, W. Schroeder, K. Martin, B, Lorensen, 1996

VTK was a key substrate that enabled rapid adoption of scientific visualization capabilities

- VTK enabled developers to create and share algorithms that apply to a wide range of meshed-based data sets

- VTK provided infrastructure to help compose these algorithms to support complex user operations

- VTK's "toolkit" style design allowed ParaView and VisIt teams to share core infrastructure while still providing flexibility that enabled multiple solutions

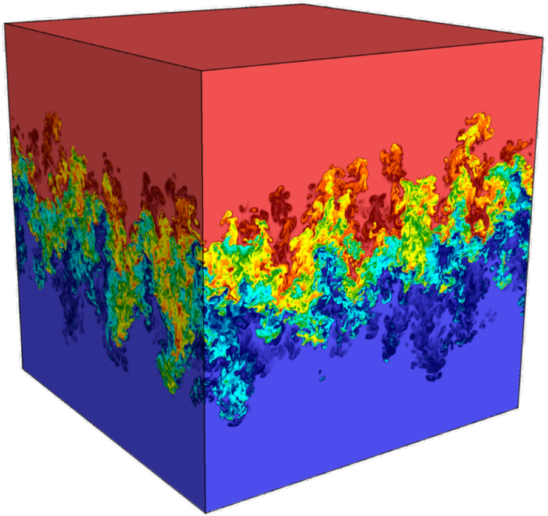


C++ APIs for mesh-based data structures

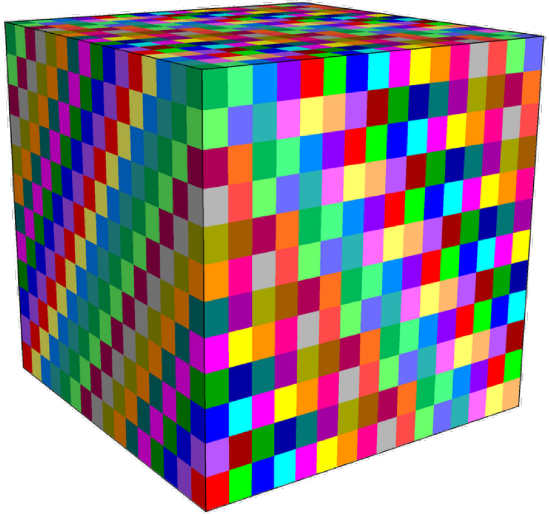
Data Source → **Filter** → **Filter** → **Data Sink**

Data-flow networks process mesh data

Scientific visualization tools need to scale across the compute nodes of a Supercomputer



Full Dataset
(27 billion total elements)



3072 sub-grids
(each 192x129x256 cells)

ASCI invested to build distributed-memory parallel visualization tools

- VTK did not have have distributed-memory parallel capabilities when ASCI started

- ASCI supported research and development addressed this gap

Los Alamos National Laboratory – Technical Report #LAUR-00-1620

A Parallel Approach for Efficiently Visualizing Extremely Large, Time-Varying Datasets

James Ahrens
Los Alamos National Laboratory

Charles Law, Will Schroeder, Ken Martin
Kitware Inc.

Michael Papka
Argonne National Laboratory

Abstract

A significant unsolved problem in scientific visualization is how to efficiently visualize extremely large time-varying datasets. Using parallelism provides a promising solution. One drawback of this approach is the high overhead and specialized knowledge often required to create parallel visualization programs. In this paper, we present a parallel visualization system that is scalable, portable and encapsulates parallel programming details for its users. Our approach was to augment an existing visualization system, the visualization toolkit(VTK). Process and communication abstractions were added in order to support task, pipeline and data parallelism. The resulting system allows users to quickly write parallel visualization programs and avoid rewriting these programs when porting to new platforms. The performance of a collection of parallel visualization programs written using this system and run on both a cluster of SGI Origin 2000s and a Linux-based PC cluster is presented. In addition to showing the utility of our approach, the results offer a comparison of the performance of commodity-based computing clusters.

executes in parallel on different data elements) to achieve improved performance on a long time series.

- **Portability** - Portability is critical to users with access to heterogeneous platforms since platform availability can change due to crashes, maintenance, purchases and removal. If the user's visualization system is portable then they can flexibly choose the best available platform instead of being constrained only to the availability of a specific platform. The system should be portable between platforms with different operating systems and underlying hardware, including between shared and distributed-memory multiprocessors.

- **Full functionality** - The system should support most of the functionality of VTK, offering parallel versions of the algorithms available. Supporting a full range of parallel visualization algorithms is critical to effectively processing large datasets since the alternative, interspersing serial algorithms with parallel algorithms can significantly degrade performance.

2000: J. Ahrens, et al, outline and demonstrate approaches for parallelism in scientific visualization

"A parallel approach for efficiently visualizing extremely large, time-varying datasets" J. Ahrens, et al, LANL Tech Report, 2000
"Large-scale data visualization using parallel data streaming" J. Ahrens, et al, IEEE CGA, 2001

ASCI invested to build distributed-memory parallel visualization tools

In ~1999/2000:

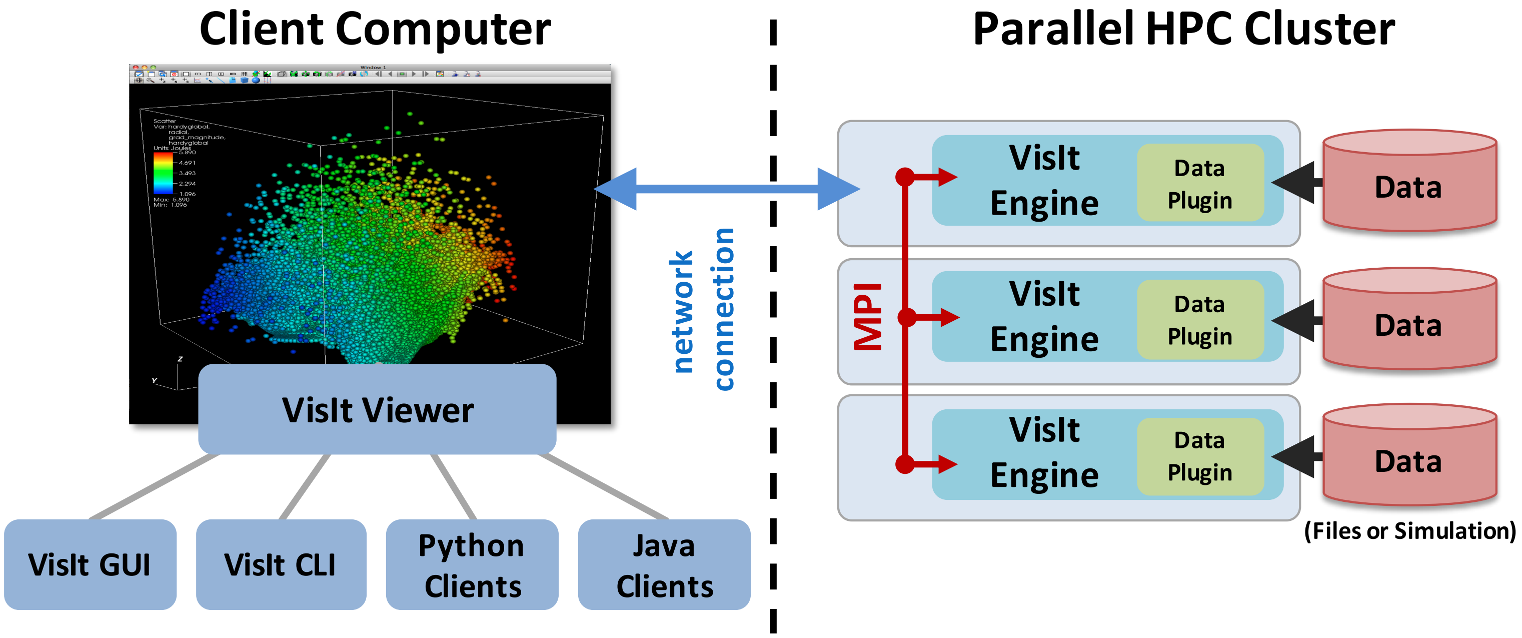
- A LLNL team began building a layer on top of VTK to support distributed-memory parallelism and created VisIt

- Kitware and LANL began to expand VTK to support distributed-memory parallelism and created ParaView

March 2000: ASCI directly supported Kitware's open source development of VTK with a 3 year contract

https://blog.kitware.com/kitware-signs-contract-to-develop-parallel-processing-tools/

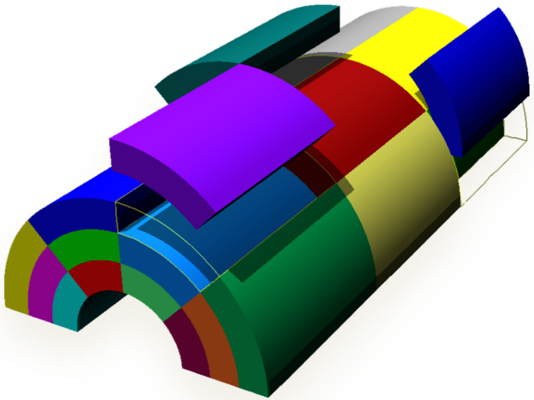
Both VisIt and ParaView employ a parallelized client-server architecture

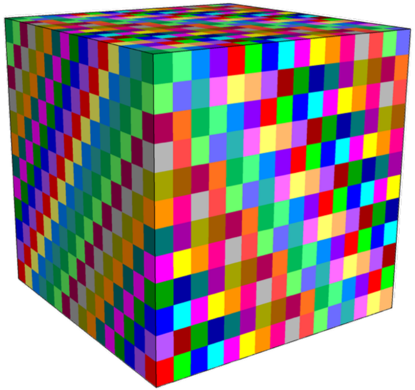


VisIt's Client-Server Architecture

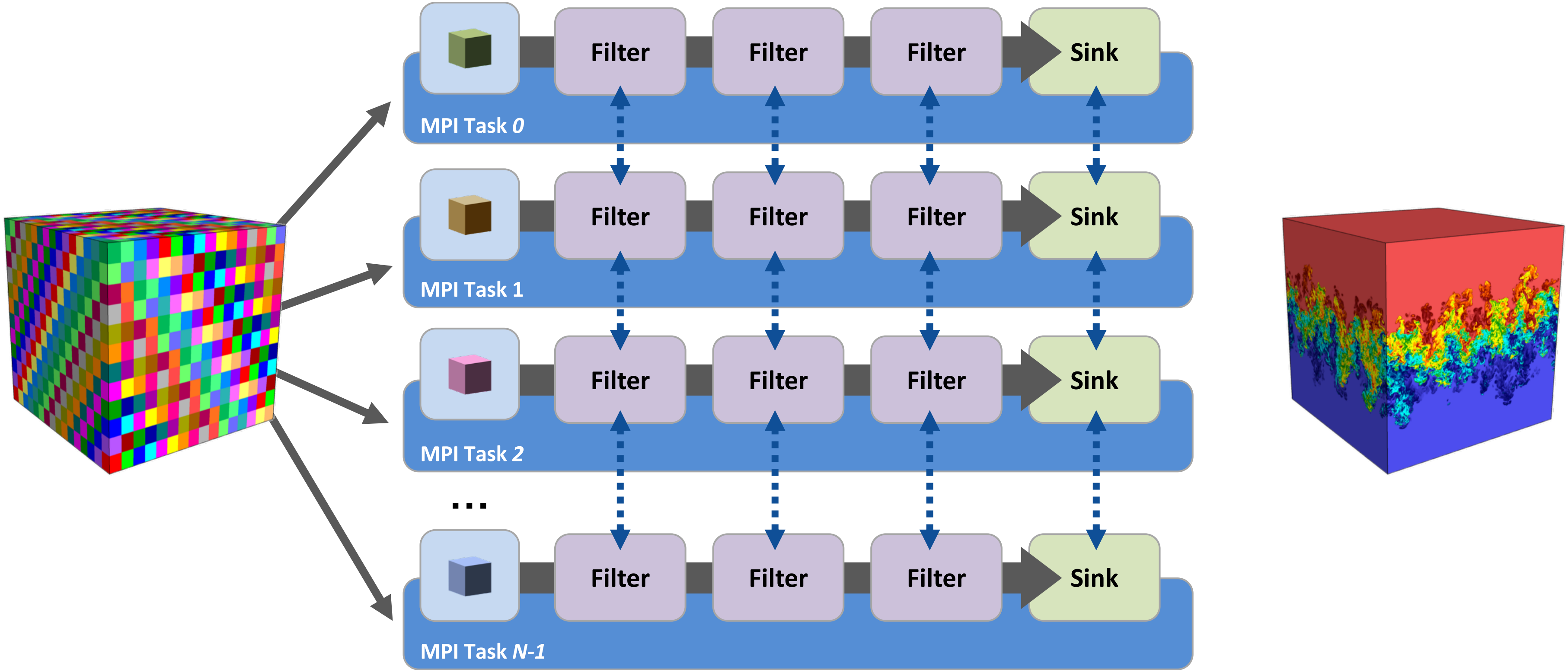
Domain decomposed meshes enable scalable parallel algorithms

- Simulation meshes are divided into smaller mesh **domains**
- Domains are partitioned across MPI tasks for processing
- Algorithms coordinate across MPI tasks when non-local info is needed





These tools use Data-flow networks in a Single Program Multiple Data (SPMD) paradigm to achieve distributed-memory parallelism



VisIt's "Contract" extension to Data-flow networks demonstrated how to flexibly support optimizations

A Contract Based System For Large Data Visualization *

Hank Childs†

University of California, Davis/Lawrence Livermore National Laboratory

Eric Brugger, Kathleen Bonnell, Jeremy Meredith, Mark Miller, and Brad Whitlock ‡

Lawrence Livermore National Laboratory

Nelson Max§

University of California, Davis

Abstract

VisIt is a richly featured visualization tool that is used to visualize some of the largest simulations ever run. The scale of these simulations requires that optimizations are incorporated into every operation VisIt performs. But the set of applicable optimizations that VisIt can perform is dependent on the types of operations being done. Complicating the issue, VisIt has a plugin capability that allows new, unforeseen components to be added, making it even harder to determine which optimizations can be applied.

We introduce the concept of a *contract* to the standard data flow network design. This contract enables each component of the data flow network to modify the set of optimizations used. In addition, the contract allows for new components to be accommodated gracefully within VisIt's data flow network system.

Keywords: large data set visualization, data flow networks, contract-based system

1 Introduction

VisIt is an end-user visualization and data analysis tool for diverse data sets, designed to handle data sets from thousands to millions to billions of elements in a single time step. The tool has a rich feature set; there are many options to subset, transform, render, and query data. VisIt has a distributed design. A server utilizes parallel compute resources for data reduction, while a client runs on a local

The source generates the requested data which becomes input to the first filter. Then *execute* phases propagate down the pipeline. Each component takes the data arriving at its input, performs some operation and creates new data at its output until the sink is reached. These operations are typical of data flow networks. VisIt's data flow network design, is unique, however, in that it also includes a *contract* which travels up the pipeline along with update requests (see Figure 1).

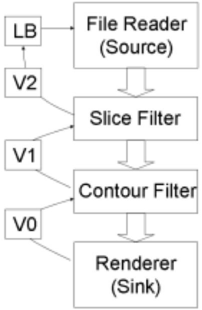
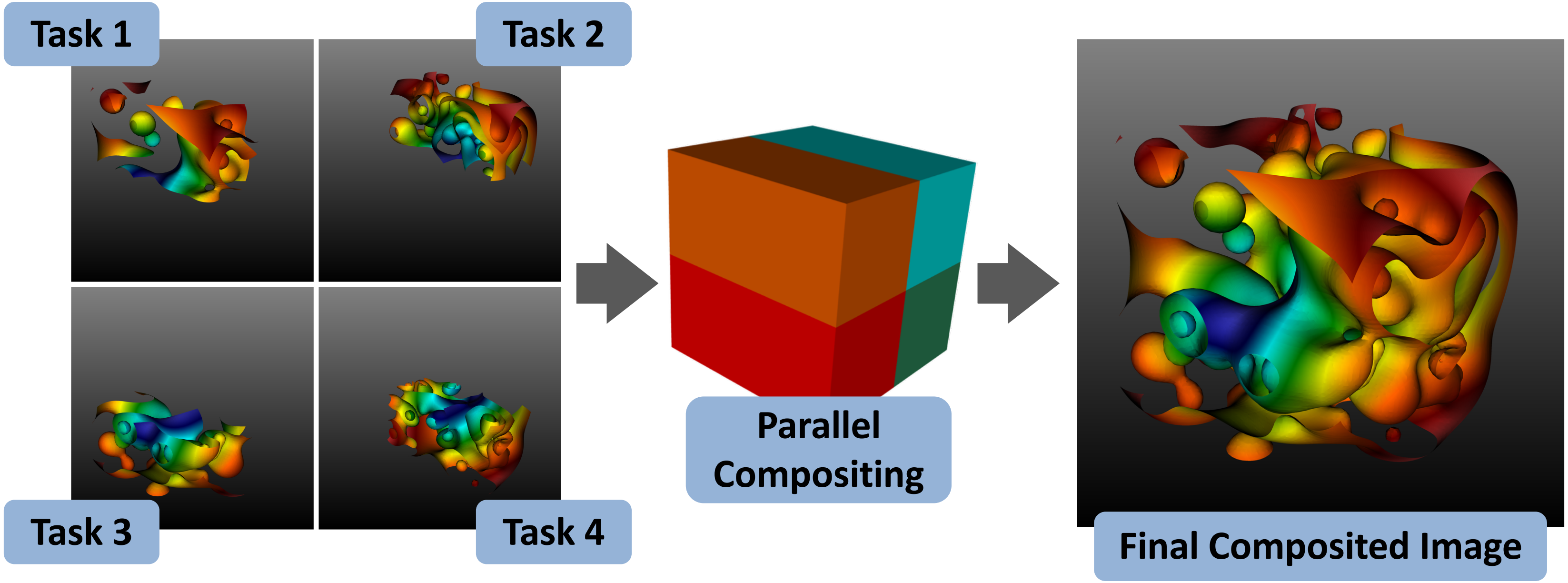


Figure 1: An example pipeline. During the update phase (denoted by thin arrows), Contract Version 0 (V0), comes from the sink. V0 is then an input to the contour filter, which modifies the contract to make Contract Version 1 (V1). This continues up the pipeline, until an executive that contains a load balancer (denoted by LB) is reached. This executive decides the details of the execution phase and passes those details to the source, which begins the execute phase (denoted by thick arrows).

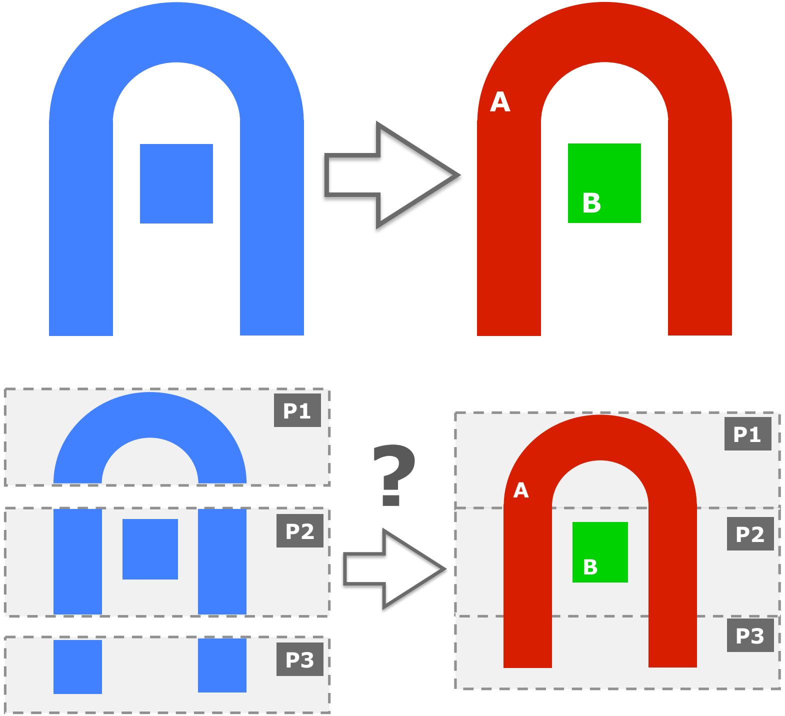
"A Contract-Based System for Large Data Visualization", H. Childs et al, IEEE VIS 2005

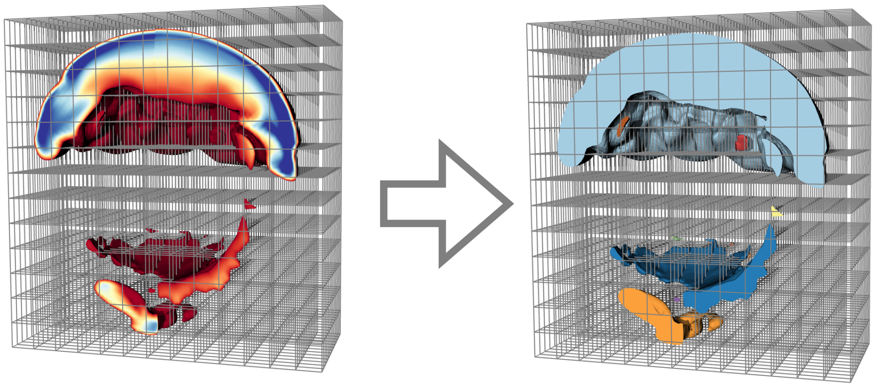
Distributed-memory compositing is the foundation for parallel rendering



ParaView and VisIt both leverage the open source IceT library (http://icet.sandia.gov) for scalable rendering

Distributed-memory mesh connected components for unstructured meshes is a challenging example



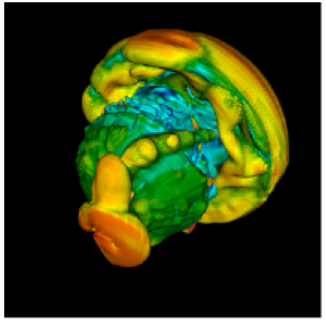


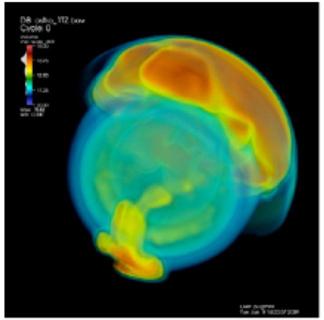
Connected components labeling of an unstructured sub-volume of 21 billion element mesh decomposed across 2197 MPI tasks using VisIt

"Data-Parallel Mesh Connected Components Labeling and Analysis", C. Harrison, H. Childs, K.P. Gaither, EGPV 2011

The VisIt team demonstrated this approach scales to very large data sizes in 2009 and 2010

HPC Cluster	Architecture	Problem Size	# of Cores
Graph (LLNL)	**x86_64**	**$20{,}001^3$ (8 T cells)**	**12K**
Dawn (LLNL)	BG/P	$15{,}871^3$ (4 T cells)	64K
Franklin (NERSC)	Cray XT4	$12{,}596^3$ (2 T cells)	32K
JaguarPF (ORNL)	Cray XT5	$12{,}596^3$ (2 T cells)	32K
Juno (LLNL)	x86_64	$10{,}000^3$ (1 T cells)	16K
Franklin (NERSC)	Cray XT4	$10{,}000^3$ (1 T cells)	16K
Ranger (TACC)	Sun	$10{,}000^3$ (1 T cells)	16K
Purple (LLNL)	IBM P5	$8{,}000^3$ (0.5 T cells)	8K



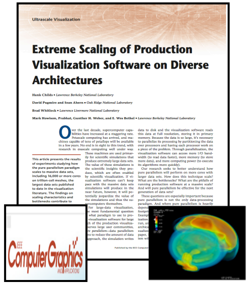


"Extreme Scaling of Production Visualization Software on Diverse Architectures", H. Childs et al, IEEE CGA, 2010 May-June

Synthetic data allowed testing data sizes well beyond simulation requirements

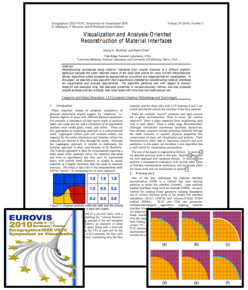
Open source code and open development helped these tools grow vibrant user and developer communities

- These communities helped drive wide adoption in DOE and beyond
- BSD style (non-copyleft) licenses have helped enable commercialization
- These tools are also used as vehicles to deploy capabilities from visualization research to users
  - Helps address the R&D adoption "valley of death"



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

DOE’s visualization community is collaborating to create open source tools ready for Exascale simulation data

Addressing node-level parallelism

- VTK-m is an effort to provide a toolkit of visualization algorithms that leverage emerging node-level HPC architectures

- We are also exploring using VTK-m and DIY to share more distributed-memory infrastructure across projects

VTK-m DIY

http://m.vtk.org https://github.com/diatomic/diy

Addressing I/O gaps with in-situ

- There are several efforts focused on in-situ infrastructure and algorithms

ALPINE (ParaView/VisIt) http://alpine.dsscale.org

ParaView Catalyst http://www.paraview.org/in-situ

VisIt LibSim https://visit.llnl.gov

SENSEI in situ http://www.sensei-insitu.org

Ascent https://github.com/Alpine-DAV/ascent

Looking forward, I see two other high impact opportunities

Simplifying in-memory coupling

- Fluid in-memory exchange between tools opens up many possibilities

- However, in-memory integration is a more intimate vs file-based exchange and developing conventions for sharing data across the ecosystem is hard

Bridging analysis ecosystems

- Data analysis ecosystems are growing explosively, have huge mindshare and powerful capabilities

- There is tremendous opportunity for leverage, but to get traction we need to lower barriers to connect data between ecosystems

These present big data representation and software engineering challenges

Embracing open source software was a key part in establishing DOE's leadership in HPC scientific visualization

Open source and open development helped:

- Build robust, widely used tools
- Scale user and developer communities
- Unlock funding beyond ASCI, including commercial endeavors
- The broader DOE visualization community collaborate and deploy research

Lawrence Livermore National Laboratory