Scientific Visualization with ParaView

Weiguang Guan
guanw@sharcnet.ca

ParaView: http://www.paraview.org/download
Code & data: https://rhpcserv.rhpcs.mcmaster.ca/~guanw/ss2015viz.zip
What is Visualization

Data → Graphical representations
What is Visualization used for

- To comprehend data
- To explore and discover
  - Patterns
  - Structures
  - Trends
- To monitor simulations
- To communicate with others

A picture is worth of thousand words numbers!
Numbers (S&P 500)

<table>
<thead>
<tr>
<th>Date</th>
<th>Open</th>
<th>High</th>
<th>Low</th>
<th>Close</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul 14, 2015</td>
<td>2,099.72</td>
<td>2,111.98</td>
<td>2,098.18</td>
<td>2,108.95</td>
</tr>
<tr>
<td>Jul 13, 2015</td>
<td>2,080.03</td>
<td>2,100.67</td>
<td>2,080.03</td>
<td>2,099.60</td>
</tr>
<tr>
<td>Jul 10, 2015</td>
<td>2,052.74</td>
<td>2,081.31</td>
<td>2,052.74</td>
<td>2,076.62</td>
</tr>
<tr>
<td>Jul 9, 2015</td>
<td>2,049.73</td>
<td>2,074.28</td>
<td>2,049.73</td>
<td>2,051.31</td>
</tr>
<tr>
<td>Jul 8, 2015</td>
<td>2,077.66</td>
<td>2,077.66</td>
<td>2,044.66</td>
<td>2,046.68</td>
</tr>
<tr>
<td>Jul 7, 2015</td>
<td>2,069.52</td>
<td>2,083.74</td>
<td>2,044.02</td>
<td>2,081.34</td>
</tr>
<tr>
<td>Jul 6, 2015</td>
<td>2,073.95</td>
<td>2,078.61</td>
<td>2,058.40</td>
<td>2,068.76</td>
</tr>
<tr>
<td>Jul 2, 2015</td>
<td>2,078.03</td>
<td>2,085.06</td>
<td>2,071.02</td>
<td>2,076.78</td>
</tr>
<tr>
<td>Jul 1, 2015</td>
<td>2,067.00</td>
<td>2,082.78</td>
<td>2,067.00</td>
<td>2,077.42</td>
</tr>
<tr>
<td>Jun 30, 2015</td>
<td>2,061.19</td>
<td>2,074.28</td>
<td>2,056.32</td>
<td>2,063.11</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Graph (S&P 500)
More examples
More examples
More and more examples
Sources of tutorial materials

- “The ParaView Tutorial”, K. Morland
- “Scientific Visualization”, Alex Razoumov, HPC Summer School 2014

NOTE: Some slices of above materials together with their scripts and data are copied and used in this tutorial.
Outline (3 hours)

- Overview of ParaView
- Basic Usage of ParaView
- Advanced Topics of ParaView
- Put It All Together
Overview of ParaView

- Introduction
  - www.paraview.org
  - Current version 4.4
  - Kitware, Los Alomas, Sandia National Labs, etc
  - Open source
  - General-purpose visualization package
  - Large user base
Overview of ParaView

• Introduction
  • Multiple platforms (Windows, Linux/Unix, MacOS)
  • Parallel visualization of large data
  • Multi-dimensional and/or Multi-variables
  • Either standalone or client/server mode
  • Rich functionalities implemented as filters
  • Many file formats
  • Various data types
  • Scripting
Overview of ParaView

- Documentations and resources
  - Wiki: [http://www.paraview.org/Wiki/ParaView](http://www.paraview.org/Wiki/ParaView)
  - Tutorials: [http://www.paraview.org/tutorials](http://www.paraview.org/tutorials)
  - Webinars: [http://www.paraview.org/webinars](http://www.paraview.org/webinars)
Overview of ParaView

• Documentations and resources
  • Mailing lists:
    • Searchable archive:
      http://public.kitware.com/pipermail/paraview
    • Users:
      http://www.paraview.org/mailman/listinfo/paraview
    • Developers
      http://www.paraview.org/mailman/listinfo/paraview-developers
  • Help online F1 or context-sensitive
Overview of ParaView

- **Architecture**

<table>
<thead>
<tr>
<th>ParaView Client</th>
<th>pvpython</th>
<th>ParaWeb</th>
<th>Catalyst</th>
<th>Custom App</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI (Qt Widgets, Python Wrappings)</td>
<td>ParaView Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VTK</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenGL</td>
<td>MPI</td>
<td>IceT</td>
<td>Etc.</td>
<td></td>
</tr>
</tbody>
</table>
Overview of ParaView

- Standalone

![Diagram showing client-server architecture]

Client

- Data Server
- Render Server
Overview of ParaView

- Client-Server mode
Overview of ParaView

- Visualization pipelines

Source-1 → Filter-1 → Filter-2 → ... → Filter-N

Source-2 → Filter-b
Overview of ParaView

- **Pipeline components**
  - **Source**: no input, one or more outputs
    - Primitive objects (cube, cylinder, cone, sphere, etc)
    - Data file readers
  - **Filter**: one or more inputs, one or more outputs
Overview of ParaView

- Connections in pipeline
  - Outputs of one component become the inputs of another
  - Compatibility of data type between inputs and outputs
Overview of ParaView

- Data types (sampling structures)

  - Uniform Rectilinear (vtkImageData)
  - Non-Uniform Rectilinear (vtkRectilinearData)
  - Curvilinear (vtkStructuredData)
  - Polygonal (vtkPolyData)
  - Unstructured Grid (vtkUnstructuredGrid)
  - Multi-block
    - Hierarchical Adaptive Mesh Refinement (AMR)
    - Hierarchical Uniform AMR
    - Octree
Overview of ParaView

- Data types at sampling points
  - Scalars (density, temperature, pressure, etc)
  - Vectors (velocity)
  - Normals
  - Texture Coordinates
  - Tensors
Overview of ParaView

- Example of pipeline

CT data reader ➔ Volume data ➔ Contour filter ➔ Polygonal mesh

- File name
- Iso-value
5 minutes
Basic Usage of ParaView

- Walk through GUI elements with simple examples
  - Primitives (Cylinder, sphere, etc)
  - Readers (air flow dynamics simulation, etc)
  - Filters (clipping, contouring, calculator, etc)

- What to learn
  - Create and manipulate pre-defined objects
  - Import objects from data files
  - View information of objects
  - Change display properties
  - Transform objects by applying filters
  - Multiple views and camera linkage
  - Save results (screenshots, animations, states)
Basic Usage of ParaView

- SHARCNet visualization stations
  - 10 viz stations (≤64GB memory, ≤12 cores, modern GPUs)
  - [https://www.sharcnet.ca/my/software/show/67](https://www.sharcnet.ca/my/software/show/67)

- ParaView on SHARCNet viz stations
  - VNC (TigerVNC)
  - Client-server mode
Basic Usage of ParaView

- Start ParaView
  - Linux/Unix: type `paraview` in a command window
  - MacOS: click ParaView in `Applications` folder
  - Windows: click ParaView from Start Menu

Then
- ParaView GUI appears
- `Pvserver` running in the background
Basic Usage of ParaView

- ParaView GUI
  - Menu Bar
  - Toolbars
  - Pipeline Browser
  - Properties Panel
  - Advanced Toggle
  - 3D View
Basic Usage of ParaView

- **Example 1**
  - Cylinder, sphere, cube
  - Multiple views and linking cameras

- **Example 2**
  - Loading file “disk_out_ref.ex2”
  - Volume rendering (transfer function)
  - Streamline and tube

- **Example 3**
  - Wavelet
  - Calculator
Basic Usage of ParaView

- Summary
  - Mouse interactions (Edit → Setting... Camera tab)
    - Me (Camera) or objects move?
  - Objects vs Views
    - An object can be displayed in more than one view
    - An object can have different appearances in different views
  - Active object (highlighted in pipeline browser)
    - Shown in Properties Panel
    - Subsequent transforms (filters) will apply to
  - Active view (highlighted by a blue outline)
    - Shown in Properties Panel
Basic Usage of ParaView

• Summary
  • Parameters
    • Define and change the underlying data
      • Geometric properties of primitives (height, radius, ...)
      • Parameters of filters (iso-value, ...)
    • Define and change only the appearance
      • Graphical representations (Surface, wireframe, points, etc)
      • Color
      • Lighting
      • Surface reflection properties
Basic Usage of ParaView

- **Summary**
  - Caution while handling large structured datasets
    - Avoid using filters that generate unstructured data

- Append Datasets
- Append Geometry
- Clean
- Clean to Grid
- Clip
- Connectivity
- D3
- Delauney 2D3D
- Extract Edges
- Linear Extrusion
- Lopp Subdivision

- Reflect
- Rotational Extrusion
- Shink
- Slice
- Smooth
- Subdivide
- Tessellate
- Tetrahedralize
- Triangle Strips
- Triangulate
5 minutes
Advanced Topics of ParaView

- Python scripting
- Importing your data
- Animation
- Visualization of large data in client-server mode
Advanced Topics of ParaView

- Python scripting
  - Scripting vs GUI-based operations
    - One GUI-based operation ⟷ one or more scripting commands
  - Three ways of scripting
    - Tools ⟷ Python Shell, you can mix
      - Run scripts
      - Interactively issue commands
      - GUI-based operations
    - Pypython: interactive client
    - Pvbatch: non-interactive batch execution
Advanced Topics of ParaView

• Python scripting
  • Basic elements
    • `from paraview.simple import *` # import paraview module
    • `sphere = Sphere()` # create a sphere
    • `Show()` # turn on visibility
    • `Render()` # refresh display
  • Help command
    • `help(paraview.simple)`
    • `help(Sphere)`
    • `help(sphere)`
Advanced Topics of ParaView

- Python scripting
  - Python basics
    - import os
    - os.getcwd()
    - os.chdir(path)
    - os.listdir(path)

Example of displaying files in current working directory:
os.listdir(os.getcwd())
Advanced Topics of ParaView

- Importing your own data
  - Write your data into
    - **Raw data file** (writeRawBinary.cpp). While loading file, you need to specify
      - Data type
      - Endianness
      - Dimensionality
      - Data array origin, extend, spacing along each dimension
    - CVS (not for large data)
    - **NetCDF** (writeNetCDF.cpp)
      - Meta data saved into file as well
    - Various VTK formats
Advanced Topics of ParaView

- Animation
  - Displaying time varying datasets
  - User defined animations
    - Dynamically changing parameters of objects you create
    - Dynamically changing parameters of camera
Advanced Topics of ParaView

- Visualization of large data in client-server mode
  - VNC and run ParaView remotely on supercomputers
  - Run ParaView in client-server mode
    - Place “servers.pvsc” file in
      - C:\Users\yourUserName\AppData\Roaming\ParaView on Windows
      - /home/yourUserName/.config/ParaView on Linux/MacOS
    - Set up passphraseless ssh by adding your public key to .ssh/authorized_keys on remote computer
    - File ➔ Connect ...
- ParaView (VTK inside) has built-in parallel mechanisms
Questions?
or
Put It All Together

Task: Create an animation, in which a skull moves in a 3D scalar field (e.g., temperature). The skull surface is color-mapped with the intensities it insects with the field. As the skull moves, colors on its surface changes as well.

- Create a skull from medical images
- Create a scalar volume representing temperature
- Attach a “Transform” filter to skull
- Map volume data on the transformed skull by using “Resample with dataset” filter
- Create an animation on parameters of the “Transform” filter