An introduction to Scientific Visualization using VisIt

Scientific Visualization Concepts

Guided Tour of VisIt

With two in-depth / hands-on visualizations:

Aneurysm (Blood Flow) Simulation

Water Flow Simulation
Tutorial Outline

• Introductions:
  — Hari Krishnan (LBNL/NERSC, VisIt Developers)
• VisIt Overview
• Guided Tour & Showcase of VisIt
• Alternative ways of using VisIt.
• Demonstrations:
  — Visualization of an Aneurysm (Blood Flow) Simulation
  — Water Flow Simulation
• Closing Remarks and Questions

Tutorials:

http://visitusers.org/index.php?title=VisIt_Tutorial

Example Datasets, Blood Flow Tutorial, Water Flow Tutorial

http://visitusers.org/index.php?title=Tutorial_Data
VisIt visualization toolkit

- 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.

Density Isovolume of a 3K^3 (27 billion cell) dataset
VisIt is a vibrant project.

- 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, ...

Project Started
LLNL ASC users transitioned to VisIt
2005 R&D 100
VACET Funded
Transition to Public SW repo
VisIt 2.0 Release
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

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 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.
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 Viewer
  - VisIt GUI
  - VisIt CLI
  - Python Clients
  - Java Clients
- Parallel Cluster
  - VisIt Engine
  - Data Plugin

Data Flow Network
- Filters

Network Connection
- MPI

Data
- (Files or Simulation)
VisIt automatically switches to a scalable rendering...

- Rendering Modes:
  - Local (hardware)
  - Remote (software or hardware)
- Beyond surfaces:
  - 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.
- **Libsim library**
  - Enables coupling of simulation codes to VisIt for in situ visualization.
VisIt’s Core
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…
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

**Mesh Types**
- Curve
- Rectilinear
- Curvilinear
- Unstructured
- Points
- Molecular
Variables

- Scalars, 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
AMR Example: Image vs. Data Resolution
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

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

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

Apply an operator to transform the data (example: Slice operator)

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

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

Apply an operator to transform the data (example: Slice operator)

Apply a second operator to transform the data (example: Elevate operator)

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

Open a database, which reads from a file (example: open file1.hdf5)
Create derived quantities from fields in the file (ex: magnitude(velocity))
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

Open a database, which reads from a file (example: open file1.hdf5)
Plot a field from the file (example: density + Pseudocolor plot)
Extract quantitative information
(example: integrate density to find mass)
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

1. **Database**: Open a database, which reads from a file (ex: open file1.hdf5)
2. **Expression**: Create derived quantities from fields in the file (ex: magnitude(velocity))
3. **Operator 1**
   - Apply an operator to transform the data (ex: Slice operator)
4. **Operator 2**
   - Apply a second operator to transform the data (ex: Elevate operator)
5. **Plot**
   - Plot a field (ex: speed + pseudocolor plot)
6. **Query**
   - Extract quantitative information (ex: max speed over cross-section)
VisIt’s core building blocks

- 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
How to get VisIt to read your data.

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

• 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, 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).


Some readers are:
- Some formats that only have one reader.
- For some formats, some developers may only have one reader.

Practical Tips
Application Code Formats

- ANSYS
- Cale
- CASTRO
- CCM
- DDCMD
- Dyna3D
- Enzo
- FLASH
- FVCOM

- Gadget
- LAMMPS
- NASTRAN
- Nek5000
- OVERFLOW
- PATRAN
- Pixie
- S3D
- ZeusMP
Application Toolkit Formats

- Adventure I/O
- BoxLib
- Chombo
- ITAPS
- OpenFOAM
- SAMRAI
- Spheral
General Scientific Data Formats

- ADIOS
- CGNS
- Exodus
- HDF5
- H5Part
- NETCDF
- PDB
- Silo
- XDMF
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.
Welcome to Silo

A mesh and field I/O library and scientific database
Specialized Scientific Data Formats

- BOW
- FITS
- GDAL
- MatrixMarket
- ProteinDataBank
- ESRI Shapefile
- XYZ

DEM from GDAL

Protein Data Bank
Visualization Formats

- VTK
- EnSight
- GMV
- Plot3D
- Tecplot
- Vis5D
- Xmdv
Graphics Formats

- Image
  - (PNG, JPEG, TIFF, BMP, etc.)
- RAW
- STL
- Wavefront OBJ

Carina Nebula
General ASCII Data Formats

- Curve2D
- Lines
- PlainText
- Point3D
VisIt’s core building blocks

- 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
Pseudocolor Rendering

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

VisIt can combine volume rendering and opaque geometry
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
There are several ways to access VisIt’s Python Client Interface.

- Launch VisIt’s CLI binary:
  - `visit -cli`

- Launch for windowless batch processing:
  - `visit -nowin -cli -s <script_file.py>`

- Control VisIt from a Python interpreter:
  - `import visit`

- Record GUI actions in to Python snippets:
  - Macro Recording provides a quick path to learn VisIt’s Python Client API.
Python Client Interface Example Script

DB: noise.silo
Cycle: 0

Width (parsec)
Depth (parsec)
Height (parsec)
Demos/Visualizations.

How to get help when you run into trouble

- FAQ
  - [https://wci.llnl.gov/simulation/computer-codes/visit/faq](https://wci.llnl.gov/simulation/computer-codes/visit/faq)
- 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/simulation/computer-codes/visit/manuals](https://wci.llnl.gov/simulation/computer-codes/visit/manuals)
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.
Conclusion: 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

Slides from the VisIt class