SAND2008-2733P

# Advanced ParaView Visualization

## Duration of Tutorial

Full day

## Level of Tutorial

Intermediate-advanced

## Abstract

ParaView is a powerful open-source turnkey application for analyzing and visualizing scientific data sets ranging from small desktop-sized problems to the worlds largest simulations and is used by numerous government, educational, and commercial institutions throughout the world. Designed to be configurable, extendible, and scalable, ParaView is built upon the Visualization Toolkit (VTK) to allow rapid deployment of visualization components. This tutorial brings together several of those who helped design and build ParaView give visualization researchers and developers detailed guidance on the behavior and abilities of the ParaView application. This knowledge will allow the tutorial participants to solve their unique visualization problems, to modify the ParaView application to their specific problem domains, or to leverage the design into their own applications. A variety of topics will be discussed during the tutorial. Participants will learn how to customize their visualization with selection, scripting, plugins, and vertical applications. The tutorial will feature how ParaView handles time and parallelism. We will also provide information on some new and advanced features including higher order elements, particle visualization, and information visualization.

## Description of Tutorial

For this tutorial we have collected a group of ParaView developers that will give detailed talks on their area of expertise. This tutorial is specifically designed for the VisWeek audience. Each speaker will impart knowledge of features beyond that of the typical ParaView usage. The presentations are chosen to help visualization researchers and developers use ParaView for difficult visualization problems and to allow them to customize ParaView for their own usage. We also hope to share some of ParaView's design specifics to help other developers incorporate these ideas in their own visualization projects.

The tutorial will cover these general topics: Python scripting, customization, selection, time, parallel visualization, higher order elements, particle visualization, and information visualization. In detail, the talks will involve the following.

### *Python Scripting*

There is now built in support for Python scripting within ParaView, and the Python interpreter can be leveraged in two different ways: automating visualization and providing custom visualization algorithms.

In the first mode of Python scripting, a script can automate the tasks of visualization by loading files, establishing visualization pipelines, summarizing results, outputting data, and writing animations. ParaView's Python bindings are implemented such that the execution mode is abstracted away. Whether in standalone mode or client/server mode, whether in serial or parallel mode, the script behaves the same and the same script can be used in all these different settings. The tutorial will describe these Python bindings, give examples on their use, and outline the different modes in which the scripts can be executed.

In the second mode of Python scripting, a script can provide customized processing inside of the visualization pipeline. This scripting works regardless of whether the pipeline is running in serial or parallel mode. The tutorial will describe how to interface with and manipulate the pipeline inputs and outputs as well as how to request pipeline behavior, such as retrieving ghost cells.

## *Customizing ParaView*

ParaView provide multiple mechanisms for customizing its behavior. One of the most powerful ways of doing so is through the use of plugins. Plugins provide a quick and easy mechanism for changing the behavior of ParaView and for deploying new visualization algorithms. In this tutorial we will first describe the basic proxy and introspection mechanism used internally within ParaView. We then discuss how to take VTK components and build them as a ParaView plugin. We then show how to extend the plugin further by customizing the GUI.

We will then discuss how to customize ParaView even further by building a "vertical application." A vertical application is simply a replacement for the ParaView GUI. We will demonstrate how leverage the existing ParaView GUI and server code to quickly build customized, parallel visualization applications.

## *Advanced Selection*

The implementation of selection in ParaView is pervasive. It provides mechanisms for linking, labeling, extraction, plotting, and inspection. Although intuitive, these features can be used together to provide even more powerful interaction with your data. In this part of the tutorial we will discuss tips and tricks you can do with selection.

## *Time in ParaView*

Many data sets have a time component that is critical to understanding them. ParaView has special mechanisms for dealing with these time components, which this tutorial will describe. We will discuss how time is handled in ParaView from the interface level and then dive down do describe how the VTK pipeline handles time in its data. Participants will learn how to create readers that provide temporal data and create filters that can request and process data that varies over time.

## *Parallel Visualization*

ParaView is world renowned for its scalable parallel visualization and rendering algorithms. This tutorial will discuss the parallelism of ParaView from theory to

implementation. We will discuss how to effectively use ParaView in large parallel environments and how to build new components that work in parallel.

### *Higher Order Elements*

Typically, field data is interpolated linearly throughout mesh cells. However, ParaView also supports meshes with position and field data with non-linear interpolants. There are many ways to represent non-linear interpolation, so processing non-linear data generally requires adaptation classes to the data format. In this tutorial we will discuss how meshes with non-linear interpolation can be represented in VTK and how to integrate this type of data into ParaView.

### *Particle Visualization*

Although it is common for simulations to compute on a fixed mesh, there are classes of simulation that compute without a mesh. These simulations typically compute the behavior of finite particles in space. Output of simulations of this type can be tricky to visualize because many of the rendering and filtering features in ParaView are designed to work on a mesh. This part of the tutorial will describe recent work to make particle visualization in ParaView more effective.

### *InfoVis in ParaView*

Sandia National Laboratories and Kitware, Inc. are beginning to incorporate information visualization technologies inside the VTK framework. As part of that work, ParaView is being leveraged to deploy these technologies. This tutorial will provide a quick preview of the upcoming InfoVis features within ParaView and some ideas of how it will be used in the future.

## Outline

1. Python scripting (60 min)
    a. Automated visualization
    b. User-defined parallel processing
2. Customizing ParaView (60 min)
    a. Introspection and proxies
    b. Plugins
    c. GUI customizations
    d. Vertical Applications
3. Advanced Selection (30 min)
4. Time in ParaView (60 min)
    a. Overview of time in ParaView
    b. Time in the VTK pipeline
    c. Temporal filters

5. Parallel Visualization (45 min)
    a. Parallel visualization algorithms
        i. Ghost levels
        ii. Data Distribution
        iii. Synchronization
    b. Parallel Rendering algorithms
6. Higher Order Elements (45 min)
7. Particle Visualization (30 min)
8. InfoVis in ParaView (30 min)

## Tutorial Held Before

This tutorial has never been held before. Although we have had ParaView tutorials in the past, these past tutorials have always been geared toward end users (i.e. non-vis folk). This tutorial focuses on advanced topics that help other visualization researchers and developers learn how to solve new problems.

## Why It's Worthwhile

ParaView is an open-source project with contributors distributed throughout the world. VisWeek provides a unique opportunity to bring together these developers for in-depth lectures on one of the most popular and advanced visualization tools.

## Acknowledgements

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

## Presenters

### *Kenneth Moreland*

kmorel@sandia.gov

Kenneth Moreland is a senior member of technical staff at Sandia National Laboratories, New Mexico. He is a major contributor to large-scale parallel visualization and rendering algorithms within ParaView. Kenneth currently leads Sandia's ParaView development work.

### *Timothy Shead*

tshead@sandia.gov

Timothy Shead works at Sandia National Laboratories, New Mexico. He made major contributions to the design and development of the new GUI and client built for

ParaView 3. Timothy currently leads the development of OverView, an information visualization application build on the ParaView framework.


### *David Thompson*

dcthomp@sandia.gov


David "Butterwaffle" Thompson is a senior member of technical staff at Sandia National Laboratories, California. He has contributed to the design and development of many ParaView features including parallel rendering, higher order elements, and temporal pipeline support. David is currently working to integrate geographic visualization with scientific and information visualization.


### *Utkarsh Ayachit*

utkarsh.ayachit@kitware.com


Utkarsh Ayachit is a member of the research staff at Kitware, Inc. Utkarsh is a major developer on the ParaView project and has helped design and develop just about every part of the client/server and parallel proxy interfaces.


### *John Biddiscombe*

biddisco@cscs.ch


John Biddiscombe works with the visualization group of the Swiss National Supercomputing Center (CSCS). John has integrated VTK and ParaView into his own visualization research and development and has contributed to many parts of the ParaView project including temporal support and particle visualization.