Introduction to ParaView

Scientific Data Analysis and Visualization for Petascale Computing

July 28, 2008

Kenneth Moreland
Sandia National Laboratories
SciDAC Institute for Ultrascale Visualization

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.
More Information

• Online Help

• The ParaView Guide

• The ParaView web page
  – www.paraview.org

• ParaView mailing list
  – paraview@paraview.org
Golevka Asteroid vs. 10 Megaton Explosion

- CTH shock physics, over 1 billion cells
Polar Vortex Breakdown

• SEAM Climate Modeling, 1 billion cells (500 million cells visualized).
Objects-in-Crosswind Fire

- Coupled SIERRA/Fuego/Syrinx/Calore, 10 million unstructured hexahedra
Scripting, Client Side

Client Side Scripting

Client

Python Interpreter

Server

ParaView

UltraVis

SciDAC Institute for Ultrascale Visualization

Sandia National Laboratories
Scripting, Server Side

- Reader
- Programmable Filter
- Python Interpreter
- Element Sizes
- Compute Error

Exact Density

Simulated Density

UltraVis
SciDAC Institute for Ultrascale Visualization
Scripting Scalability
Large Scale AMR
Large Data Fragmentation Analysis
ParaView Architecture

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

Client

Data Server

Render Server

ParaView

UltraVis

SciDAC Institute for Ultrascale Visualization

Sandia National Laboratories
Client-Server

Data Server

Render Server

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

• C++
• CMake ([www.cmake.org](http://www.cmake.org))
• MPI
• OpenGL (or Mesa3D [www.mesa3d.org](http://www.mesa3d.org))
• Qt 4.2.3 – Qt 4.3.X (optional)
• Python (optional)
  • [http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server#Compiling](http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server#Compiling)
Connecting to a ParaView Server

http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server#Running_the_Server
The Parallel Visualization Pipeline

- Read
- Isosurface
- Reflect
- Render
The Parallel Visualization Pipeline

1. Read
2. Isosurface
3. Reflect
4. Render

1. Read
2. Isosurface
3. Reflect
4. Render

1. Read
2. Isosurface
3. Reflect
4. Render

1. Read
2. Isosurface
3. Reflect
4. Render
Data Parallel Pipelines

• Duplicate pipelines run independently on different partitions of data.
Data Parallel Pipelines

• Duplicate pipelines run independently on different partitions of data.
Data Parallel Pipelines

• Some operations will work regardless.
  – Example: Clipping.
Data Parallel Pipelines

• Some operations will work regardless.
  – Example: Clipping.
Data Parallel Pipelines

• Some operations will work regardless.
  – Example: Clipping.
Data Parallel Pipelines

• Some operations will have problems.
  – Example: External Faces
Data Parallel Pipelines

• Some operations will have problems.
  – Example: External Faces
Data Parallel Pipelines

- Ghost cells can solve most of these problems.
Data Parallel Pipelines

• Ghost cells can solve most of these problems.
Data Partitioning

• Partitions should be load balanced and spatially coherent.
Data Partitioning

- Partitions should be load balanced and spatially coherent.
Data Partitioning

- Partitions should be load balanced and spatially coherent.
Load Balancing/Ghost Cells

• Automatic for Structured Meshes.
• Partitioning/ghost cells for unstructured is “manual.”
• Use the D3 filter for unstructured
  – (Filters → Alphabetical → D3)
The Parallel Visualization Pipeline

- **Read**
  - **Isosurface**
  - **Reflect**
  - **Render**

- **Read**
  - **Isosurface**
  - **Reflect**
  - **Render**

- **Read**
  - **Isosurface**
  - **Reflect**
  - **Render**

- **Read**
  - **Isosurface**
  - **Reflect**
  - **Render**
Parallel Rendering
Parallel Rendering
Rendering Modes

• Still Render
  – Full detail render.

• Interactive Render
  – Sacrifices detail for speed.
  – Provides quick rendering rate.
  – Used when interacting with 3D view.
Level of Detail (LOD)

• Geometric decimation.
• Used only with Interactive Render

Original Data  Divisions: 50x50x50  Divisions: 10x10x10
Image Size LOD

- ParaView’s parallel rendering overhead proportional to image size.
- To speed up interactive rendering, ParaView can render smaller sized images and inflate them.

Original Data | Subsample Rate: 2 pixels | Subsample Rate: 4 pixels | Subsample Rate: 8 pixels
Color Depth LOD

- Squirt is used to transfer images from server to client.
- Squirt is a run length encoder optimized for images.
- Run lengths improved by masking out some color bits.

24-bit mask  
19-bit mask  
10-bit mask
Parameters for Large Data

• Use Immediate Mode Rendering on.
• Use Triangle Strips off.
• Try LOD Threshold off.
  – Also try LOD Resolution 10x10x10.
• Always have remote rendering on.
• Turn on subsampling.
  – Try larger subsampling rates.
• Squirt Compression on.
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

