Introduction to ParaView

Dan Lipsa,
R&D Engineer, Kitware Inc.
Contents

• ParaView description, architecture and history
• Running ParaView in parallel.
• Hands-on practice: vector visualization, data analysis
What is ParaView?

An open-source application and architecture for display and analysis of scientific datasets.

• Application - you don’t have to write any code to analyze your data
• Architecture - designed to be extensible if you want to code
  Custom apps, plugins, python scripting, Catalyst, ParaViewWeb
• Open-source – BSD 3-clause license

• Display - excels at traditional scientific vis qualitative 3D rendering
• Analysis - data drill down through charts, stats, all the way to values
• ParaView – parallel - scales from notebooks to world’s largest supercomputers
ParaView Architecture

UI (Qt Widgets, Python Wrappings)

ParaView Server

VTK

OpenGL | MPI | IceT | Etc.
History

http://www.paraview.org/Wiki/ParaView_Release_Notes

• 1999 LANL/Kitware project (via ASCI Views)
  – Build an end user tool from VTK (TCL-TK interface)
  – Make VTK scale
  – October 2002 first public release, version 0.6

• 2002-2005 Versions 0.6 through 2.6
  – Continued growth under DOE Tri Labs, Army Research Lab and various other partnerships

• September 2005 ParaQ project started
  – Sandia, Kitware and CSimSoft
  – Make ParaView easier to use (Qt interface)
  – Add quantitative analysis
  – May 2007 version 3.0 released

• Continuing to evolve
  – 3.2, 3.4, 3.6, 3.8, 3.10, 3.12, 3.14, 3.98
  – 4.0.1, 4.1, 4.2, 4.3.1 (January 2015)
User Interface

- Menu Bar
- Toolbars
- Pipeline Browser
- Object Inspector
  - Properties
  - Display
  - View
  - Information
- Online help
- View(s)
Help

- Online help

- The ParaView Guide (262 pages)
- The ParaView Tutorial
- ParaView Mailing Lists
- ParaView Wiki
- http://www.paraview.org/documentation/
How to Use ParaView

1. Read in data: File → Open, hit Apply
   - Over 100 file formats supported
   - Help/Readers - readers compiled in

2. Add a filter to process data:
   - Tune filter properties, hit Apply
   - Repeat Step 2 as needed

3. Tune Display (for all Filter, View pairs) and View (for all Views) parameters

4. Save datasets, rendered results (screenshot or animation) or application state

Filter = an object that operates on data: reads its inputs and produces one or more outputs
Reader = reads a file and produces an output
Source = produces an output (Cylinder)
Pipeline object = a filter, reader or source
Filter Properties and the Apply Button

- Active Filter properties
- ParaView is meant to process large data – it might take a long time when changing a filter property.
ParaView Dataset Types

- vtkImageData
- vtkRectilinearGrid
- vtkStructuredGrid
- vtkPolyData
- vtkUnstructuredGrid

- Multi-blocks
- AMR
- Time-varying data

- points, cells
- values associated points and/or cells: scalars, vectors, tensors
Object Inspector
Information Tab

- Information about the Active Filter’s output
- Dataset Type
- Size (Bytes, #points, #cells)
- Geometric bounds
- Structured bounds
- Arrays:
  - Name
  - Association: point, cell
  - Data Type
  - Data Ranges (and scalar/vector)
- Temporal Domain
Multiple Views

- Split (Vertical, Horizontal), Maximize, Close buttons
- Active view: Display and View properties pertain to it
- Right-click to link cameras
Display Properties

- **A Representation** (a display): object that stores visual characteristics of one particular data set in one particular view
- Properties associated with the Active Filter and Active View.
Color Map Editor

Mapping
Scalar Range – Color Palette

- Rescale to data range
- Rescale to custom range
- Rescale to data range over all time-steps
- Invert the transfer function
- Choose preset
- Save to preset
Color Map (Transfer Function) Editor
View Properties

Properties associated with the Active View
Find properties (for Filters, Displays and Views)

- Search for properties
- Toggle on/off advanced properties

Advanced Properties
Multi-View Visualization Pipeline

Source

Data

Filter

Data

Filter

Data

View

Representation

Representation

Representation

Representation

View
Pipeline Browser

Source

Data

Filter

Data

Filter

Data

Representation

View

Representation

Representation

View

Representation

View
Filter Properties – acts on active filter

Source

Data

Filter

Data

Filter

Data

Filter

Data

Filter

Data

Representation

Representation

Representation

Representation

Representation

View
Information Tab – shows output data for the active filter
Display Properties – representation properties for active filter and view
View Properties – acts on active view
Filters

- Filters Menu
  - Recent
  - Common
  - Data Analysis
  - Statistical
  - Temporal
  - Alphabetical
- Quick Launch
  - PC/Linux
    - CTRL-Space
  - Mac
    - ALT-Space
- Apply Undo/Redo

- Calculator
- Contour
- Clip
- Slice
- Threshold
- Extract Subset
- Glyph
- Stream Tracer
- Warp By Vector
- Group Datasets
- Extract Group
Query Data by Attributes Values – Find Data Dialog
Query Data Visually - Selection

- Visually select interesting data
- shown in all compatible views
- can then label, extract etc
  - ‘Select Cells On’ to get nearest cells
  - Select Points On’ to get nearest points
  - ‘Select Cells Through’ to get all cells intersecting a frustum
  - ‘Select Points Through’ for selecting points inside a frustum
Exporting data, images, sessions

- **File → Save** *
  - Active filter’s data, prompted for file format
    - List of file formats given in help primarily kitware formats + exodus, ensight, xdmf/hdf5, csv
  - Screen shot, either selected view or all
  - Export visible scene in a format for high quality rendering
    - eps, pdf, ps, svg, pov, vrml, webgl, x3d, x3db
  - Movie
    - Image sequence, avi, ogg, ffmpeg → avi
  - State
    - for restoring ParaView session later
What to Expect from Parallel Processing

Serial + Parallel = 1 (percentage that you cannot/can parallelize)

• **Amdahl’s Law**
  \[ \text{Speedup(CPUs)} = \frac{1}{\text{Serial} + \frac{\text{Parallel}}{\text{CPUs}}} \]
  aka Strong scaling
  If data size is fixed, don’t expect great scalability.

  More processors != faster

• **Gustafson’s Law**
  \[ \text{Speedup(CPUs)} = \text{CPUs} \times \frac{\text{Parallel}}{\text{Parallel} + \text{Serial}} \]
  aka Weak scaling
  As data size grows, you must have more resources.

  More disk and IO = higher resolution possible
What to Expect from Parallel Processing

- Parallel $\neq$ Faster
- Parallel $\rightarrow$ Larger Problem Size
Large Data processed by ParaView

1 billion cell asteroid detonation simulation

½ billion cell weather simulation

source: Sandia National Lab
N component Data Parallelism for X GByte

Client
Control, Display and Rendering of Small Data

Data Server

Render Server
Depth Composite

Reader

Filter

X/N GB

Reader

Filter

X/N GB

Reader

Filter

Reader

Filter

Tile Display
## ParaView’s Running Modes

<table>
<thead>
<tr>
<th>Mode</th>
<th>Diagram</th>
<th>Description</th>
</tr>
</thead>
</table>
| Builtin aka Standalone aka Serial | ![Diagram](image) | all components within one process (client may be GUI or pvpython)  
paraview || pvpython |
| Combined Server       | ![Diagram](image) | data processing and parallel rendering in MPI job of combined processes. control from TCP connected client.  
mpiexec -n x pvserver &;  
paraview # or pvpython #+ Connect |
| Batch                 | ![Diagram](image) | server is an MPI job which directly runs a python script  
mpiexec -n x pvbatch \  
vis_script.py |

**DS** = data server  
**RS** = render server
Connecting to a Server

- Follow instructions at [http://www.alcf.anl.gov/user-guides/paraview-cooley](http://www.alcf.anl.gov/user-guides/paraview-cooley)
- File → Connect

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooley</td>
<td>cs://cc010.cooley.pub.alcf.anl.gov:8000</td>
</tr>
</tbody>
</table>

![Server Configuration Screen](image)
Level of Detail – Maintain Interactivity

Type 1: Geometrically based

• Edit → Settings → Render View →
• LOD threshold = 0.1
• Down-samples geometry while interacting
Level of Detail – Maintain Interactivity

Type 2: Image Based

- Edit → Settings → Render View →
- Remote Render Threshold = 0.1
- Image Reduction Factor = 10
- Down-samples pixels while interacting
Hands on practice: vector visualization
(see also http://www.paraview.org/Wiki/The_ParaView_Tutorial)

- Load disk_out_ref.ex2
  - An exodus format file
  - Enable all variables
- Information tab
  - Multi-block (group of datasets)
  - Not time varying
  - Roughly 8000 cells and points, 2MB
  - 11.5 units in diameter, 20 units in height
- Show as surface with edges to see structure
- Set opacity to 0.5
- Looks like a cylinder with a recess
Hands on practice: vector visualization

- Apply slice filter
  - Align with z and use 10 offset values
- Color by Temp
- Show temp color key
- Adjust opacity of reader(0.1) and slice(1.0) so that you can see temperature variation clearly
Hands on practice: vector visualization

- Apply warp filter
  - Warp slices along V vector field with a scale factor of 0.1
- Compare with display of slice
  - note see how vector field pushes up in center and down further out
  - We are seeing convection of a heated gas, it rises at the heat source
Hands on practice: vector visualization

- Change warp opacity to .2
- Apply streamline filter
  - Starts from seed points and advects along vector field to show you vector flow
- Apply tube filter
  - Gives infinitely thing streamlines extent so we can see them well
- Set opacity to 1.0 and color by vorticity
  - We are seeing rotation
  - A heated plate is spinning in gas
- Manipulate streamline’s seed points
Putting it together: data analysis
Thank You. Questions?