ParaView & VTK on BigRed-II

Big Red-II Workshop
Indiana University
Advanced Visualization Lab
Bill Sherman
October 2, 2013
What are ParaView & VTK?

• Open-source visualization API & Tool.
• A flexible BSD-2 license
• Maintained and supported by Kitware, Inc.
• Commercial maintenance and support.
VTK

• A collection of algorithms tailored for scientific visualization
• Dataflow pipeline w/o the user-interface
• Multi-language API
  – C++
  – TCL
  – Python
  – Java
• www.vtk.org / kitware.com
ParaView

- An open-source, scalable, multi-platform visualization application.
- Support for distributed computation models to process large data sets.
- An open, flexible, and intuitive user interface.
- An extensible, modular architecture based on open standards.
- [www.paraview.org](http://www.paraview.org) / kitware.com
VTK vs. ParaView

• Why use one versus the other?
• VTK:
  – Developing your own application
  – Legacy Code / Skills
• ParaView:
  – Want to use an existing application
  – Want to easily take advantage of parallel processing features
  – Want to work on remote data
VTK Development

• Started in 1993 as the example code to accompany a book on scientific visualization.
  – ParaView 0.6 released October 2002.
• 1998 team left GE R&D to form Kitware, Inc. to support what had become a large world-wide community.
• Continues to evolve:
  – Recent additions include GPU rendering
  – VTK 6.0 released June 2013.
• http://www.vtk.org/Wiki/VTK/Learning_VTK
VTK Language Choices

**C++**
- Native to the library
- Fast

**Tcl**
- Interpretive
- Less code
- Portable applications

**Python**
- Interpretive
- Less code
- Object Oriented
- Portable applications
- Choice of GUIs

**Java**
- More portable (than C++)
- Pre-compiled for MS-Windows*
VTK UI Choices

Code  
(the default!)

GUIs (ie. applications built on VTK)
Paraview (menu driven)
VisIt (menu driven)
VisTrails (VPL w/ Provenance)
MeVisLab (focus on Medical Imaging)
Example 1: a sphere

vtkSphereSource

vtkPolyDataMapper

vtkActor

vtkRenderer

vtkRenderWindow

vtkRenderWindow-Interactor
Example 1: a sphere

# Create a basic sphere object
vtkSphereSource sphere
    sphere SetPhiResolution 12
    sphere SetThetaResolution 12

# Map poly data into renderable geom
vtkPolyDataMapper sphereMapper
    sphereMapper SetInput [sphere GetOutput]

vtkActor sphereActor
    sphereActor SetMapper sphereMapper
Example 1: a sphere

```plaintext
# Setup standard rendering
vtkRenderer ren1
  ren1 SetBackground 1 1 1
vtkRenderWindow renWin
  renWin addRenderer ren1
  renWin SetSize 800 600
# Create & initialize interactor
vtkRenderWindowInteractor iren
  iren SetRenderWindow renWin
  [iren GetInteractorStyle] to Trackball Camera
  iren Initialize
ren1 AddActor sphereActor
```
The Dataflow Paradigm

• Comprised of modules that:
  – Get/make data
  – Filter data
  – Map data
  – Render Geometry
  – Etc.
• Modules are linked by “pipes” through which data flows
• Provides a visualization of the visualization
DataFlow in VTK

- VTK is object oriented
  - Everything is an “object”
  - Modules are called “process objects”
  - The data in the pipes are “data objects”

- VTK breaks rank with most other dataflow visualization tools by making a sharp distinction between the “visualization” modules and the “graphics” modules.
Data Types

- Uniform Rectilinear (vtkImageData)
- Non-Uniform Rectilinear (vtkRectilinearGrid)
- Curvilinear (vtkStructuredGrid)
- Polygonal (vtkPolyData)
- Unstructured Grid (vtkUnstructuredGrid)

Multi-block
- Hierarchical Adaptive Mesh Refinement (AMR)
- Hierarchical Uniform AMR
- Octree
VTK on BigRed-II
Data modification

• Splatify example – take two point cloud datasets and produce a volume of relative densities (using a Gaussian “splat” technique):
  % module load vtk
  % aprun splatify
TCL version of splatify

set input_file "evdata_${timestep}.vtk"
set output_file "evsplat1Mi_${timestep}.vtk"

##### Setup VTK
package require vtk

vtkPolyDataReader reader
    reader SetFileName $input_file
    reader SetScalarsName "type"
    reader SetVectorsName "velocity"
    reader Update
Get type-1 data

```python
vtkThreshold threshA
    threshA SetInputConnection [reader GetOutputPort]
    threshA ThresholdBetween 1 1
    threshA Update

vtkGaussianSplatter splatA
    splatA SetInputConnection [threshA GetOutputPort]
    splatA SetSampleDimensions $resolution $resolution $resolution
    splatA SetRadius $radius
    splatA ScalarWarpingOff
    splatA SetAccumulationModeToSum
    splatA SetModelBounds {*}$modelbounds
    splatA Update
set arrayA_size [$arrayA GetNumberOfTuples]
```
Get type-2 data

vtkThreshold threshB
  threshB SetInputConnection [reader GetOutputPort]
  threshB ThresholdBetween 2 2
  threshB Update

vtkGaussianSplatter splatB
  splatB SetInputConnection [threshB GetOutputPort]
  splatB SetSampleDimensions $resolution $resolution $resolution
  splatB SetRadius $radius
  splatB ScalarWarpingOff
  splatB SetAccumulationModeToSum
  splatB SetModelBounds {*}$modelbounds
  splatB Update

set arrayB_size [$arrayB GetNumberOfTuples]
Blend the data

# Create the new data field
for { set count 0 } { $count < $arrayA_size } { incr count } {
  set valueA [arrayA GetTuple1 $count]
  set valueB [arrayB GetTuple1 $count]
  if {$valueA == 0 && $valueB == 0} {
    set resultA -0.1
    set resultB -0.1
    incr empty
  } elseif {[expr $valueA + $valueB] < $discount} {
    set resultA -0.1
    set resultB -0.1
    incr discounted
  } else {
    set resultA [expr $valueA / ($valueA + $valueB)]
    set resultB [expr $valueB / ($valueA + $valueB)]
    incr counted
  }
  $arrayA SetTuple1 $count $resultA
}
Write the data

vtkDataSetWriter writer
writer SetInputConnection [splatA GetOutputPort]
writer SetFileTypeToBinary
writer SetFileName $output_file
# Add some data provenance in the header
writer SetHeader "Vesperini data ($input_file)
processed with Gaussian Splat (splatify.tcl), resolution
$resolution, radius $radius, model bounds
$modelbounds', discount at $discount -- empty/
discounted/counted = $empty/$discounted/$counted"

writer SetScalarsName "blend"
writer Update
ParaView
ParaView Development

• Started in 2000 as collaborative effort between Los Alamos National Laboratories and Kitware Inc. (lead by James Ahrens).
  – ParaView 0.6 released October 2002.

• September 2005: collaborative effort between SNL, Kitware Inc. and CSimSoft to rewrite user interface to be more user friendly and develop quantitative analysis framework.
  – ParaView 4.0 released June 2013.
Current Funding

- ARL
- ERDC
- US Army (SBIR)
- US Air Force (STTR)
- ONR
- Support Contracts
  - Electricity de France
  - Microsoft
- Other contributors
  - Swiss National Supercomputing Centre
  - DOE SLAC
  - Ohio State
  - Mississippi State
  - RPI
# ParaView Application Architecture

<table>
<thead>
<tr>
<th>ParaView Client</th>
<th>pvpython</th>
<th>Custom App</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI (Qt Widgets, Python Wrappings)</td>
<td>ParaView Server</td>
<td></td>
</tr>
<tr>
<td>VTK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenGL</td>
<td>MPI</td>
<td>IceT</td>
</tr>
</tbody>
</table>
Data Ranges

• Used for all ranges of data size.
• Landmarks of SNL usage:
  – 6 billion structured cells (2005).
  – Billions of AMR cells (2008).
  – Scaling test over 1 Trillion cells (2010).
More Information

• Online Help – F1
• The ParaView User’s Guide
  – http://paraview.org/Wiki/ParaView/Users_Guide/Table_Of_Contents

• The ParaView web page
  – www.paraview.org
• ParaView mailing list
  – paraview@paraview.org
Basic Usage
User Interface

- Menu Bar
- Toolbars
- Pipeline Browser
- Object Inspector
- 3D View
GUI Component Collections
Creating a Cylinder Source

1. Go to the **Source** menu and select Cylinder.
2. Click the **Apply** button to accept the default parameters.
Creating a Cylinder Source

1. Go to the **Source** menu and select **Cylinder**.
2. Click the **Apply** button to accept the default parameters.
3. Increase the **Resolution** parameter.
4. Click the **Apply** button again.
Pipeline Object Controls

- Apply
- Reset
- Delete
Simple Camera Manipulation

• Drag left, middle, right buttons for rotate, pan, zoom.
  – Also use Shift, Ctrl modifiers.
Undo Redo

Undo
Redo

Camera Undo
Camera Redo
Render View Options

[Image of Render View Options dialog box]

- General
- Lights
- Annotation

Light Kit Parameters
- Key: 0.60
- Fill: 0.40
- Back: 0.50
- Head: 0.50
- Int: 0.75
- K:R: 3.00
- K:B: 3.50
- K:H: 3.00

Options:
- Default Light
- SetLightColor
- Light Kit Parameters
- Maintain Luminance

Buttons:
- Apply
- Reset
- Ok
Properties/Display/Information
Creating a Cylinder Source

1. Go to the Source menu and select Cylinder.
2. Click the Apply button to accept the default parameters.
3. Increase the Resolution parameter.
4. Click the Apply button again.
5. Delete the Cylinder.
**Supported Data Types**

- ParaView Data (.pvd)
- VTK (.vti, .vtu, .vtp, .vts, .vtr)
- VTK Legacy (.vtk)
- VTK Multi Block (.vtm, .vtmb, .vtmg, .vthd, .vthb)
- Partitioned VTK (.pvtu, .pvts, .pvti, .pvtr)
- ADAPT (.nc, .cdf, .elev, .ncd)
- ANALYZE (.img, .hdr)
- ANSYS (.inp)
- AVS UCD (.inp)
- BOV (.bov)
- BYU (.g)
- CCSM MTSD (.nc, .cdf, .elev, .ncd)
- CCSM STSD (.nc, .cdf, .elev, .ncd)
- CEAucl (.ucd, .inp)
- CMAT (.cmat)
- CTRL (.ctrl)
- Chombo (.hdf5, .h5)
- Claw (.claw)
- Comma Separated Values (.csv)
- Cosmology Files (.csv)
- Curve2D (.curve, .ultra, .ult, .u)
- DBCMDD (.dcmdd)
- Digital Elevation Map (.dem)
- Dyna3D (.dyn)
- EnSight (.case, .sos)
- Enzo boundary and hierarchy
- Exodus (.g, .e, .exe, .ex2, .ex2v, etc)
- ExtrudedVol (.exvol)
- FVCOM (MTMD, MTSD, Particle, STSD)
- Facet Polygonal Data
- Flash multiblock files
- Fluent Case Files (.cas)
- GGCM (.3df, .mer)
- GTC (.h5)
- GULP (.trg)
- Gadget (.gadget)
- Gaussian Cube File (.cube)
- JPG Image (.jpg, .jpeg)
- LAMMPS Dump (.dump)
- LAMMPS Structure Files
- LODI (.nc, .cdf, .elev, .ncd)
- LODI Particle (.nc, .cdf, .elev, .ncd)
- LS-DYNA (.k, .lsdyna, .d3plot, .d3plot)
- M3DCI (.h5)
- MFIX Unstructured Grid (.RES)
- MM5 (.mm5)
- Meta Image (.mhd, .mha)
- Miranda (.mir, .raw)
- Multilevel 3d Plasma (.m3d, .h5)
- NASTRAN (.nas, .f06)
- Nek5000 Files
- Nrrd Raw Image (.nrrd, .nhdr)
- OpenFOAM Files (.foam)
- PATRAN (.neu)
- PLOT2D (.p2d)
- PLOT3D (.xyz, .q, .x, .vp3d)
- PLY Polygonal File Format
- PNG Image Files
- POP Ocean Files
- ParaDIS Files
- Phasta Files (.pht)
- Pixie Files (.h5)
- ProSTAR (.cel, .vrt)
- Protein Data Bank (.pdb, .ent, .pdb)
- Raw Image Files
- Raw NRRD image files (.nrrd)
- SAMRAI (.samrai)
- SAR (.sar, .sar)
- SAS (.sasgeom, .sas, .sasdata)
- SESAME Tables
- SLAC netCDF mesh and mode data
- SLAC netCDF particle data
- Silo (.silo, .pdb)
- Sphaler (.spheral, .sv)
- SpyPlot CTH
- Spy Plot (.case)
- Stereo Lithography (.stl)
- TFT Files
- TIFF Image Files
- TSurf Files
- Tecplot ASCII (.tec, .tp)
- Tecplot Binary (.plt)
- Tetrad (.hdf5, .h5)
- UNIC (.h5)
- VASP CHGCA (.CHG)
- VASP OUT (.OUT)
- VASP POSTCAR (.POS)
- VPIC (.vpc)
- VRML (.wrl)
- Velodyne (.vld, .rst)
- VizSchema (.h5, .vsh5)
- Wavefront Polygonal Data (.obj)
- WindBlade (.wind)
- XDMF and hdf5 (.xmf, .xdmf)
- XMol Molecule
Custom Data Import: Prototype with Python

- A “programmable source” lets you program data readers right in the GUI.
- Uses wrappings for the basic VTK classes.
- Good for prototyping readers.

```python
import random
output = self.GetOutput()
npoints = 12
points = vtk.vtkPoints()
for p in xrange(npoints):
    points.InsertNextPoint(random.random())
output.SetPoints(points)

verts = vtk.vtkCellArray()
for c in xrange(npoints):
    verts.InsertNextCell(1)
    verts.InsertCellPoint(c)
output.SetVerts(verts)
```
Custom Data Import:
Plugin Containing a Reader

• Plugins: shared object libraries that can be dynamically loaded into ParaView.
• Any VTK reader object can be added.
A mini tutorial
Load disk_out_ref.ex2

1. Open the file disk_out_ref.ex2.
2. Load all data variables.
3. Click Apply
Data Representation

Toggle Color Legend

Reset Scalar Range

Mapped Variable

Vector Component

Representation

Edit Colors
Geometry Representations

Points  Wireframe  Surface  Surface with Edges  Volume
Common Filters

- Calculator
- Contour
- Clip
- Slice
- Threshold
- Extract Subset
- Glyph
- Stream Tracer
- Warp (vector)
- Group Datasets
- Extract Level
Filters Menu

- Recent
- Common
- Cosmology
- Data Analysis
- Statistics
- Temporal
- Alphabetical
Apply a Filter

1. Make sure that disk_out_ref.ex2 is selected in the pipeline browser.
2. Select the contour filter.
3. Change parameters to create an isosurface at Temp = 400K.

Change to Temp

Change to 400
Apply a Filter

1. Make sure that disk_out_ref.ex2 is selected in the pipeline browser.
2. Select the contour filter.
3. Change parameters to create an isosurface at Temp = 400K.
4. Apply
Create a Cutaway Surface

1. Select `disk_out_ref.ex2` in the pipeline browser.
2. From the Filters list, select `Extract Surface`.
3. Apply
Create a Cutaway Surface

1. Select disk_out_ref.ex2 in the pipeline browser.
2. From the quick launch, select Extract Surface.
3. Create a clip filter.
5. Apply
6. Apply
Multiview
Multiview

1. Split the view horizontally. (choose “3D View” view option)
2. Make Clip1 visible.
3. Color surface by Temp.
Multiview

1. Split the view horizontally.
2. Make Clip1 visible.
3. Color surface by Temp.
4. Right-click view background, Link Camera…
5. Click other view.
Multiview

1. Split the view horizontally.
2. Make Clip1 visible.
3. Color surface by Temp.
4. Right-click view background, Link Camera...
5. Click other view.
6. Click
Modifying Layout of Views
Reset ParaView
Streamlines

1. Open disk_out_ref.ex2. Load all variables.
2. Add stream tracer.
3. [Apply]
Streamlines

1. Open disk_out_ref.ex2. Load all variables.
2. Add stream tracer.
3. 
4. From the quick launch, select Tube
5. 

Apply
Reset ParaView
Plotting

1. Open disk_out_ref.ex2. Load all variables.
2. Clip, uncheck, 1. Set “Color by” to “Temp”
3. Select disk_out_ref.ex2
4. Filters → Data Analysis → Plot Over Line.
3D Widgets
Plotting

1. Open disk_out_ref.ex2. Load all variables.
2. Clip, uncheck, 1. Set “Color by” to “Temp”
3. Select disk_out_ref.ex2
4. Filters → Data Analysis → Plot Over Line.
5. Once line satisfactorily located,
Interacting with Plots

- Left, middle, right buttons to pan, zoom.
- Mouse wheel to zoom.
- Reset view to plot ranges.
Reset ParaView
Volume Rendering

1. Open disk_out_ref.ex2. Load all variables.
2. Change variable viewed to Temp.
3. Change representation to Volume.
Transfer Function Editor
Transfer Function Editor
Time and Animation
Reset ParaView
Loading Data with Time

1. Open the file can.ex2.
2. Select all variables.
3. 
4. 
5. 
Animation Toolbar

- First Frame
- Previous Frame
- Play
- Next Frame
- Last Frame
- Loop Animation
- Current Time
- Current Time Step
Animation Pitfall

1. Go to first time step. 
2. Color by EQPS variable. 
3. Turn on color legend. 
4. Play (or skip to last time step ).
Animation Pitfall

1. Go to first time step. ◀
2. Color by EQPS variable.
3. Turn on color legend. ▓
4. Play ▶ (or skip to last time step ▶ ).
5. Fix with Rescale to Data Range. ←→
Data Range Workarounds

• Go to representative time and hit (rescale data button)
• In Settings change On File Open to Goto last timestep.
Data Range Workarounds

- Open color scale editor dialog
- Set scalar range.

![Color Scale Editor dialog](image)
Data Range Workarounds

- Open color scale editor dialog
- Rescale to Temporal Range
Animation View

View → Animation View
Animation View

View → Animation View

Animation Modes: Sequence, Real Time, and Snap To TimeSteps
Changing Animation Timing

1. Open can.ex2.
   Load all variables.
2. Change animation mode to Real Time.
   NOTE: Default animation time is 10 sec.
3. ▶
Changing Animation Timing

1. Open can.ex2.
   Load all variables.

2. Change animation mode to Real Time.
   NOTE: Default animation time is 10s

3. 

4. Change animation time to 60 sec.

5. 
   again.
Smoothing the Animation

1. From the quick launch, select Temporal Interpolator.
2. 
3. Verify mode set to Real Time.
4. Split view. Show can.ex2 in one and TemporalInterpolator1 in the other. Link the cameras.
5. 

ParaView
Adding Text Annotation

1. If you have fallen behind, you can reset ParaView and reload can.ex2.
2. Sources → Text
3. Type a message in text edit box
4. Apply
Text Position

Lower Left Corner

0.32  0.93

Use Window Location
Annotate Time

1. Sources → Annotate Time

2. Apply
Annotate Time

1. Sources → Annotate Time
2. Apply
3. Select can.ex2
4. From quick dialog, select Annotate Time Filter
5. Apply
6. Move annotation around.
Reset ParaView
ParaView on BigRed-II
ParaView Architecture

• Three tier
  – Data Server
  – Render Server
  – Client
Standalone

Client

Data Server  Render Server
Client-Server

Data Server

Render Server

Client
Client-Render Server-Data Server
Requirements for Installing ParaView Server

- C++
- CMake (www.cmake.org)
- MPI
- OpenGL (or Mesa3D www.mesa3d.org)
- Qt 4.6 (optional)
- Python (optional)
  - http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server#Compiling
Get Interactive Compute Node

login% qsub -l -l walltime=1:00:00 \
    -l nodes=1:ppn=32 -l gres=ccm -q cpu \
    -d . generic.pbs

aprun% module load ccm

aprun% ccmlogin
On the compute node

```
nidXXXXXX% module load PrgEnv-gnu/4.1.40
nidXXXXXX% module load paraview
nidXXXXXX% which pvserver
/N/soft/cle4/paraview/gnu/4.0.1/bin/pvserver
nidXXXXXX% aprun <nodes> pvserver -rc
   -ch=<hostname/IP>
```
Connecting to a ParaView Server

[Image of a software interface for connecting to a ParaView Server]

http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server#Running_the_Server
Connecting to a ParaView Server

http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server#Running_the_Server
Image Size LOD

- ParaView’s parallel rendering overhead proportional to image size.
- Can use smaller images for interactive rendering.
Parallel Rendering Parameters

Edit → Settings, Render View → Server
Parameters for Large Data

• Use Immediate Mode Rendering off for GPU, on for CPU.
• Try LOD Threshold off.
  – Also try LOD Resolution 10x10x10.
• Always have remote rendering on.
• Turn on subsampling.
• Image Compression on.
Parameters for Low Bandwidth

• Try larger subsampling rates.
• Try Zlib compression and fewer bits.
Parameters for High Latency

- Turn up Remote Render Threshold
- Make Client Outline Threshold lower than that.
- Play with the LOD Threshold and LOD Resolution to control geometry sent to client.
Further Reading

• [http://www.paraview.org/Wiki/ParaView](http://www.paraview.org/Wiki/ParaView)
• [http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server](http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server)
Further Reading
Visualization and Customization

Further Reading
Parallel VTK Topics


Further Reading
Advanced Pipeline Execution


Further Reading
Parallel Rendering


AVL ParaView tutorial

- Date: TBD
- Topics include:
  - Advanced techniques for exploring data and extracting quantitative information
  - Python calculator / Find Data
  - Python scripting
  - Producing animations
  - Batch mode
  - Stereoscopic and immersive rendering
- **Slower pace for hands-on interaction!**
- Get on the list if you are interested: shermanw@iu.edu