Scalable and Distributed Visualization using ParaView

Eric A. Wernert, Ph.D.
Senior Manager & Scientist, Advanced Visualization Lab
Pervasive Technology Institute, Indiana University

Big Data for Science Workshop – July 26, 2010
Outline

1. Background on Visualization – 10 min
2. Introduction to ParaView – 15 min
3. Examples with various data types – 20 min
4. Advanced Features – 5 min
5. Q & A – 10 min
Resources

• ParaView
  – main page - http://www.paraview.org
  – Wiki - http://paraview.org/Wiki/ParaView

• Visualization in General
  – VizWorld – www.vizworld.com
1. Background on Visualization
What is Visualization?

- a visual representation of data, “picture is worth a thousand numbers”
- any technique for creating images, diagrams, or animations to communicate a message or reveal information

Images from Kitware
Types of Visualization

- **Scientific Visualization or SciVis**
  - Data has a natural spatial representation
  - Many specialty areas have special-purpose software tools – chemistry, geospatial, medical imaging, meteorology

- **Information Visualization or InfoVis**
  - Data has no natural spatial representation
  - Traditional statistical graphics, Tufte examples

- Many other types of vis - Simulation, Design, etc.
  - Each has special tools and requirements

*We’re focusing on general-purpose scientific visualization*
Samples of Scientific Visualizations

Images from IU Advanced Visualization Lab
Examples of Information Visualization
Visualization in the Scientific Workflow

• Exploration – early phases
  – I’ve got a bunch of data and need to get a general look at it. What are the trends, outliers, interesting things to look at?

• Debugging – early-middle
  – Can I get some sense of where the code is breaking down?

• Confirmation – early-late
  – Does the data show what I expect?

• Measurement & Analysis – middle-late
  – How can I visually extract information and statistics necessary for my science?

• Presentation – late
  – How can I best communicate my data and methods to the target audience?
General Visualization Pipeline

• Read → Filter → Map → Render → Interact
• Note: ParaView overloads the term “filter”
Explicit Data-Flow Paradigm

- AVS
- OpenDX
- Iris Explorer

Easy for users to develop their own custom apps
Types of Data

Dimensionality
• 1D, 2D, 3D
• Single timestep or series

Layout of Cells
• Uniform Rectilinear Grid (incl. Image data)
  – Coordinates implied
• Non-Uniform Rectilinear Grid
• Curvilinear Grid (Structured Grid)
• Unstructured Grid
• Polygonal Data
• Unstructured Points
Types of Data (2 of 2)

Datasets

Points in 3D

Connected?

Data?

Vector

Scalar

Uconnected

Connected?

Explicit

Implicit

In the cells

On the points

Image from MayaVi
Types of Rendering

Style
• Points
• Lines
• Surfaces (Polygonal)
• Volumetric
• Non-Photo-Realistic (NPR)

Interactivity
• Real-time - ~30 FPS
• Pseudo Realtime – few FPS
• Batch – render out movie frames
Types of Displays

- desktop/laptop
- stereoscopic
- ultra-high-res
- immersive displays
Types of Interaction

- filtering, thresholding
- querying (numeric, conditional)
- visually selecting
- cutting & slicing
- moving & manipulating
- measuring
- grouping & extracting
- changing representation
- changing color mapping
- …
2. Introduction to ParaView
ParaView Pedigree

- Visualization Toolkit (VTK)
  - API for scientific visualization
  - Developed at GE Corporate R&D in 1993
  - Kitware formed in 1998
  - Additional Libraries include Insight Toolkit (ITK, Image Processing) and Titan (InfoVis library)
  - VTK is used in other SciVis packages such as VisIt, 3DSlicer, MayaVi, VisTrails

- ParaView
  - End-user application developed in 1999
  - Developed by Kitware, Sandia, and Los Alamos
  - Adds GUI and parallel back-end to VTK
Scalable & Distributed Visualization Using ParaView

VTK/ParaView pipeline

Data Sources
Data files or generated

Filters
Modify the data in some way

Mappers
Convert data into geometric objects

Actors
Adjust visible properties; make geometry renderable

Renderers & Windows
Draw actors on the screen

Display & Interface Devices
**VTK/ParaView pipeline**

- **Places to divide pipeline**

  - *Transfer filtered data*
  - *Transfer geometry or graphics commands*
  - *Transfer pixels or images*

**Data Sources**
- Data files or generated

**Filters**
- Modify the data in some way

**Mappers**
- Convert data into geometric objects

**Actors**
- Adjust visible properties; make geometry renderable

**Renderers & Windows**
- Draw actors on the screen

**Display & Interface Devices**
ParaView Architecture

- Components
  - GUI/display client
  - data server
  - render server

- Key to ParaView’s scalability and distribute-ability
  - architected in from the start
  - most other vis apps don’t and therefore fail to scale
ParaView Architecture

- Other capabilities
  - Parallel backend for data, mapping, and rendering
  - Option to display to parallel tiled display
GUI Components

- Sources and filters
- Parameters for current filter
- View
- Filters
- Center of rotation
- Toolbar
- Message box
- Progress bar
- Error message

Slide from Kitware
Supported File Formats

- Chombo
- SpyPlot / CTH (.sptch)
- Exodus
- Partitioned Exodus
- HDF5 Raw (.h5)
- Extensible Data Format (.xdmf)
- EnSight (.case .sos)
- Plot3D (.xyz)
- VRML (.wrl)
- Protein Data Bank (.pdb)
- XMol Molecule (.xyz)
- Stereo Lithography (.stl)
- BYU (.g)
- Gaussian Cube File (.cube)
- Digital elevation map (.dem)

- ParaView Data (.pvd)
- VTK PolyData (.vtp)
- VTK Unstructured Grid (.vtu)
- VTK Image (.vti)
- VTK Structured Grid (.vts)
- VTK Rectilinear (.vtr)
- VTK Legacy (.vtk)
- VTK Partitioned PolyData (.pvtp)
- VTK Partitioned Unstructured grid (.pvtu)
- VTK Partitioned Image (.pvti)
- VTK Partitioned Structured Grid (.pvts)
- plus more...

You can also use built-in data sources to experiment and learn.
Methods to Keep things Interactive

• Demands of the workflow
  – data size, # of timesteps, visualization style & complexity, resolution (# of pixels)

• Computer limitations
  – Memory, CPU, graphics (polygons & pixels), networking bandwidth, internal bus bandwidth

• Methods to preserve interactivity
  – Sub-sampling & decimation
  – Level of Detail (LOD)
  – Caching of timesteps
  – Manual update of changes (Accept button)
Methods to Keep things Interactive

- Decimation
- Distributed / local rendering
- Image reduction
3. Examples
Example 1 – Nuclear “Pasta”

• Simulation of dense neutron-neutron interaction – Results in “pasta” like forms

• Data from Charles Horowitz (IU Physics) and Don Berry (IU PTI), c. 2005
Example 1 – Nuclear Pasta

- Point data – single timestep
  - Glyph filter
- Temporal data – point data over time
  - Animation Features
- Derived vector data
  - Python filter & Calculator filter
- Derived volume data
  - Isosurface (Contour)
  - Volume Rendering
Example 1 – Nuclear Pasta

• The given data may be just the beginning
• There is no single “right” way to visualize most data sets
  – Some representations are better than others for specific tasks
  – There are many “wrong” ways to visualize
• Applications like ParaView make it easy to explore parameters interactively
  – Any ParaView program can also be turned into a VTK program or script for a focused community
Example 2 – Seismic Tomography

• Data computed by Gary Pavlis and colleagues, IU Geology, c. 2007
Example 2 – Seismic Tomography

• Section of the earth’s crust → curvilinear grid
  – Slicing and clipping
  – More with color maps
  – More with isosurfaces

• Converting between volume formats
  – curvilinear grid -> tetrahedrons to allow volume rendering

• For movies, see http://seismo.geology.indiana.edu/pavlis/current_research/imaging/USArrayEars09.htm
4. Advanced Features
Client-Server Setup

See *pvserver*

- *Mpirun*
- *Offscreenrender*

Normal (forward) connection

Reverse connection (IU Quarry)
Data & Rendering Decomposition

• Image-space method – distribute data and render per tile of the image.

• Object-space method – divide data, render, composite pieces by depth.

Images from equalizer.com
Other Advanced Features

- Batch Animation
  - See pvbatch
- Scripting
  - pvpython shell & Python console
  - Scriptable Python filter
- Extending
  - Macros
  - Plug-ins
  - Custom Filters
5. Questions ?