High Performance Scientific computing
Lecture 14

S. Gopalakrishnan

Scientific Visualization

Why Scientific Visualization

Visualization is the process of mapping scientific data into images

Humans understand information in a image much better than we understand a bunch of numbers

Data into knowledge: Visualization presents scientific data to the human visual and cognitive system for analysis and interpretation

Why Scientific Visualization

The goal: extract/communicate knowledge/insight.

Our goal: Convert scientific data into visual form

- Exploration:understand the data

  Interactively examine the data

  Looking for aspects of the data that are interesting

- Communication:communicate the data to peers

  Within the research group
  To external community (presentations,publications)

Why Scientific Visualization

Frequently deal with data that represents a natural phenomenon and has a natural spatial structure

- Medical data (measured – CT, MRI)
- Geo-spatial data (computed/measured)
- Engineering data (computed/measured)
- Chemistry/Biological data (computed/measured)

Exploration: requires interactive visualization so we need to generate images in (almost) real time

Humans have limited tolerance for delay, so good graphics hardware is usually a necessity

Visualization Packages

Most modern visualization packages use a pipelined, component-based architecture (ParaView, MayaVi, VisIT, Avizo)

Users can quickly assemble modular software components into a “finished application.”

Flexible: components can be combined in a multitude of ways, thereby allowing a researcher to accomplish a wide variety of visualization tasks

Extensible: developers can add new components to the system, thereby extending the system’s functionality

Paraview

Project began in 2000

- Kitware, Inc. and Los Alamos National Laboratory.

ParaView is open source

- ParaView is supported by Kitware
- Kitware contributes to ParaView development

ParaView is built on top of the Visualization Toolkit (VTK)

- VTK came out of GE Research

First public release in October 2002: ParaView 0.6
ParaView 3 released in May 2007

http://www.paraview.org/paraview/resources/software.html

Paraview

Paraview provides a full set of tools for manipulating, transforming, processing, rendering and animating data

Allows for visualization and analysis methods based on points, lines, areas, volumes, images or geometric primitives in any combination

Provides powerful parallel execution and advanced display (3D stereoscopic viewing)

Paraview Architecture

ParaView uses a client-server model

- In stand-alone mode the client does all processing on the local machine

The client process is the user interface

- it always runs on a workstation (desktop)

The server does computation

- In a single process on a local/remote machine, or
- In many processes on a cluster
- The server processes can be split into data server and render server

Paraview Architecture

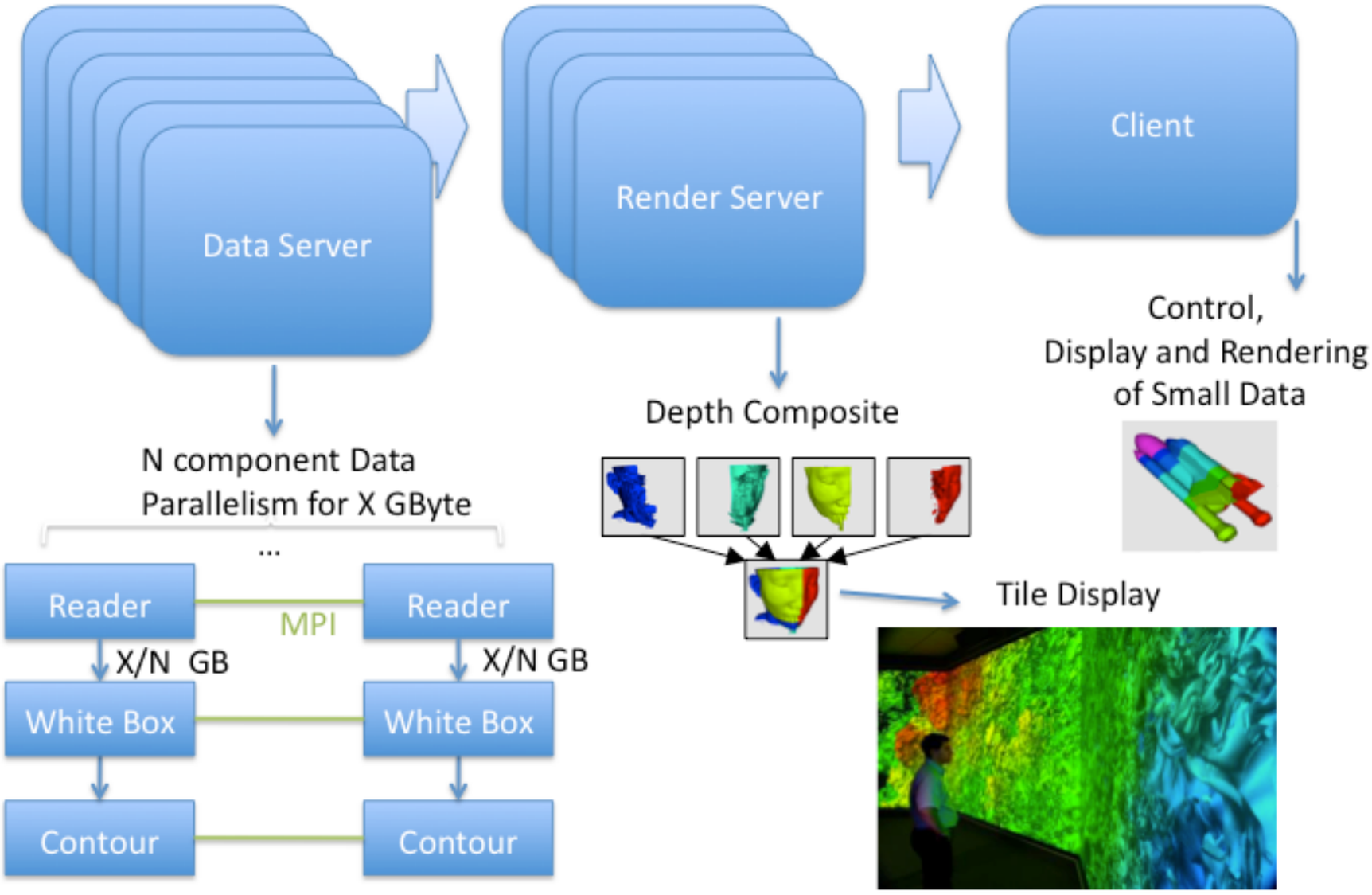


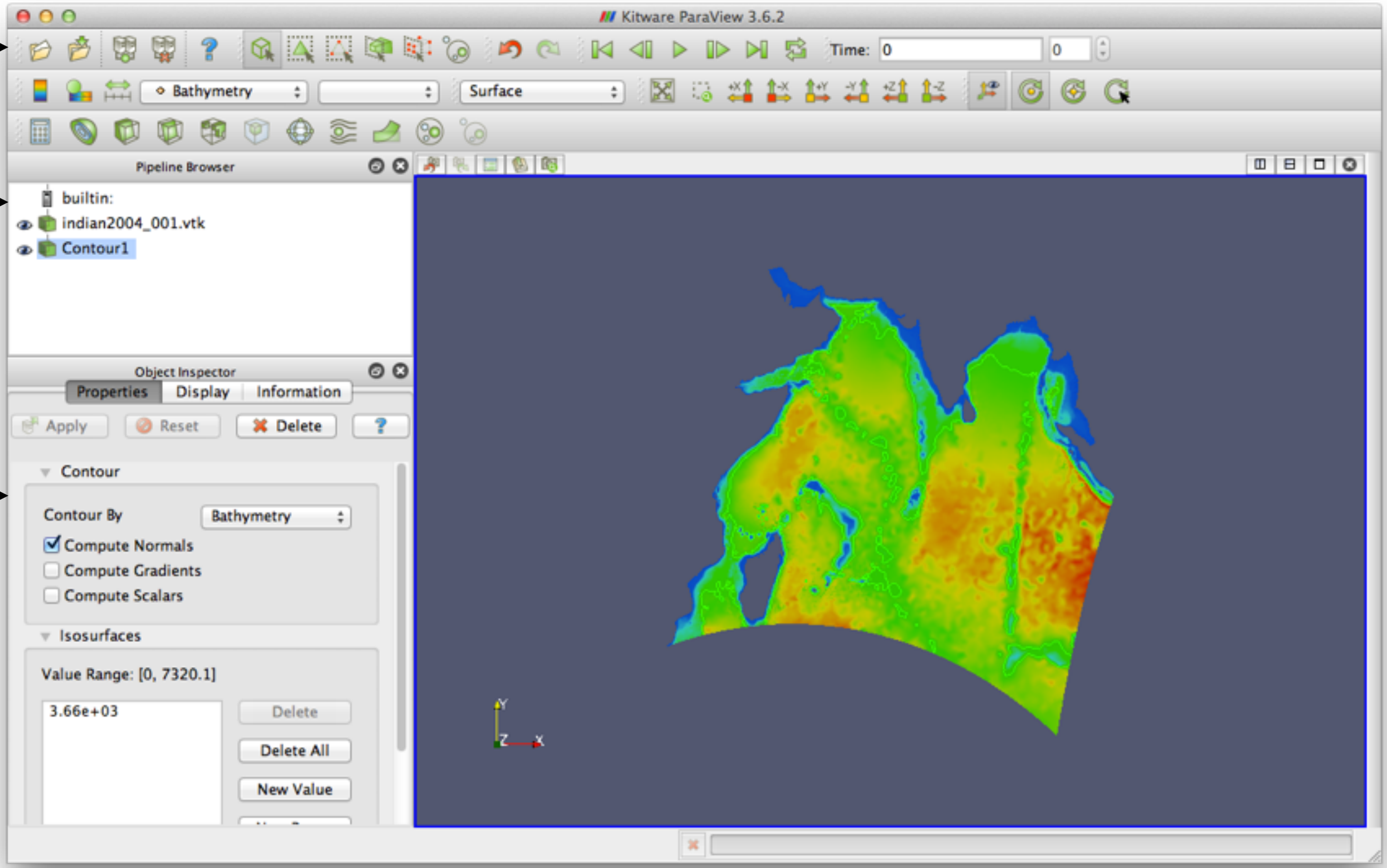
Image: www. paraview.org

Paraview Interface

Toolbars

Pipeline Browser

Object Inspector



Visualization Pipeline

Think of a visualization pipeline in terms of data flow

To process the data we use operators:

- data flows from a source (reader)
- data is modified/manipulated (filters)
- data is transformed to geometry (mappers)
- geometry is rendered

Data travels between operators (connections)

Visualization Pipeline

A data **source** is providing data to the visualization system (e.g. read in a data file)

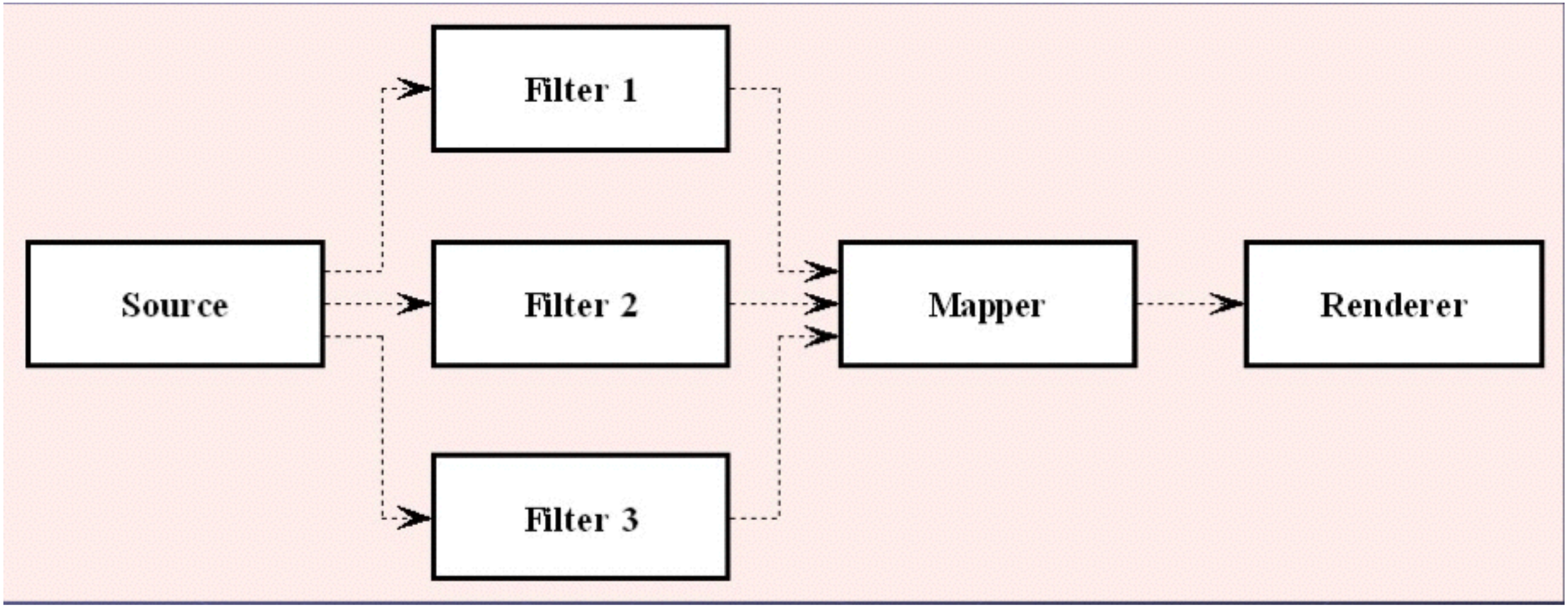
A **filter** takes data from the source, manipulates it in some way and passes it on

- selecting the data of interest...
- transformations,scaling,thresholding,...

A **mapper** takes the data and creates geometric primitives which can then be rendered

- Maps data to geometry(points,lines,polygons)
- Maps data to color,size,etc.

Visualization Pipeline



Paraview

ParaView supports many techniques for generating renderable objects from data.

For scalar data these include:

- Points and glyphs
- Contours and isosurfaces
- Histograms
- Two-dimensional and three-dimensional plots

For vector data

- Arrow plots, streamlines, etc

Paraview

Visualizations can be annotated with:

- Ribbons, tubes, axes, text
- Display of data locations, meshes and boundaries

Data interactions are also supported:

- probing (selecting a location in a volume)
- picking (selecting a location on the surface of an object)
- arbitrary surface and volume sampling
- arbitrary cutting/mapping planes

Paraview

ParaView is built with VTK and understands VTK's data formats

There are 5 "classic" formats (old but common)

STRUCTURED_POINTS

STRUCTURED_GRID

RECTILINEAR_GRID UNSTRUCTURED_GRID

POLYDATA

All of these files will have the extension ".vtk" Also understands VTK XML data formats

Paraview

ParaView can import many common scientific file formats including:

VTK: .vtk, .pvtk, .vtp, .vtu, .vti, .vts, .vtr, .pvtp, .pvtu, .pvti, .pvts, .pvtr , .vtm, .vtmb, .vthb

Paraview: .pvd
Self-describing data formats: HDF5, netCDF

Ensight: .case, .sos
Protein Data Bank: .pdb
Xmol Molecule Files: .xyz
Gaussian Cube Files: .cube
POP Ocean Files: .pop
Images: .png, .tif
RAW (binary): .raw

Can add a custom reader for your own data format