Visualization and Analysis of Massive Data with VisIt

ATPESC 2014
Argonne Training Program on Extreme-Scale Computing
Monday August 11, 2014

Cyrus Harrison
Lawrence Livermore National Laboratory
cyrush@llnl.gov
Tutorial Outline

- VisIt Project Intro [5 min]
- Guided tour of VisIt [25 min]
- Hands on with an Aneurysm Simulation [30 min]
Tutorial Resources

- Tutorial Prep:

- Example Datasets:
  http://visitusers.org/index.php?title=Tutorial_Data

- Blood Flow Hands-on:

- More Tutorial Materials (From past SC Tutorials):
  http://visitusers.org/index.php?title=VisIt_Tutorial

- Cyrus’ Email: cyrush@llnl.gov
Aneurysm Simulation Data

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.
VisIt supports a wide range of use cases.

- Data Exploration
- Comparative Analysis
- Quantitative Analysis
- Visual Debugging
- Presentation Graphics
Examples of VisIt’s visualization capabilities.

- Streamlines
- 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 cells)

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

We are enhancing VisIt’s pipeline infrastructure to also support threaded processing.
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, Univ of Oregon, Univ of Utah, Intelligent Light, …
- Over 75 person years of effort, 1.5+ million lines of code.
VisIt scales well on current HPC platforms.

<table>
<thead>
<tr>
<th>Machine</th>
<th>Architecture</th>
<th>Problem Size</th>
<th># of Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph</td>
<td>X86_64</td>
<td>$20,001^3$ (8 T cells)</td>
<td>12K</td>
</tr>
<tr>
<td>Dawn</td>
<td>BG/P</td>
<td>$15,871^3$ (4 T cells)</td>
<td>64K</td>
</tr>
<tr>
<td>Franklin</td>
<td>Cray XT4</td>
<td>$12,596^3$ (2 T cells)</td>
<td>32K</td>
</tr>
<tr>
<td>JaguarPF</td>
<td>Cray XT5</td>
<td>$12,596^3$ (2 T cells)</td>
<td>32K</td>
</tr>
<tr>
<td>Juno</td>
<td>X86_64</td>
<td>$10,000^3$ (1 T cells)</td>
<td>16K</td>
</tr>
<tr>
<td>Franklin</td>
<td>Cray XT4</td>
<td>$10,000^3$ (1 T cells)</td>
<td>16K</td>
</tr>
<tr>
<td>Ranger</td>
<td>Sun</td>
<td>$10,000^3$ (1 T cells)</td>
<td>16K</td>
</tr>
<tr>
<td>Purple</td>
<td>IBM P5</td>
<td>$8,000^3$ (0.5 T cells)</td>
<td>8K</td>
</tr>
</tbody>
</table>

Scaling Studies of Isosurface Extraction and Volume Rendering (2009)

VisIt is also used daily by domain scientists.
The VisIt team focuses on making a robust, usable product for end users.

- Regular releases (~ 6 / year)
  - Executables for all major platforms
  - End-to-end build process script `build_visit`

- Customer Support and Training
  - 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
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

- **Fields:**
  - Scalar, Vector, Tensor, Material volume fractions, Species

**VisIt currently supports over 110 file formats.**
VisIt employs a parallelized client-server architecture.

**Local Components**
- VisIt GUI
- VisIt CLI
- Python Clients
- Java Clients

**Parallel Cluster**
- VisIt Engine
- Data Plugin
- MPI

**Data Flow Network**
- Filter
- (Files or Simulation)

**Network Connection**
- Data
- Data
- Data
VisIt automatically switches to a scalable rendering mode for large data sets.

- Rendering Modes:
  - Local (hardware)
  - Remote (software or hardware)

- Beyond surfaces:
  - VisIt also provides scalable volume rendering.
**Goal:** Identify and track coherent vortical structures in turbulent flow as time evolves.

Collaboration with Kelly Gaither, TACC et al (IEEE CG&A July/August 2012)

VisIt was used to calculate isosurfaces, identify connected components, and extract component features.
Analysis Example: Turbulence Operators

- **Goal:** Provide one set of turbulence tools that can be used across multiple codes.
- **Application:** Validate RANS model parameters from high fidelity DNS simulations.

Joint work with Oleg Schilling and Britton Olson, LLNL

We are developing scripted building blocks for flow analysis, including field means and fluctuations.
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.

- **Libsim library**
  - Enables coupling of simulation codes to VisIt for in situ visualization.
VisIt is used as a platform to deploy visualization research.

- **Research Collaborations:**
  - 2006 – 2011
  - 2012 – 2017

- **Research Focus:**
  - Next Generation Architectures
  - Parallel Algorithms
  - In-Situ Processing

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

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

- **Methods research:**
  - How to incorporate statistics into visualization.
VisIt: What’s the Big Deal?

- Everything works at scale
- Robust, usable tool
- Features that span the “power of visualization”:
  - Data Exploration
  - Confirmation
  - Communication
- Features for different kinds of users:
  - Visualization Experts
  - Code Developers
  - Code Consumers

Healthy future: Vibrant Developer and User Communities
Resources

- **User resources:**
  - Main website: [http://www.llnl.gov/visit](http://www.llnl.gov/visit)
  - Wiki: [http://www.visitusers.org](http://www.visitusers.org)
  - Email: visitusers@ornl.gov

- **Development resources:**
  - Email: visit-developers@ornl.gov
  - SVN: [http://portal.nersc.gov/svn/visit](http://portal.nersc.gov/svn/visit)
Hands On Visualizations
30 minute Hands on visualization of a Blood Flow Simulation.

In-depth hands on visualization of a Water Flow Simulation.

Materials for Hand-on session after dinner:

[Supporting Slides]
VisIt’s Core Abstractions
VisIt’s core abstractions

- **Databases**: How datasets are read
- **Plots**: How you render data
- **Operators**: How you manipulate data
- **Expressions**: Mechanism for generating derived quantities
- **Queries**: How to access quantitative information
Examples of VisIt Pipelines

- Databases: how you read data
- Plots: how you render data
- Operators: how you transform/manipulate data
- Expressions: how you create new fields
- Queries: how you pull out quantitative information

![Diagram of VisIt Pipelines]

- Open a database, which reads from a file (example: open file1.hdf5)
- Make a plot of a variable in the database (example: Volume plot)
Examples of VisIt Pipelines

- Databases: how you read data
- Plots: how you render data
- Operators: how you transform/manipulate data
- Expressions: how you create new fields
- Queries: how you pull out quantitative information

Example VisIt Pipelines:

1. Open a database, which reads from a file (example: open file1.hdf5)
2. Apply an operator to transform the data (example: Slice operator)
3. Plot a variable in the database (example: Pseudocolor plot)
Examples of VisIt Pipelines

- **Databases:** how you read data
- **Plots:** how you render data
- **Operators:** how you transform/manipulate data
- **Expressions:** how you create new fields
- **Queries:** how you pull out quantitative information

**Example VisIt Pipelines**

1. **Database**
   - Open a database, which reads from a file (example: open file1.hdf5)

2. **Operator 1**
   - Apply an operator to transform the data (example: Slice operator)

3. **Operator 2**
   - Apply a second operator to transform the data (example: Elevate operator)

4. **Plot**
   - Plot a variable in the database (example: Pseudocolor plot)
Examples of VisIt Pipelines

- **Databases**: how you read data
- **Plots**: how you render data
- **Operators**: how you transform/manipulate data
- **Expressions**: how you create new fields
- **Queries**: how you pull out quantitative information

**Diagram**:

- **Database**: Open a database, which reads from a file (example: open file1.hdf5)
- **Expression**: Create derived quantities from fields in the file (example: magnitude(velocity))
- **Plot**: Plot the expression variable (example: Pseudocolor plot)
Examples of VisIt Pipelines

- **Databases**: how you read data
- **Plots**: how you render data
- **Operators**: how you transform/manipulate data
- **Expressions**: how you create new fields
- **Queries**: how you pull out quantitative information

**Database**
- Open a database, which reads from a file (example: open file1.hdf5)

**Plot**
- Plot a field from the file (example: density + Pseudocolor plot)

**Query**
- Extract quantitative information (example: integrate density to find mass)
Examples of VisIt Pipelines

- **Databases**: how you read data
  - Open a database, which reads from a file (example: open file1.hdf5)

- **Plots**: how you render data
  - Create derived quantities from fields in the file (example: magnitude(velocity))
  - Plot a field (example: speed + Pseudocolor plot)
  - Extract quantitative information (example: maximum speed over cross-section)

- **Operators**: how you transform/manipulate data
  - Apply an operator to transform the data (example: Slice operator)
  - Apply a second operator to transform the data (example: Elevate operator)

- **Expressions**: how you create new fields
  - Expression

- **Queries**: how you pull out quantitative information
  - Query
Visualization Techniques for Mesh-based Simulations
Terminology

- **Meshes**: discretization of physical space
  - Contains “zones” / “cells” / “elements”
  - Contains “nodes” / “points” / “vertices”
    - VisIt speak: zone & node

- **Fields**: variables stored on a mesh
  - **Scalar**: 1 value per zone/node
    - Example: pressure, density, temperature
  - **Vector**: 3 values per zone/node (direction)
    - Example: velocity
      - Note: 2 values for 2D, 3 values for 3D
  - More fields discussed later…
Pseudocolor

- Maps scalar fields (e.g., density, pressure, temperature) to colors.
Contour / Isosurface
Volume rendering

Emitter → Film/image

Plotting Techniques
Particle advection: the foundation of flow visualization

- Displace massless particle based on velocity field
- \( S(t) = \) position of curve at time \( t \)
  - \( S(t_0) = p_0 \)
    - \( t_0 \): initial time
    - \( p_0 \): initial position
  - \( S'(t) = v(t, S(t)) \)
    - \( v(t, p) \): velocity at time \( t \) and position \( p \)
    - \( S'(t) \): derivative of the integral curve at time \( t \)

This is an ordinary differential equation
Streamlines
Meshes

- All data in VisIt lives on a mesh
- Discretizes space into points and cells
  - (1D, 2D, 3D) + time
  - Mesh dimension need not match spatial dimension (*e.g.* 2D surface in 3D space)
- Provides a place for data to be located
- Defines how data is interpolated
Scalors, Vectors, Tensors

Associated with points or cells of a mesh
• Points: linear interpolation
• Cells: piecewise constant

Can have different dimensionality than the mesh (e.g. 3D vector data on a 2D mesh)
Materials

- Describes disjoint spatial regions at a sub-grid level
- Volume/area fractions
- VisIt will do high-quality sub-grid material interface reconstruction
Species

- Similar to materials, describes sub-grid variable composition
  - Example: *Material “Air” is made of species “N₂”, “O₂”, “Ar”, “CO₂”, etc.*

- Used for mass fractions

- Generally used to weight other scalars (e.g. partial pressure)
Parallel Meshes

- Provides aggregation for meshes
- A mesh may be composed of large numbers of mesh “blocks”
- Allows data parallelism
AMR meshes

- Mesh blocks can be associated with patches and levels
- Allows for aggregation of meshes into AMR hierarchy levels
Practical Tips for Using VisIt
Practical Tips for Using VisIt

- How to get VisIt to read your data
- How to get help when you run into trouble
How to get VisIt to read your data.

- There is an extensive manual on this topic: “Getting Data Into VisIt”

https://wci.llnl.gov/codes/visit/manuals.html

- Three ways:
  - Use a known format
  - Write a file format reader
  - In situ processing
File formats that VisIt supports

- **110+ Total Readers:** ADIOS, BOV, Boxlib, CCM, CGNS, Chombo, CLAW, EnSight, ENZO, Exodus, FLASH, Fluent, GDAL, Gadget, Images (TIFF, PNG, etc), ITAPS/MOAB, LAMMPS, NASTRAN, NETCDF, Nek5000, OpenFOAM, PLOT3D, PlainText, Pixie, Shapefile, Silo, Tecplot, VTK, Xdmf, Vs, and many more


- Some readers are more robust than others.
  - For some formats, support is limited to flavors of a file a VisIt developer has encountered previously (e.g. Tecplot).
File formats that VisIt supports

- **110+ Total Readers:** ADIOS, BOV, Boxlib, CCM, CGNS, Chombo, Exodus, FLASHPIT, PLOT3D, PlainText, Tecplot, VTK, Xdmf, and many more.

- Some readers are more robust than others.
- For some formats, support is limited to flavors of a file a VisIt developer has encountered previously (e.g. Tecplot).

File formats that VisIt supports

- Common array writing libraries:
  - NETCDF
    - VisIt reader understands many (but not all) conventions
  - HDF5
    - Pixie is most general HDF5 reader
    - Many other HDF5 readers
- Xdmf: specify an XML file that describes semantics of arrays in HDF5 file
- VizSchema (Vs): add attributes to your HDF5 file that describes semantics of the arrays.
VTK File Format

- The VTK file format has both ASCII and binary variants.

- Easiest way to write VTK files: use VTK modules
  - ... but this creates a dependence on the VTK library

- You can also try to write them yourself, but this is an error prone process.

- Third option: visit_writer
VisIt Writer writes VTK files

- It is a “library” (actually a single C file) that writes VTK-compliant files.
  - The typical path is to link visit_writer into your code and write VTK files

- There is also Python binding for visit_writer.
  - The typical path is to write a Python program that converts from your format to VTK

- Both options are short term: they allow you to play with VisIt on your data. If you like VisIt, then you typically formulate a long term file format strategy.

- More information on visit_writer:
```python
import visit_writer
import math
import sys

nX = 20
nY = 20
conn = []
for i in range(nX-1):
    for j in range(nY-1):
        pt1 = j*(nX) + i;
        pt2 = j*(nX) + i+1;
        pt3 = (j+1)*(nX) + i+1;
        pt4 = (j+1)*(nX) + i;
        conn.append(["quad", pt1, pt2, pt3, pt4])

pts = []
rad = []
for i in range(nX):
    for j in range(nY):
        pts.extend([[float(i), float(j), 0]])
        rad.append(math.sqrt(i*i + j*j))

var_datum = ["radius", 1, 1, rad]
vars = [var_datum]
visit_writer.WriteUnstructuredMesh("ugrid.vtk", 0, pts, conn, vars)
sys.exit()
```
Silo file format

- Silo is a mature, self-describing file format that deals with multi-block data.
- It has drivers on top of HDF5 and “PDB”.
- Fairly rich data model
- More information:
  - https://wci.llnl.gov/codes/silo/
Welcome to Silo
A mesh and field I/O library and scientific database

- Structured Rectilinear Mesh
- Gridless Point Mesh
- Structured (Curvilinear) Mesh
- Arbitrary Subsets
- Silex browser for Silo files
- Constructive Solid Geometry (CSG) Mesh
- Unstructured Zoo (UCD) Mesh
- Adaptive Mesh Refinement (AMR) Mesh
- Mixing Materials
- Arbitrary Polyhedral Mesh
- XY Curve
How to get VisIt to read your data.

- There is an extensive manual on this topic: “Getting Data Into VisIt”
  
  https://wci.llnl.gov/codes/visit/manuals.html

- Three ways:
  - Use a known format
  - Write a file format reader
  - In-situ processing

These topics are covered in the manual
Practical Tips for Using VisIt

- How to get VisIt to read your data
- How to get help when you run into trouble
How to get help when you run into trouble

- **FAQ**
  - [http://visit.llnl.gov/FAQ.html](http://visit.llnl.gov/FAQ.html)

- **VisIt Users Mailing List**
  - Address: [visit-users@elist.ornl.gov](mailto:visit-users@elist.ornl.gov)
  - Info: [https://elist.ornl.gov/mailman/listinfo/visit-users](https://elist.ornl.gov/mailman/listinfo/visit-users)
  - Archive: [https://elist.ornl.gov/pipermail/visit-users/](https://elist.ornl.gov/pipermail/visit-users/)

- **VisIt Users Wiki**
  - [http://www.visitusers.org](http://www.visitusers.org)

- **VisIt Users Forum**
  - [http://visitusers.org/forum/YaBB.pl](http://visitusers.org/forum/YaBB.pl)

- **Priority support for specific user groups:**
  - VisIt-help-{XYZ} Mailing Lists

- **Reference Manuals**
  - [https://wci.llnl.gov/codes/visit/manuals.html](https://wci.llnl.gov/codes/visit/manuals.html)
FAQ: http://visit.llnl.gov/FAQ.html

Frequently Asked Questions

1. Contact information
2. Supported platforms
3. Optimal hardware/software
4. Debugging problems starting VisIt or opening files
5. Stereo rendering
6. VisIt won’t run on Linux
7. Slow performance on Linux
8. Slow performance using SSH
9. No output in visualization window
10. Accessing data on remote machine
11. Running VisIt in parallel
12. Supported data file formats
13. Getting your data into VisIt
14. Making a movie of your data
15. Setting your user name to connect to a remote machine
16. Cannot connect to a remote computer
17. Building VisIt on a Windows computer
18. Installing VisIt on a MacOS X computer
19. Hanging at 12% on Windows computers
20. Getting the Plugin Developer’s Guide
21. Writing a plugin for VisIt
22. When new versions of VisIt are released
23. What is new in the latest version of VisIt
24. Compilers that can be used to build VisIt
25. VisIt’s licensing agreement
26. Slow performance with ATI cards on Linux
27. Custom plugins with a downloaded VisIt binary
28. Getting HDF5 data into VisIt
29. Getting NETCDF data into VisIt
30. When I run VisIt on my Linux machine, I get a black screen
31. I get the message ‘Publisher cannot be verified’ when installing VisIt on Windows
32. Which libraries should I enable in build_visIt?
VisIt-users Mailing List

- You may only post to mailing list if you are also a subscriber.
- Approximately 400 recipients, approx. 300 posts per month.
- Developers monitor mailing list, strive for 100% response rate.
- Response time is typically excellent (O(1 hour)).
  - International community participates … not unusual for a question from Australia to be answered by a European, while all US developers are asleep.
- List Address: visit-users@ornl.gov
- More information: https://email.ornl.gov/mailman/listinfo/visit-users
- Archive: https://email.ornl.gov/pipermail/visit-users/
VisItusers.org

- Great source for VisIt tips and recipes.
- Users section has lots of practical advice:
  - “I solved this problem using this technique”
  - “Here’s my script to do this analysis”

VisItusers.org is the VisIt project’s staging area for usage recipes and future formal documentation.
VisIt Users Forum

- [http://www.visitusers.org/forum](http://www.visitusers.org/forum)
- Increasingly popular option; you can post without receiving 300 emails a month
  - But it is viewed by less people and less well supported.
- Google indexes these pages.
Some customer groups pay for priority VisIt support:

- These customers can post directly to specific visit-help-{XYZ} support lists without subscribing.
- The messages are received by all VisIt developers and supported collectively.

Current Lists:

- visit-help-asc, visit-help-scidac, visit-help-gnep, visit-help-ascem
Manuallys & Other Documentation

- Getting Started Manual
- Users Manual
- Python Interface
- Getting Data Into VisIt
- VisIt Class Slides
- VisIt Class Exercises
- {Tutorials}
Resources

- **Presenters:**
  - Cyrus Harrison  
  
- **User resources:**
  - Main website: [http://www.llnl.gov/visit](http://www.llnl.gov/visit)
  - Wiki: [http://www.visitusers.org](http://www.visitusers.org)
  - Email: visitusers@ornl.gov

- **Development resources:**
  - Email: visit-developers@ornl.gov
  - SVN: [http://portal.nersc.gov/svn/visit](http://portal.nersc.gov/svn/visit)