

# Introduction to ParaView

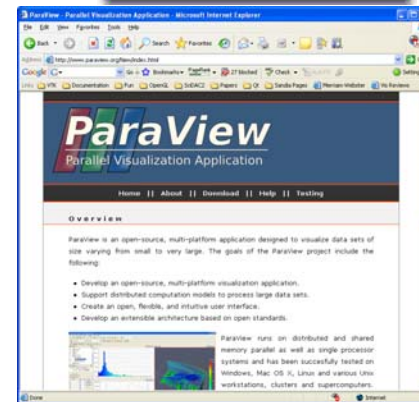
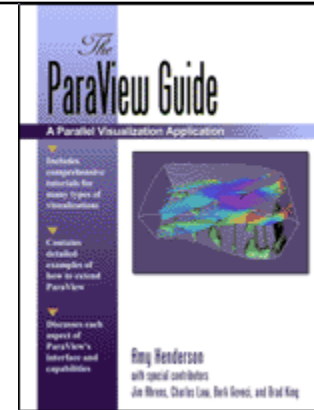
## Scientific Data Analysis and Visualization for Petascale Computing

July 28, 2008

Kenneth Moreland  
Sandia National Laboratories  
SciDAC Institute for Ultrascale Visualization

# More Information

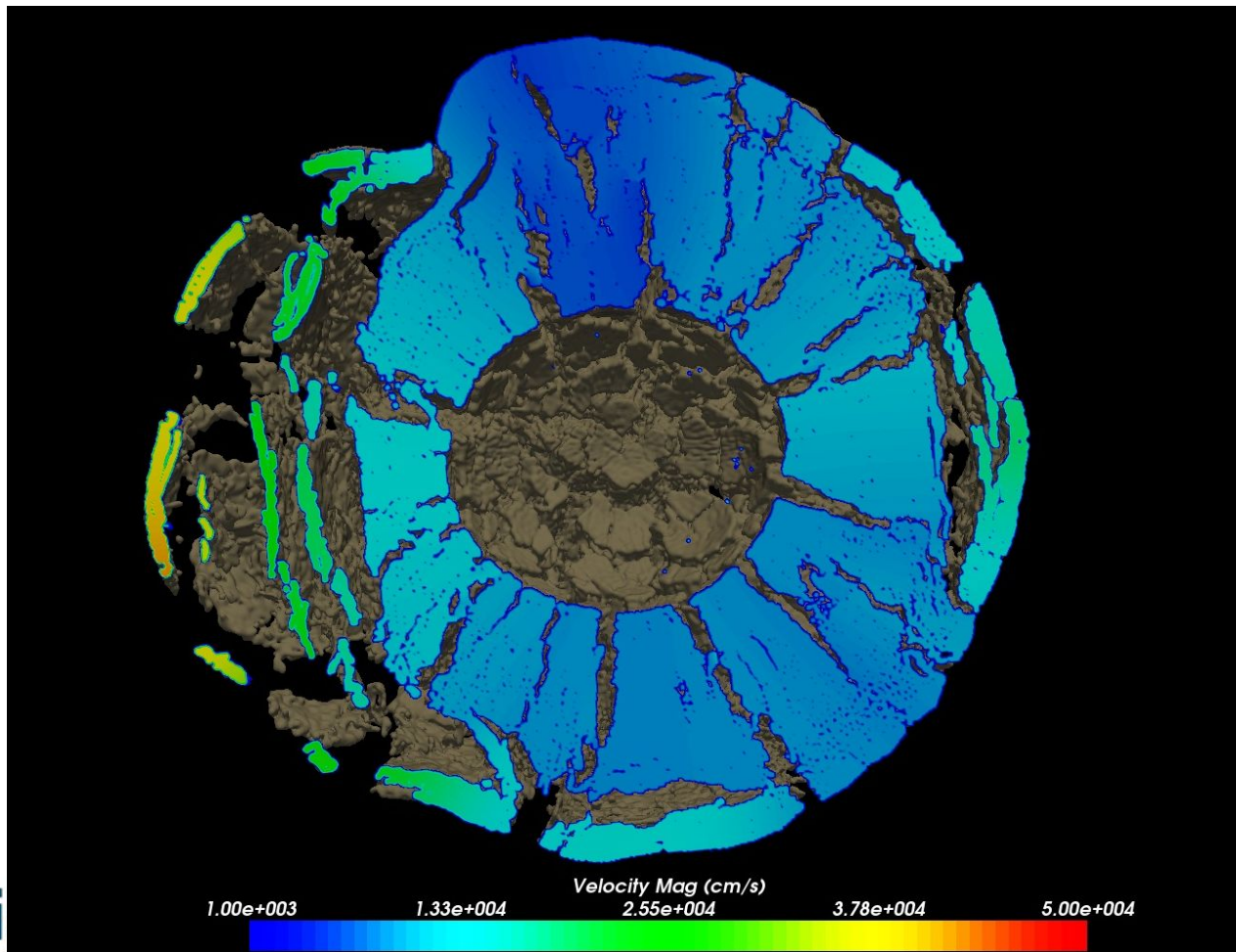
- Online Help ?
- *The ParaView Guide*
- The ParaView web page
  - [www.paraview.org](http://www.paraview.org)
- ParaView mailing list
  - [paraview@paraview.org](mailto:paraview@paraview.org)



# Golevka Asteroid vs. 10 Megaton Explosion

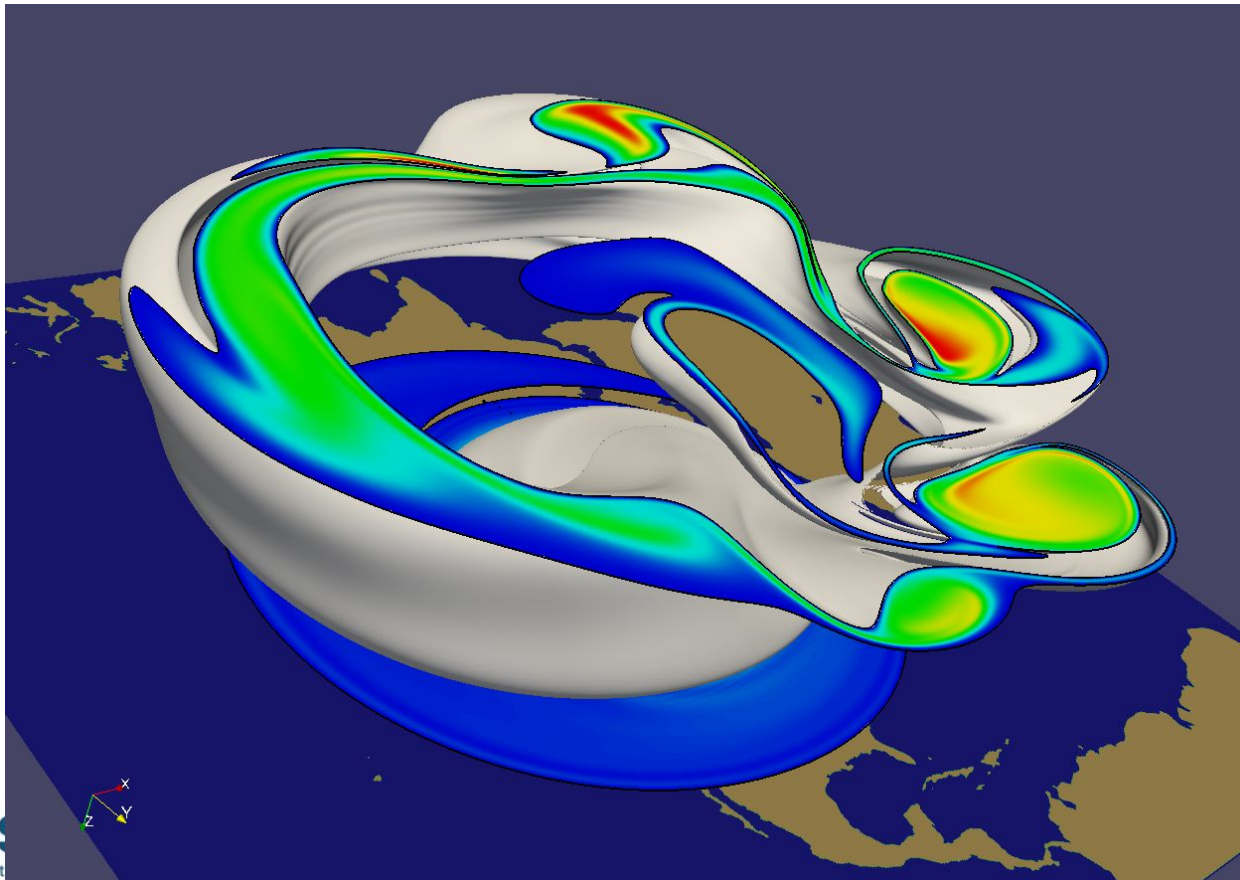


- CTH shock physics, over 1 billion cells



# Polar Vortex Breakdown

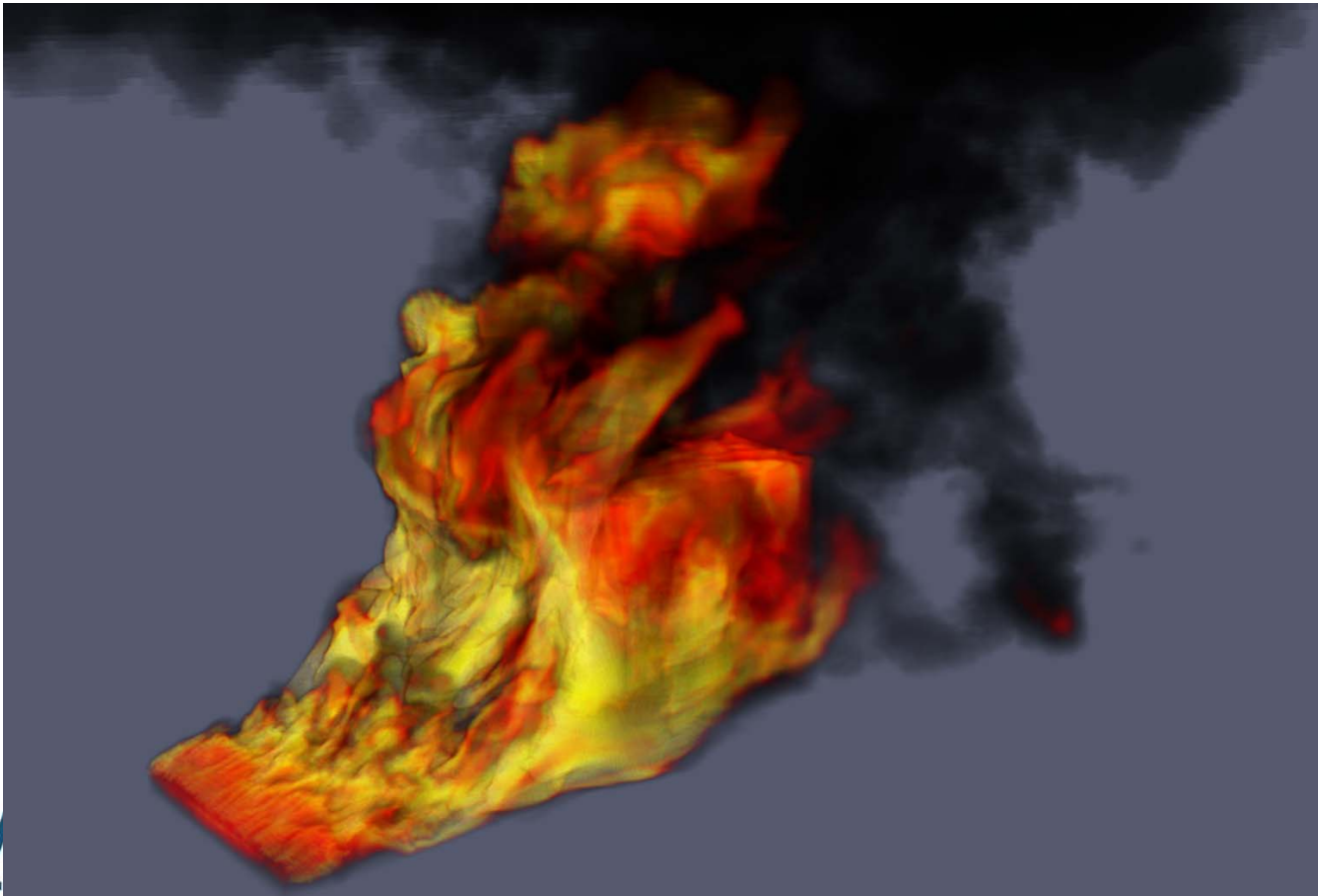
- SEAM Climate Modeling, 1 billion cells (500 million cells visualized).



# Objects-in-Crosswind Fire

---

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



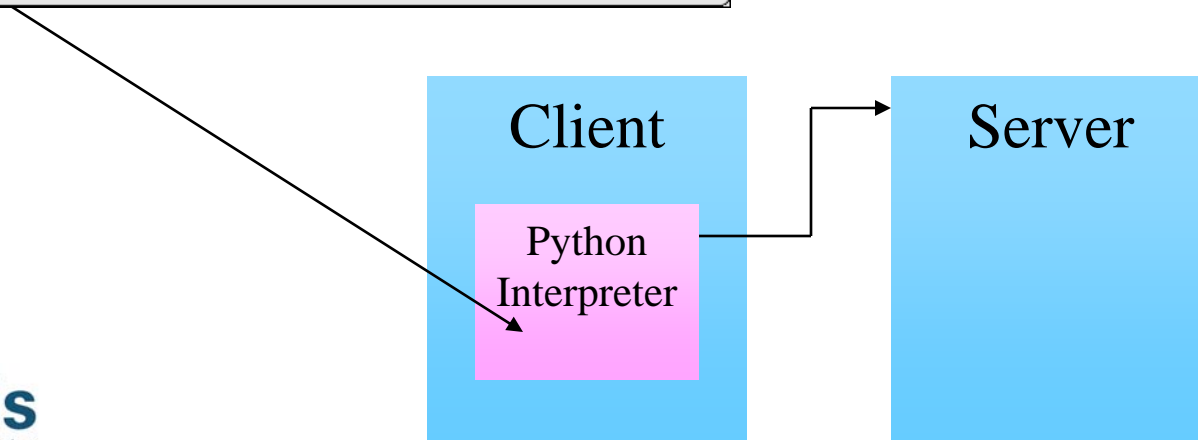


# Scripting, Client Side

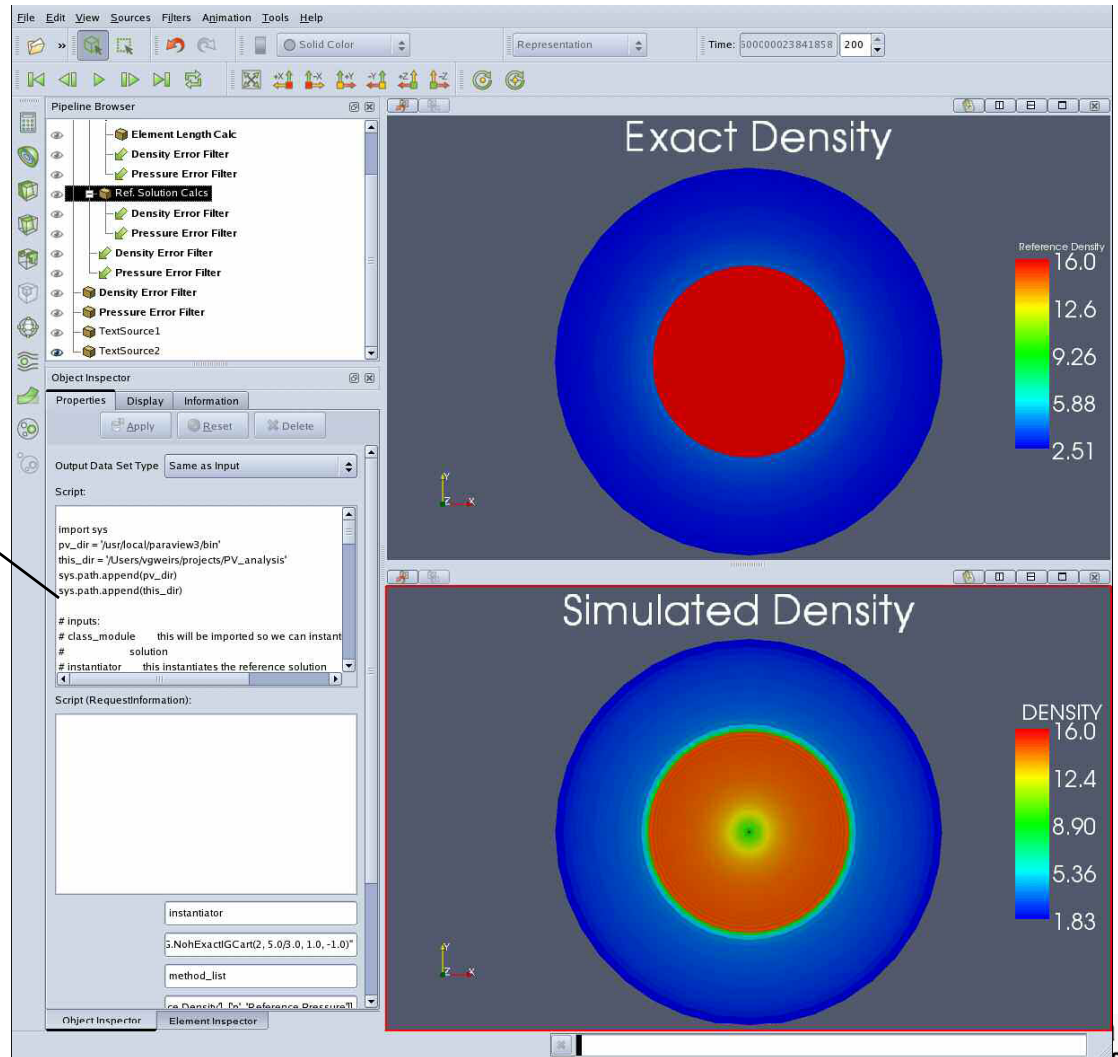
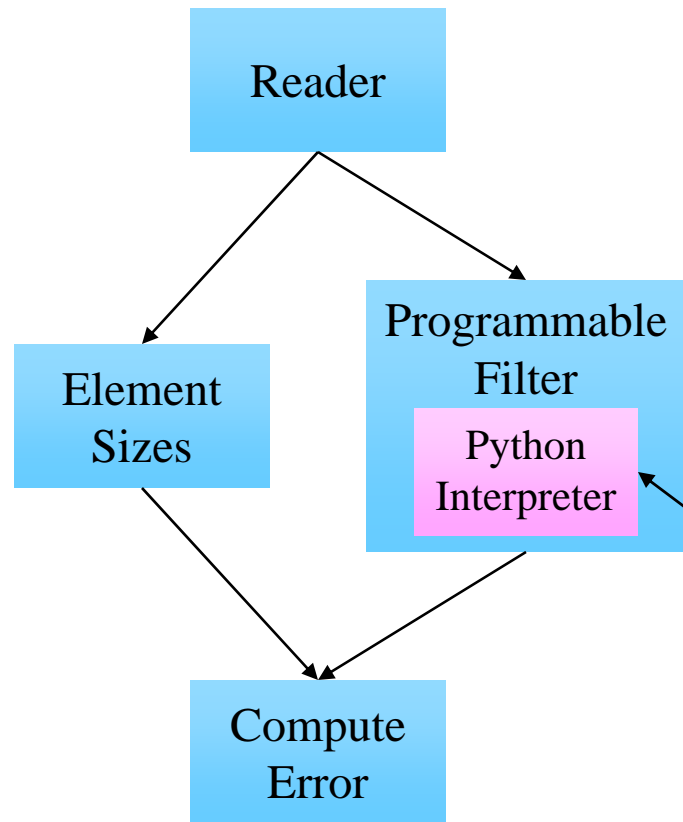
```

Python Shell
Python 2.3.4 (#1, Feb 6 2006, 10:38:45)
[GCC 3.4.5 20051201 (Red Hat 3.4.5-2)] on linux2
>>>
>>> import sys
>>> sys.path.append('/home/vgweirs/projects/paraview/PV_analysis')
>>> import proto_script
Number of elements = 13824000
applying operation
Domain Volume = 7.99904203415
applying operation
Sum of Lengths = 115358.40625
Characteristic Length = 0.00834479211878
applying operation
LI Density Error = 0.0
applying operation
>>>
    
```

Run Script    Clear    Close



# Scripting, Server Side



# Scripting Scalability

The screenshot displays the ParaView 3.0.2 interface with the following components:

- Top Panel:** File, Edit, View, Sources, Filters, Animation, Tools, Help. ProcessId and Surface are selected. Time: 500000023841858, 200.
- Pipeline Browser:**
  - csrc/localhost
  - NohSpherical3D.exe.216.000
  - Volume Calc
  - Element Length Calc
  - Density Error Filter
  - Pressure Error Filter
  - Ref. Solution Cals
  - Density Error Filter
  - Pressure Error Filter
  - Density Error Filter
  - Pressure Error Filter
  - Density Error Filter
  - Pressure Error Filter
  - ProcessIdScalars1 (selected)
  - Cut1
  - Pressure Error Filter
- Object Inspector:**
  - Properties, Display, Information
  - Statistics:
    - Type: Unstructured Grid
    - Number of Cells: 13824000
    - Number of Points: 14886936
    - Memory: 2.19e+03 MB
  - Data Arrays:
 

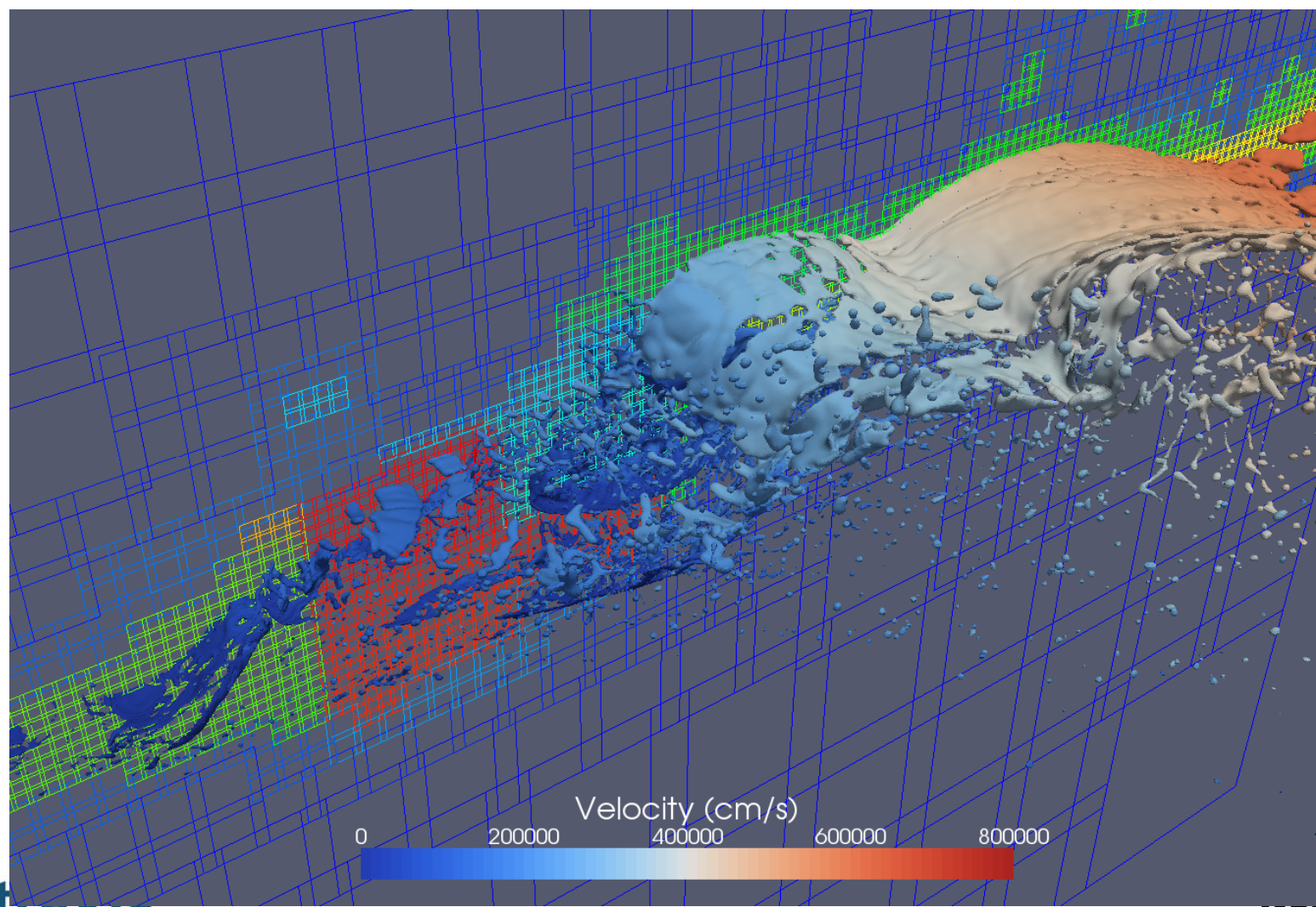
Name	Data Type	Data Ranges
ProcessId	int	[0, 107]
Weighted Density Error	float	[4.22498e-14, 4.1...]
  - Bounds:
    - X range: -0.655 to 0.655 (delta: 1.309)
    - Y range: -0.655 to 0.655 (delta: 1.309)
    - Z range: -0.655 to 0.655 (delta: 1.309)
  - Time:
 

Index	Value
- Visualization:**
  - Top-left: Weighted Density Error plot with a color scale from 1.39e-12 to 4.20e-06.
  - Bottom-left: Weighted Density Error plot with a color scale from 1.39e-12 to 4.20e-06.
  - Right: ProcessId visualization with a color scale from 0.00 to 107. Text overlay: "13 Million Element Mesh 108 Processor Visualization Colored by Processor ID".



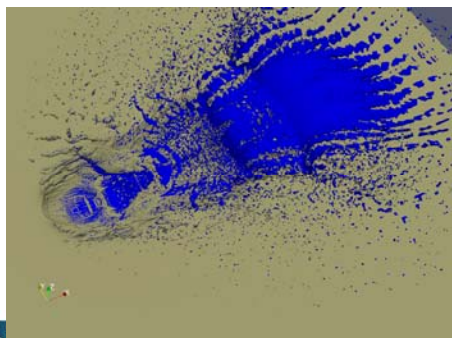
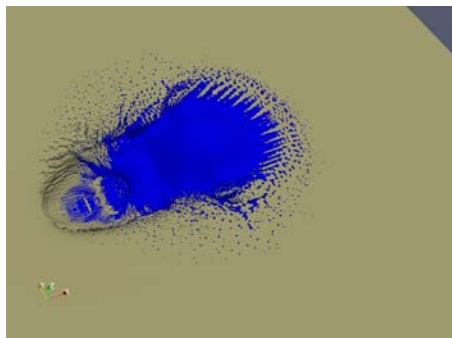
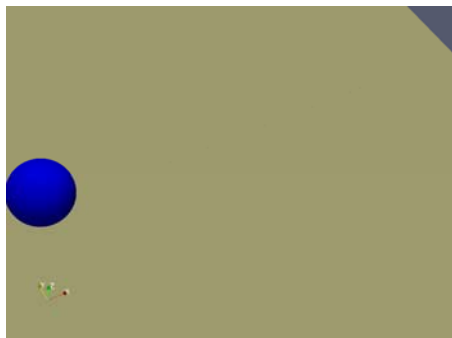


# Large Scale AMR



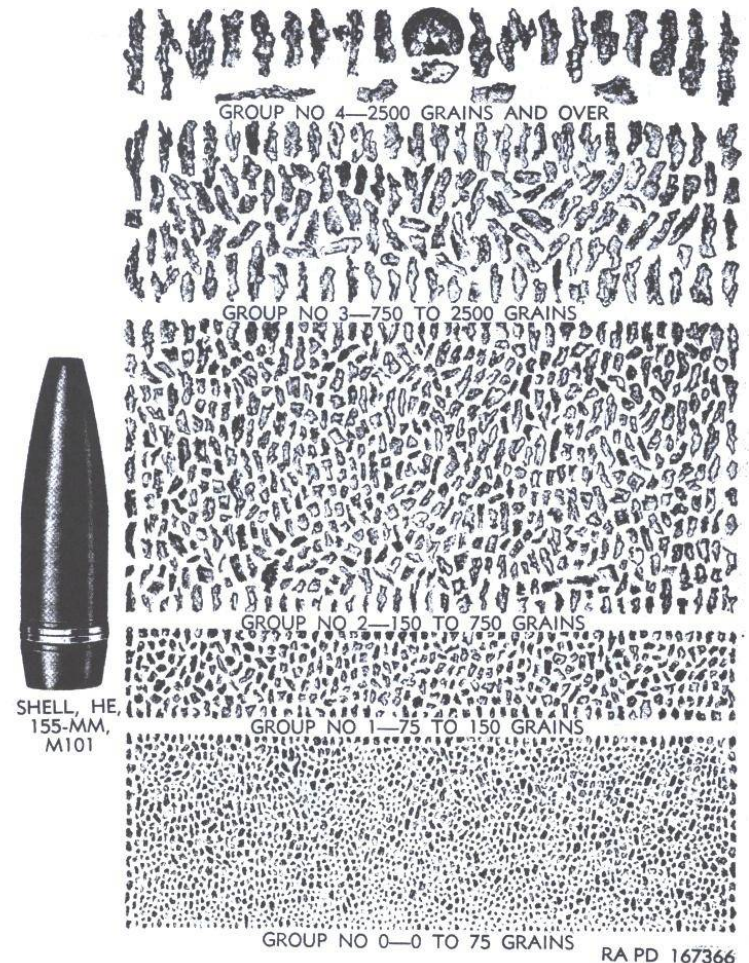
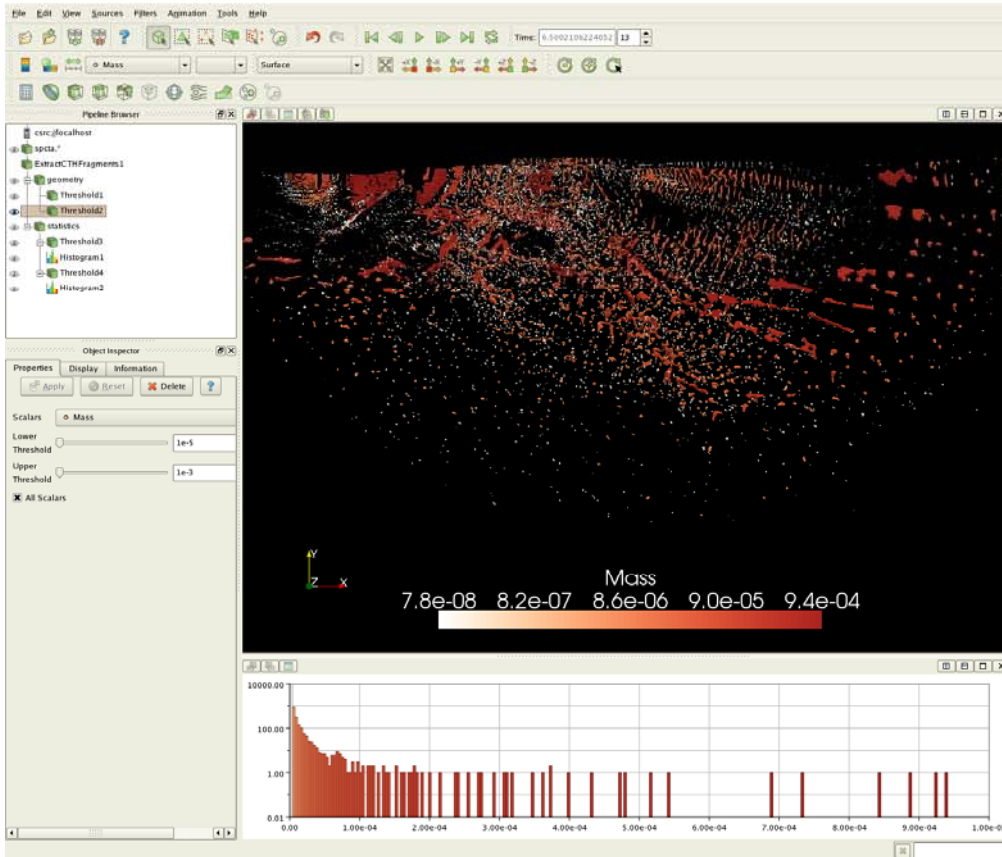


# Large Data Fragmentation Analysis



QuickTime™ and a decompressor are needed to see this picture.

# Large Data Fragmentation Analysis





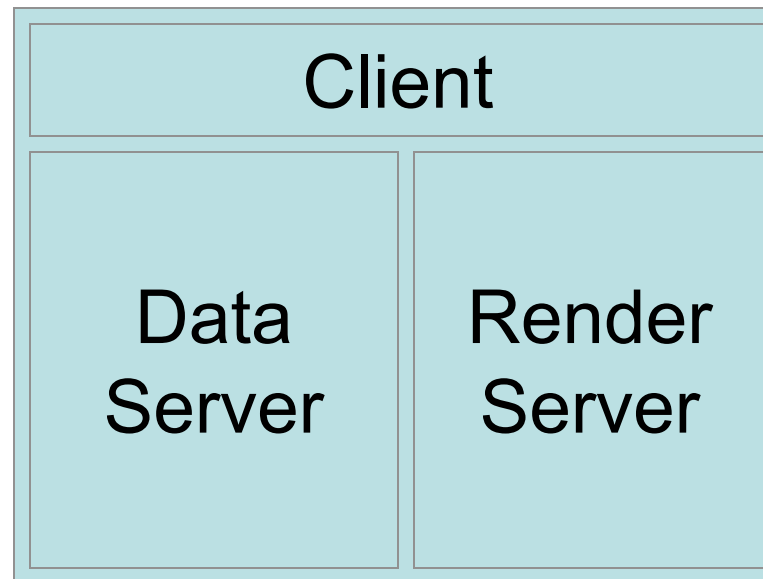
# ParaView Architecture

---

- Three tier
  - Data Server
  - Render Server
  - Client

# Standalone

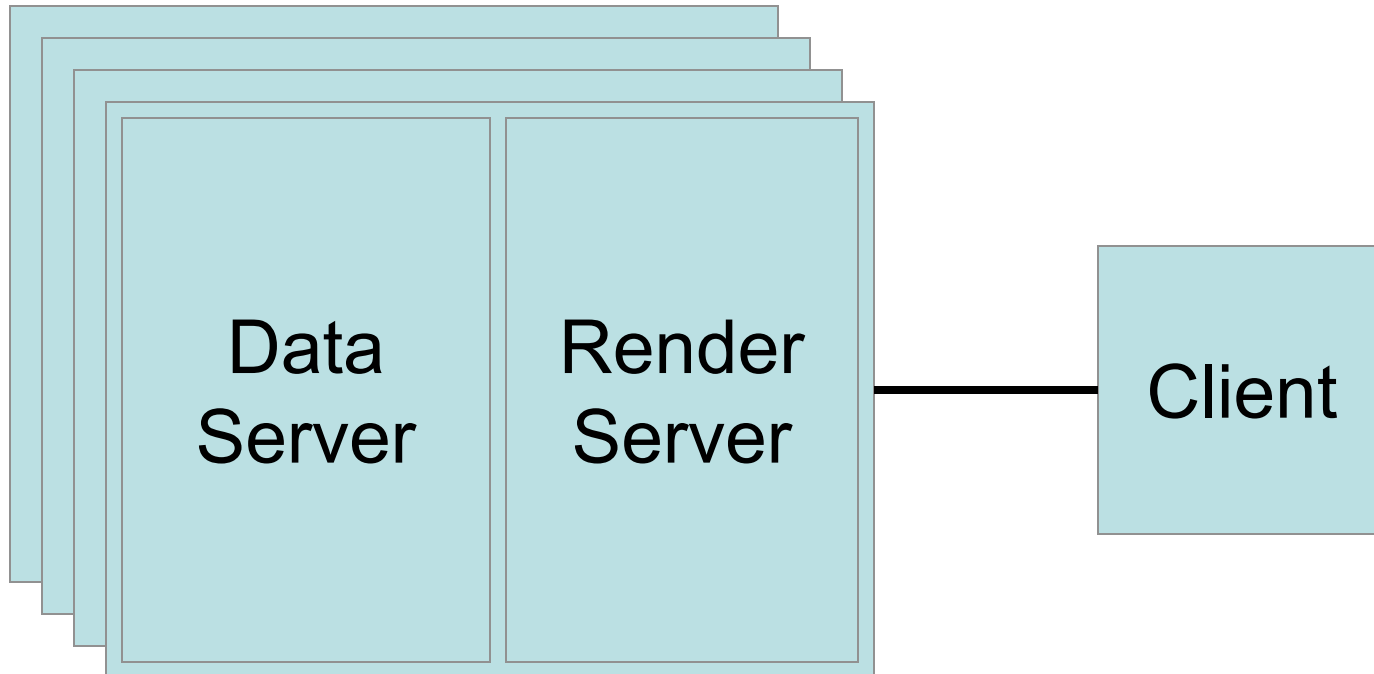
---





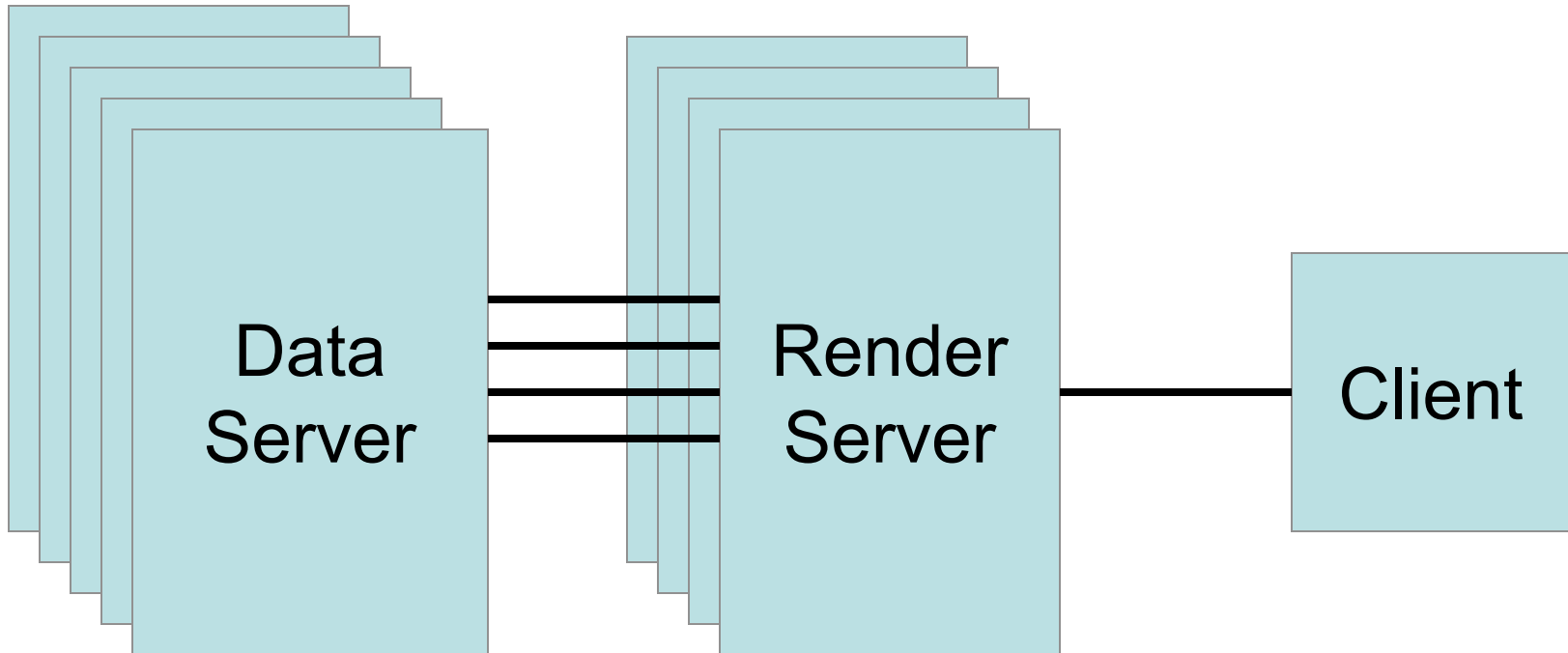
# Client-Server

---



# Client-Render Server-Data Server

---





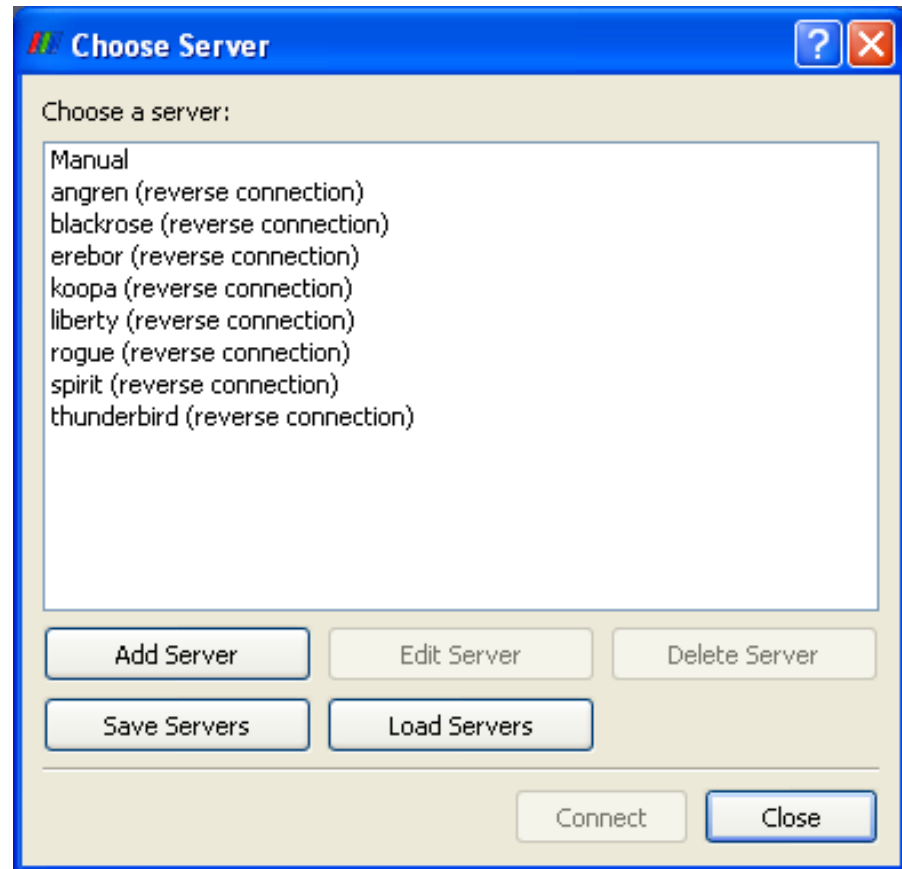
# Requirements for Installing ParaView Server

---



- 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

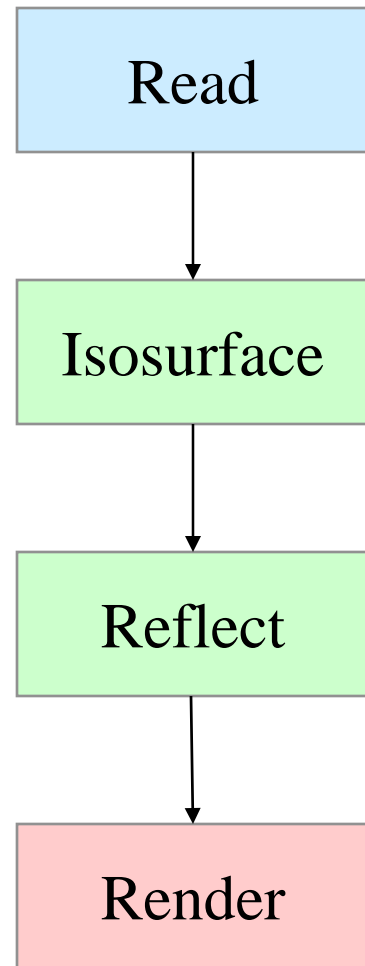


[http://www.paraview.org/Wiki/Setting\\_up\\_a\\_ParaView\\_Server#Running\\_the\\_Server](http://www.paraview.org/Wiki/Setting_up_a_ParaView_Server#Running_the_Server)



# The Parallel Visualization Pipeline

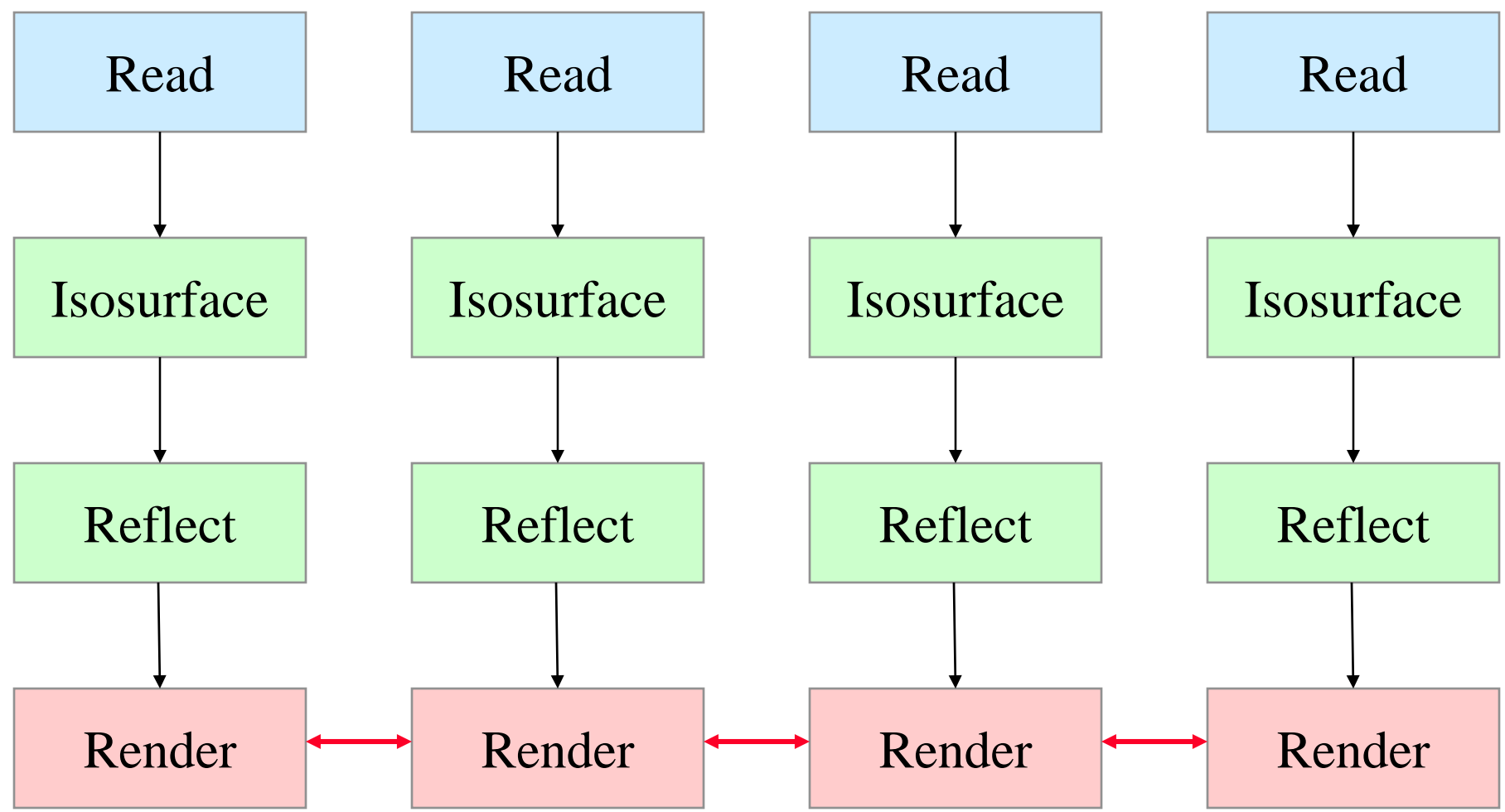
---







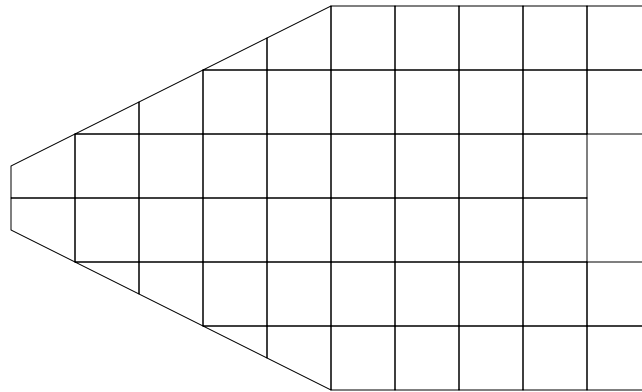
# The Parallel Visualization Pipeline



# Data Parallel Pipelines

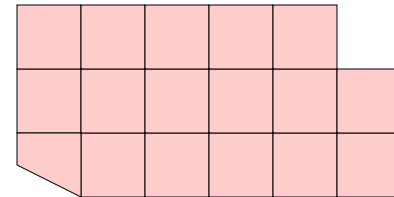
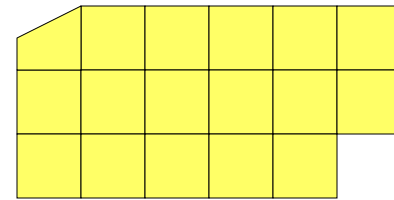
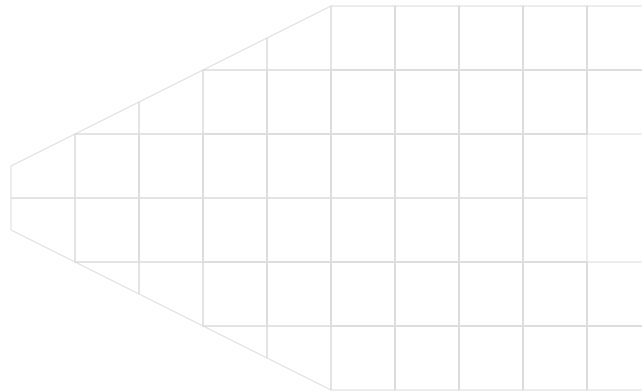
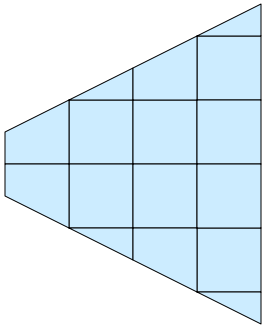
---

- Duplicate pipelines run independently on different partitions of data.



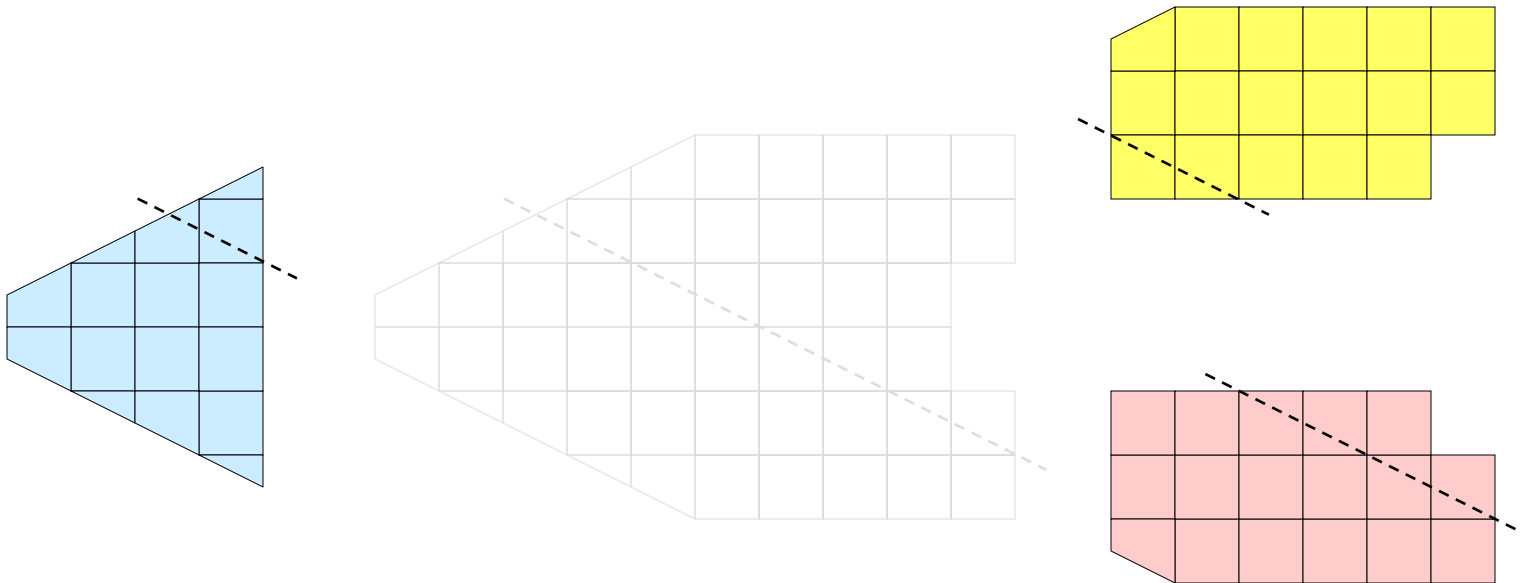
# Data Parallel Pipelines

- Duplicate pipelines run independently on different partitions of data.



# Data Parallel Pipelines

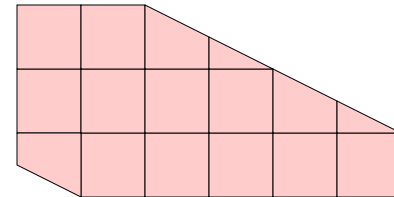
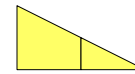
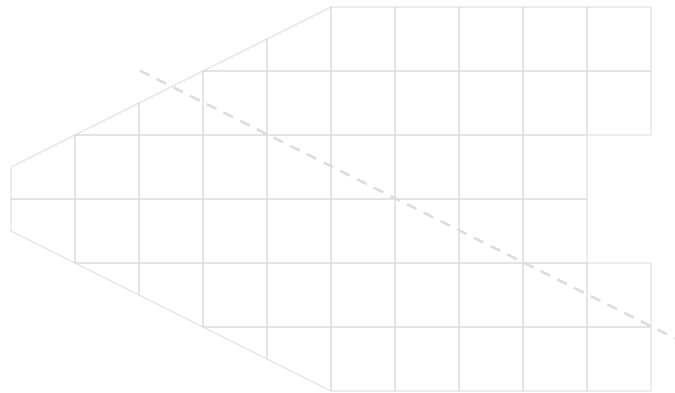
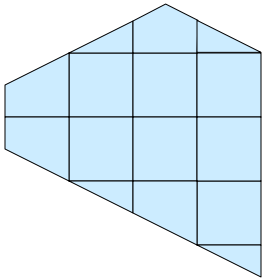
- Some operations will work regardless.
  - Example: Clipping.



# Data Parallel Pipelines

---

- Some operations will work regardless.
  - Example: Clipping.

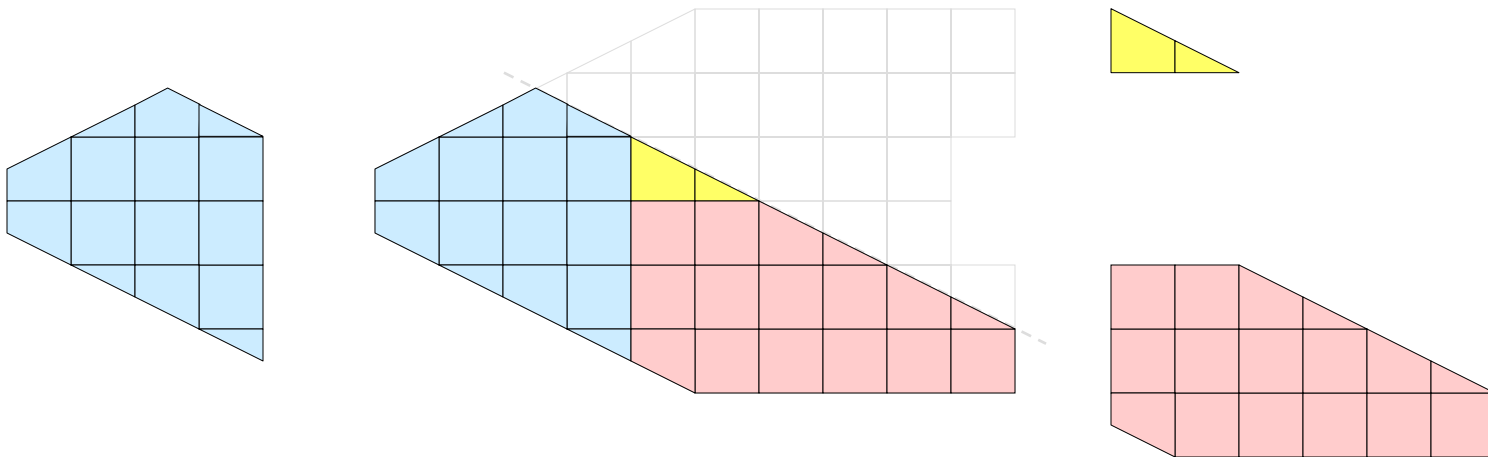




# Data Parallel Pipelines

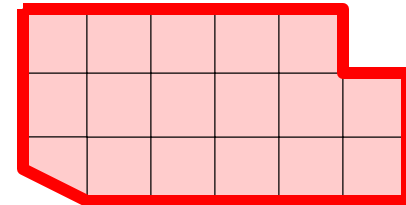
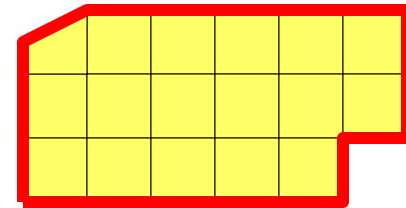
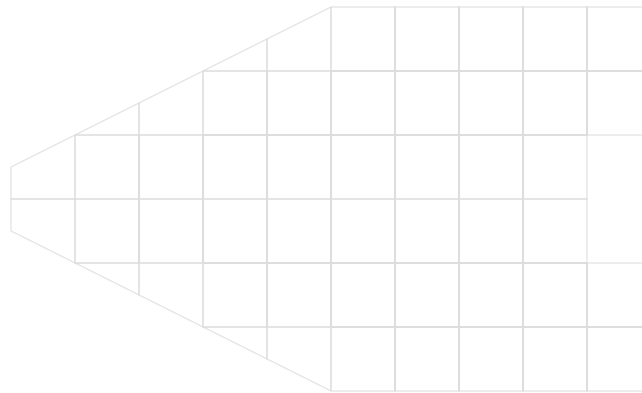
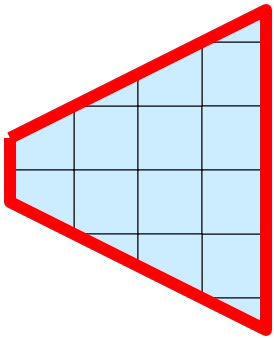
---

- Some operations will work regardless.
  - Example: Clipping.



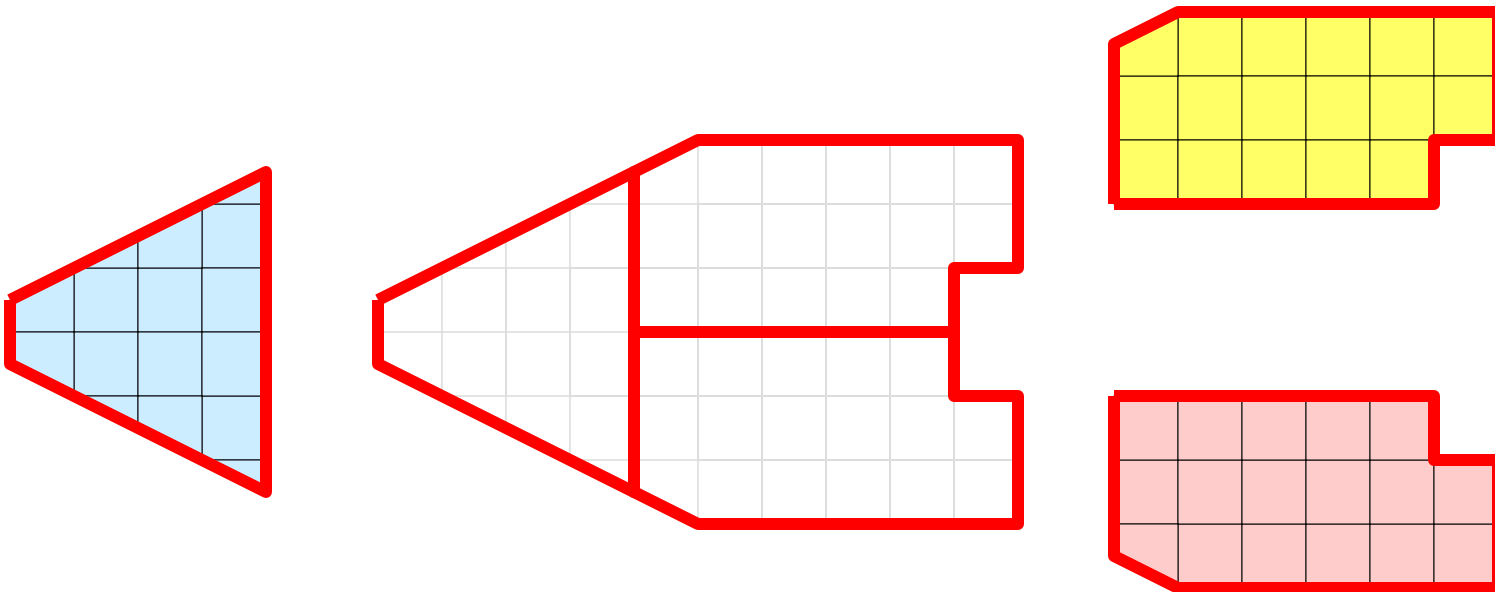
# Data Parallel Pipelines

- Some operations will have problems.
  - Example: External Faces



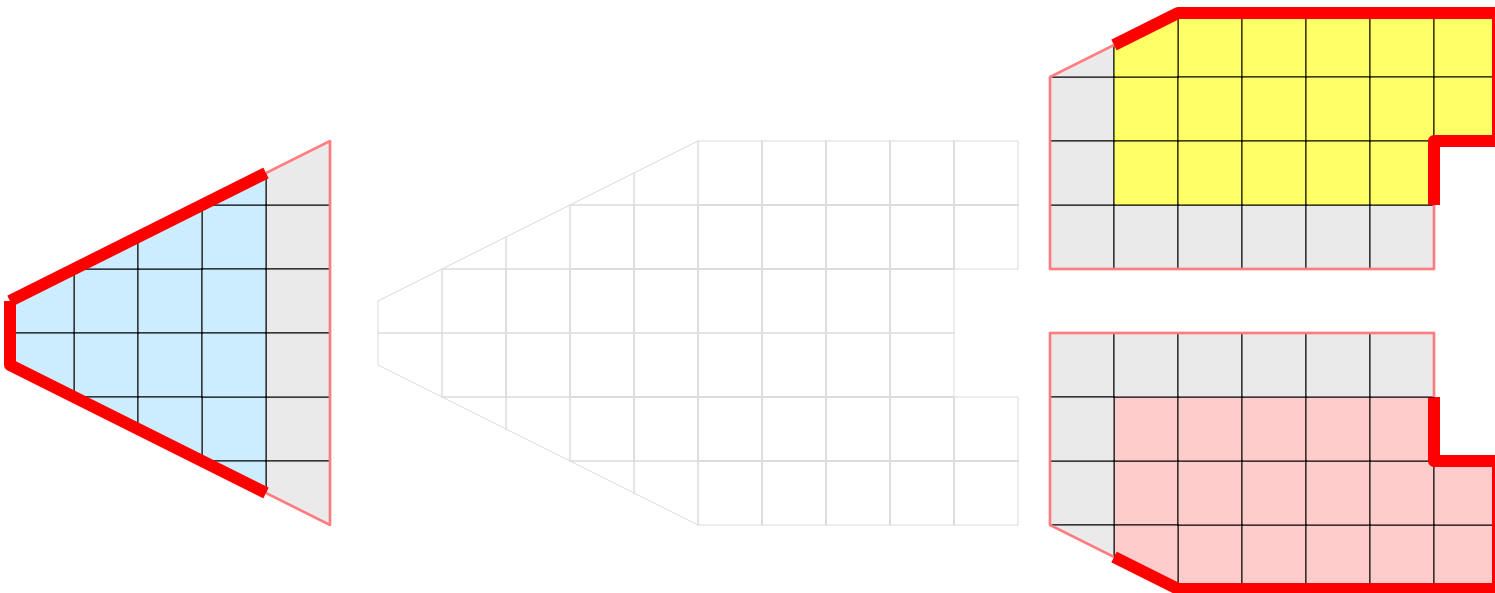
# Data Parallel Pipelines

- Some operations will have problems.
  - Example: External Faces



# Data Parallel Pipelines

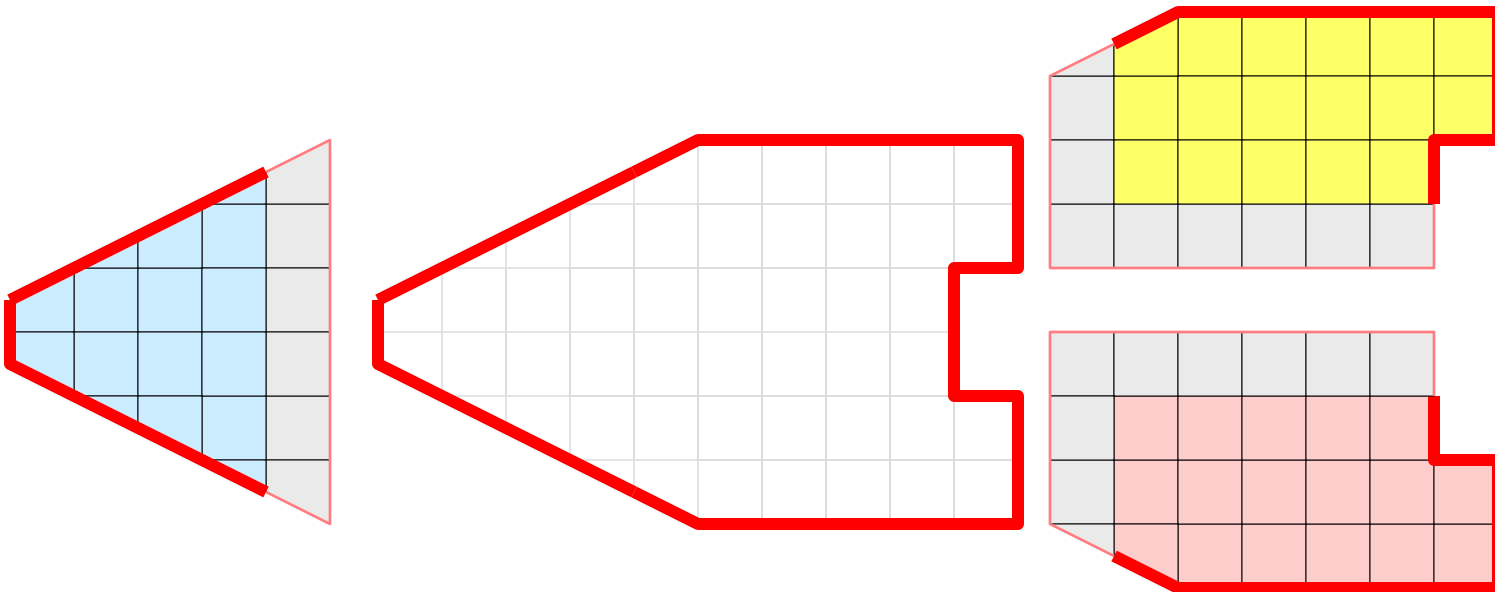
- Ghost cells can solve most of these problems.



# Data Parallel Pipelines

---

- Ghost cells can solve most of these problems.

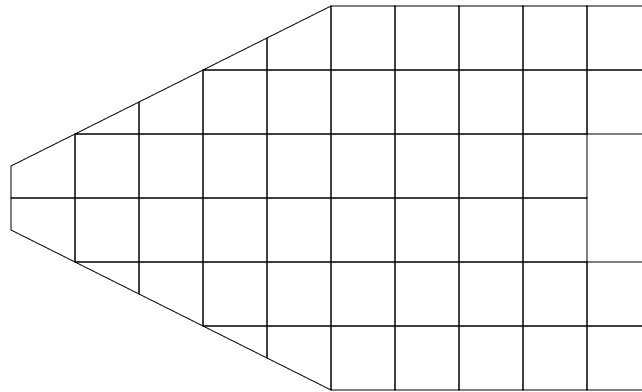




# Data Partitioning

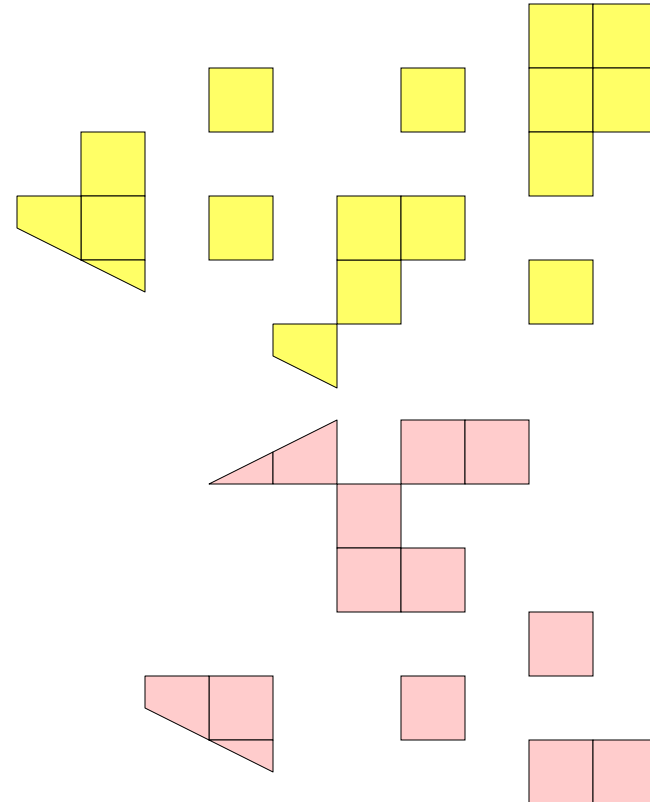
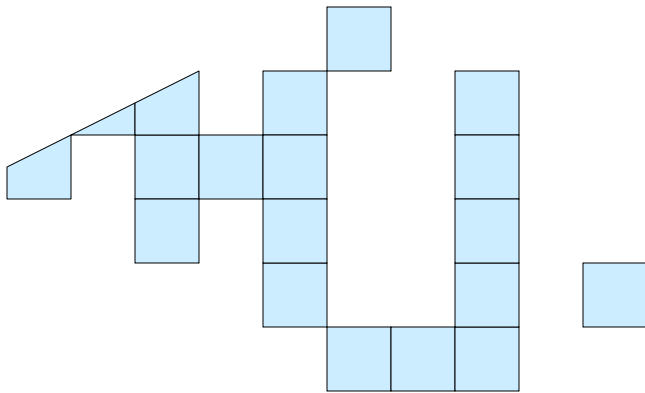
---

- Partitions should be load balanced and spatially coherent.



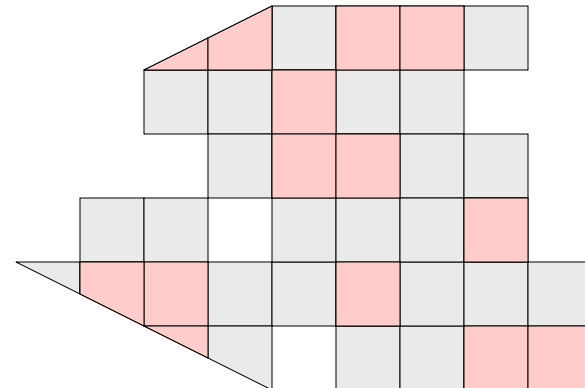
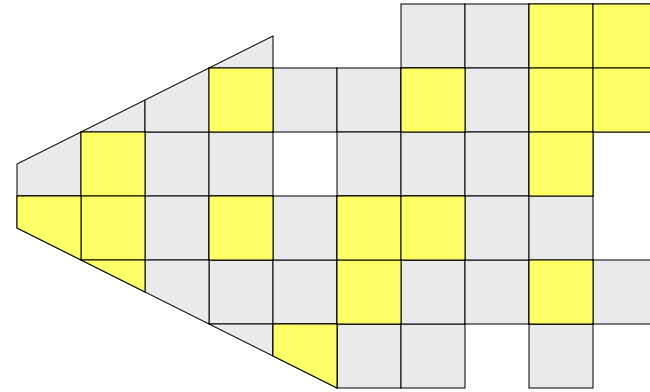
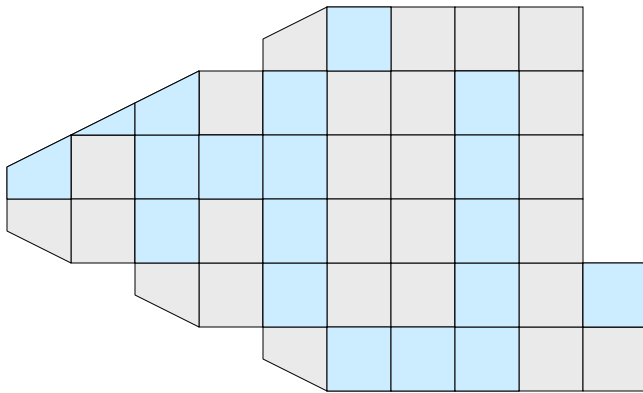
# Data Partitioning

- Partitions should be load balanced and spatially coherent.



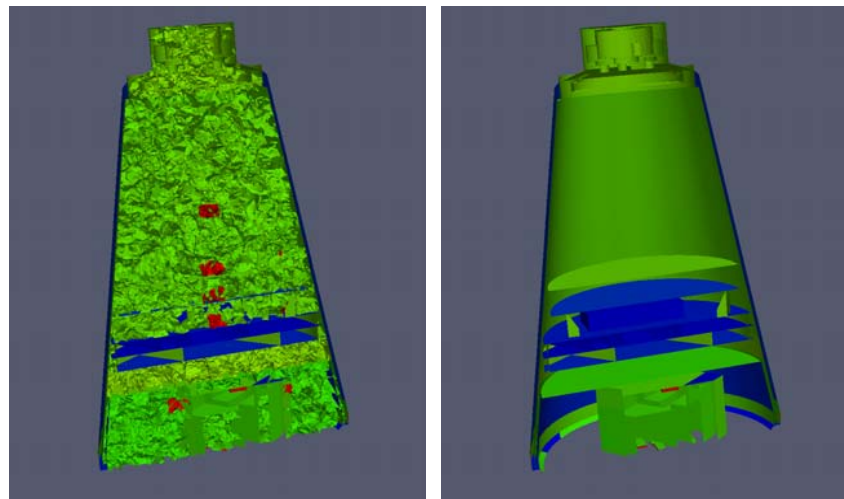
# Data Partitioning

- Partitions should be load balanced and spatially coherent.



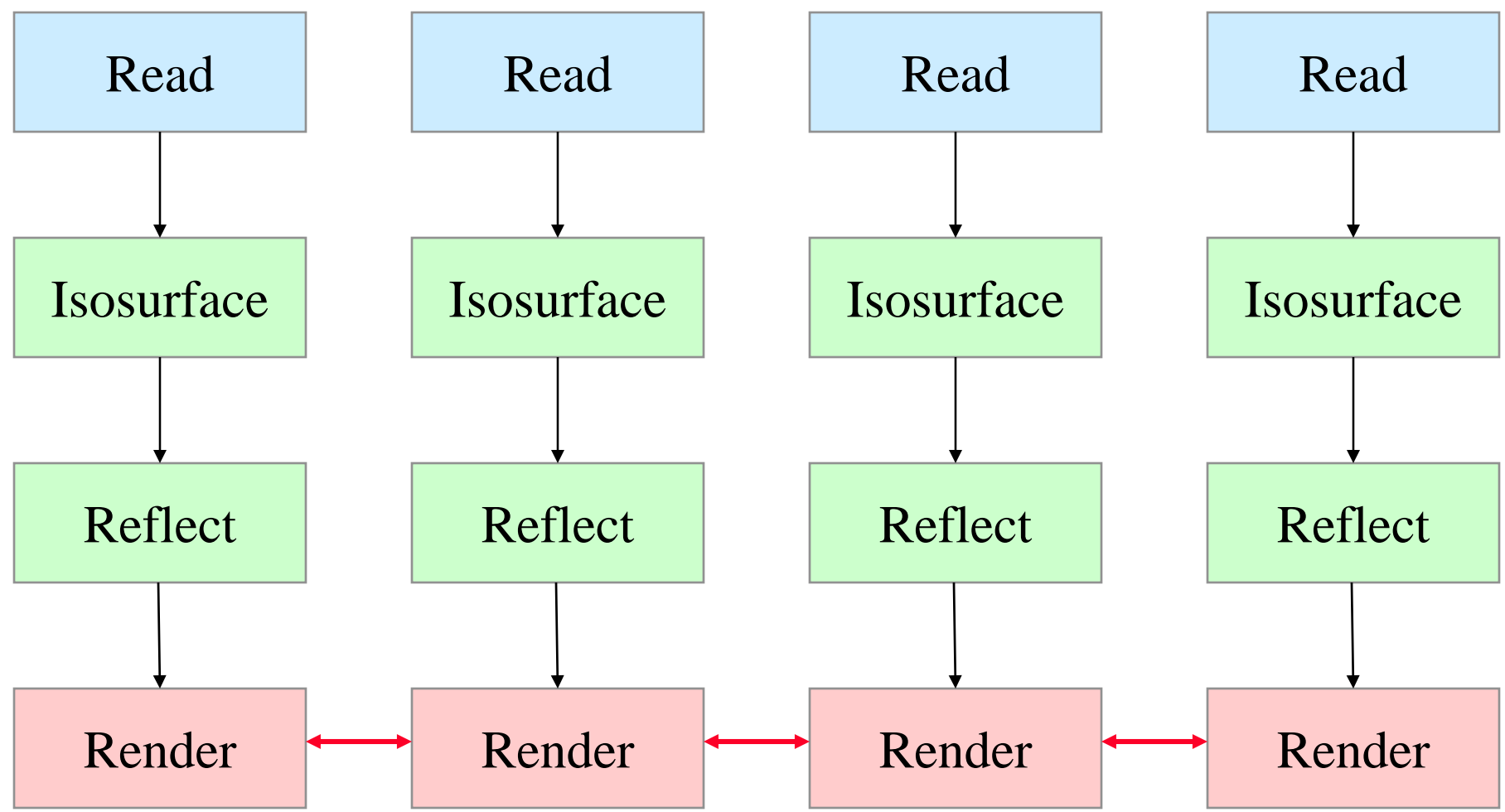
# Load Balancing/Ghost Cells

- Automatic for Structured Meshes.
- Partitioning/ghost cells for unstructured is “manual.”
- Use the D3 filter for unstructured
  - (Filters → Alphabetical → D3)



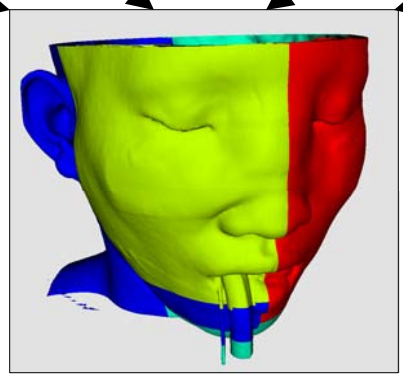
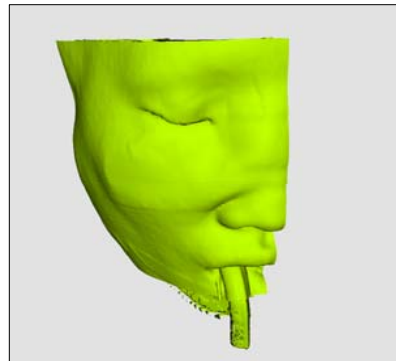
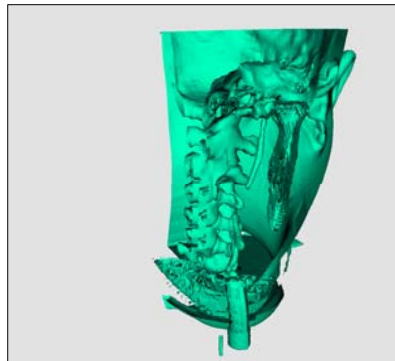
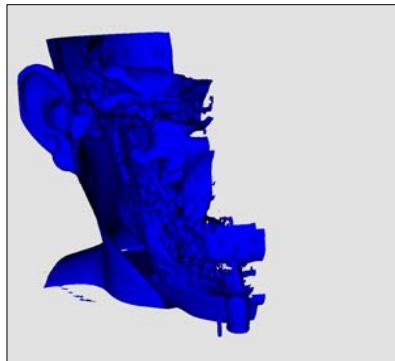


# The Parallel Visualization Pipeline

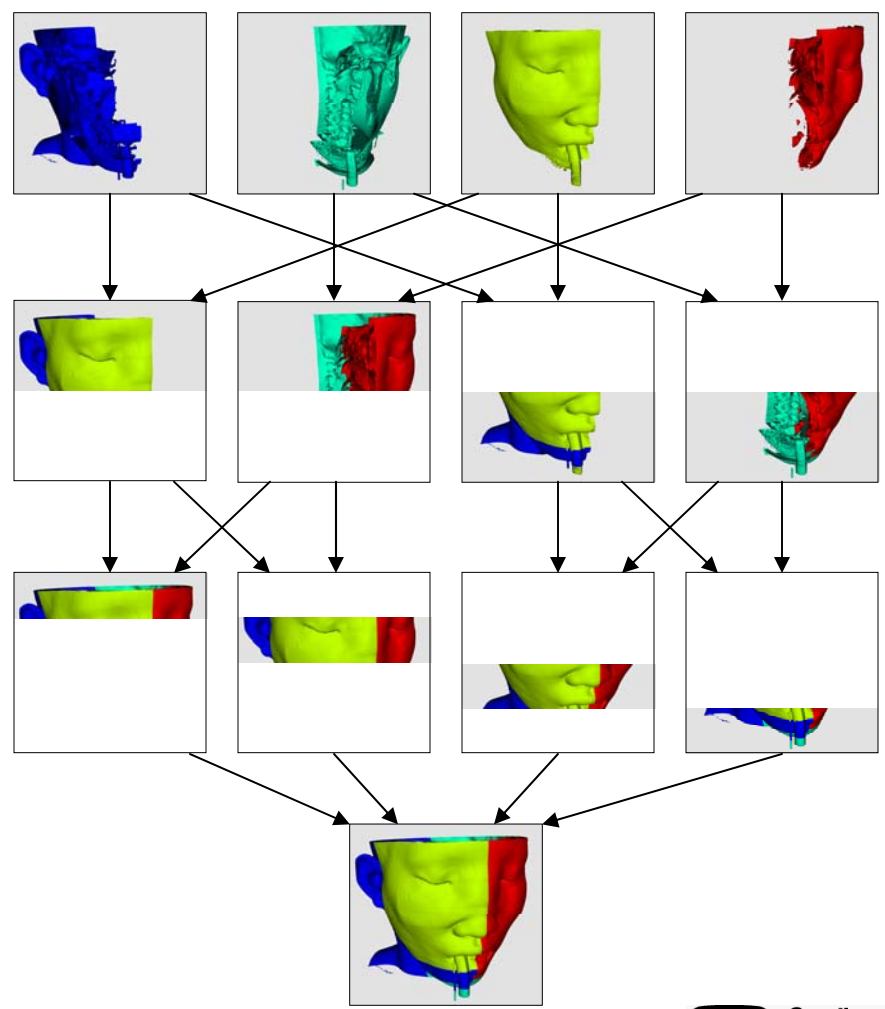
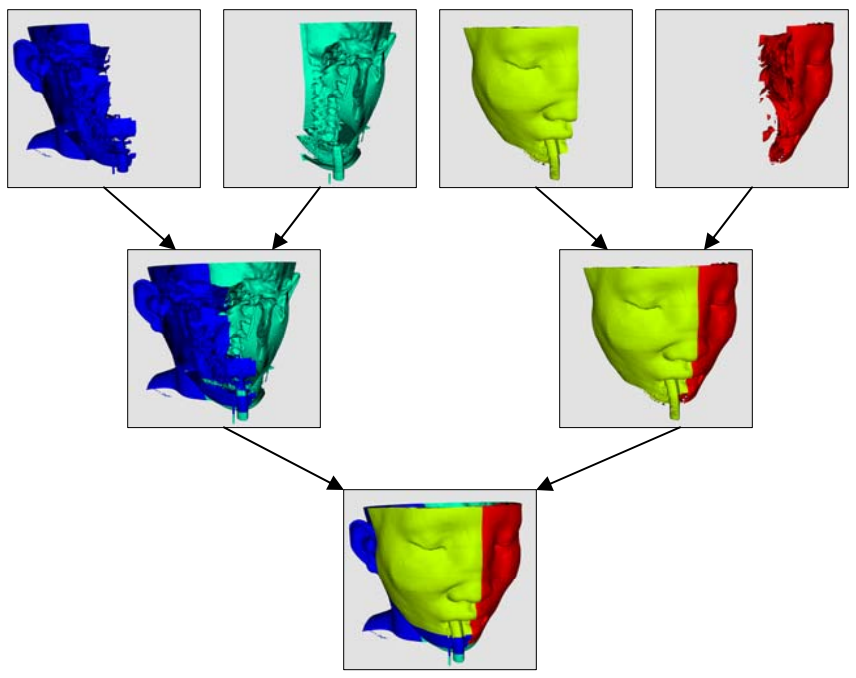




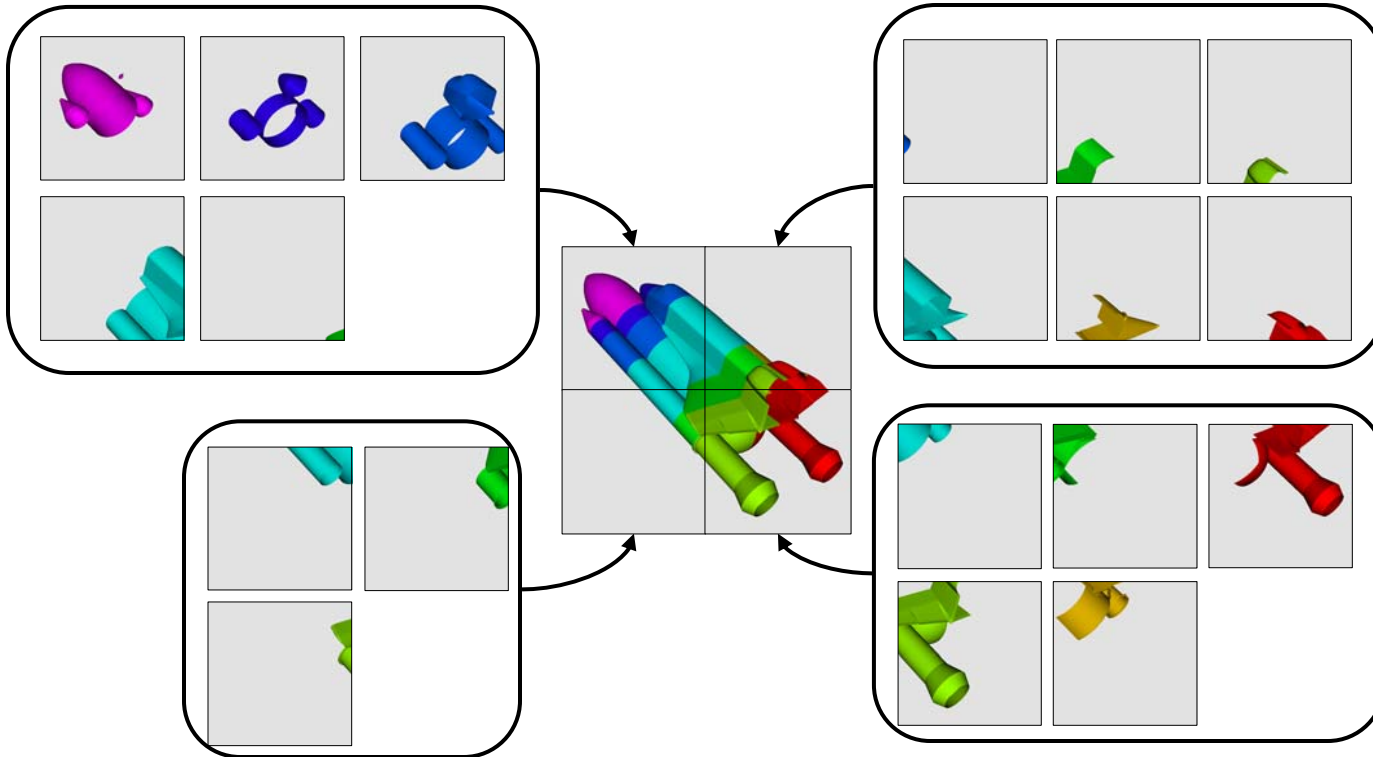
# Parallel Rendering



# Parallel Rendering



# Tiled Displays





# Rendering Modes

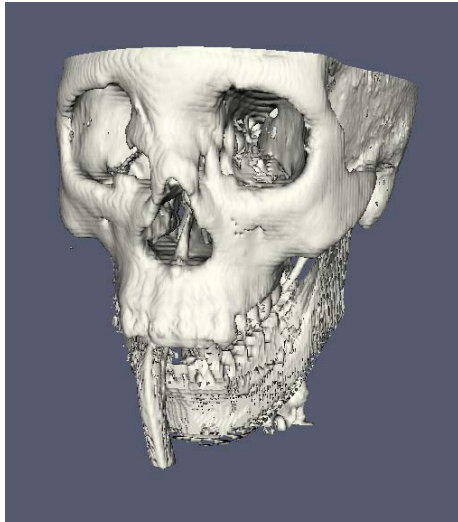
---

- Still Render
  - Full detail render.
- Interactive Render
  - Sacrifices detail for speed.
  - Provides quick rendering rate.
  - Used when interacting with 3D view.

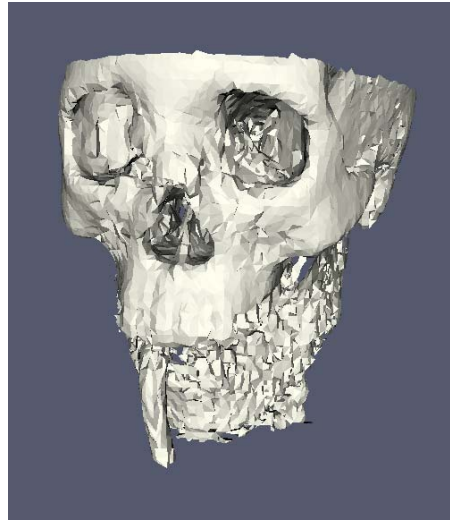
# Level of Detail (LOD)

---

- Geometric decimation.
- Used only with Interactive Render



Original Data



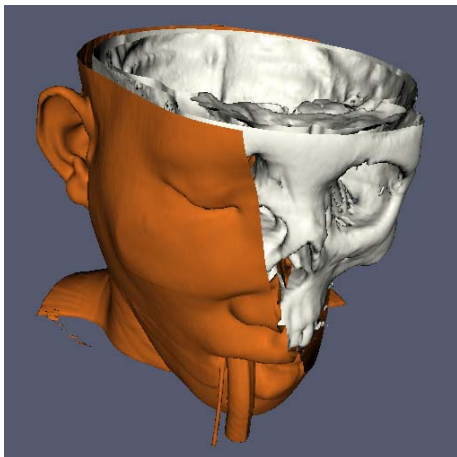
Divisions: 50x50x50



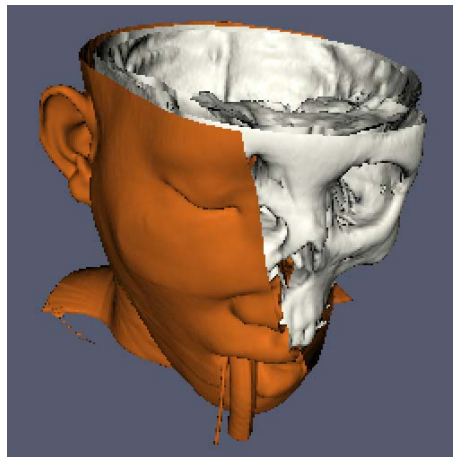
Divisions: 10x10x10

# Image Size LOD

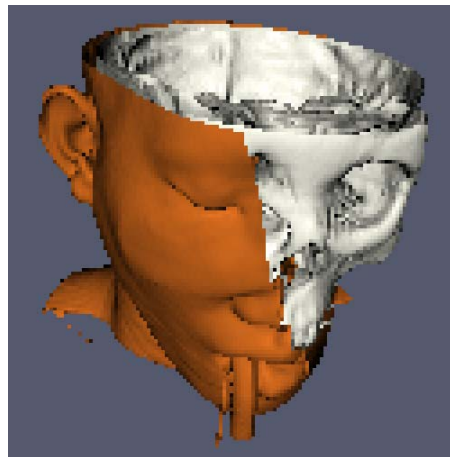
- ParaView's parallel rendering overhead proportional to image size.
- To speed up interactive rendering, ParaView can render smaller sized images and inflate them.



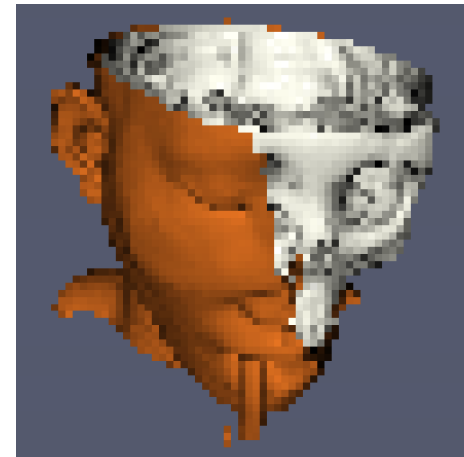
Original Data



Subsample Rate: 2 pixels



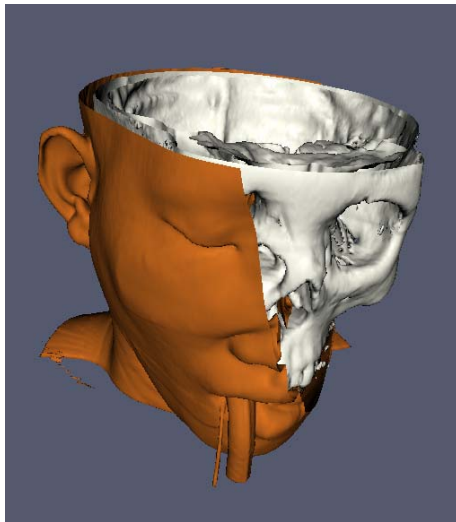
Subsample Rate: 4 pixels



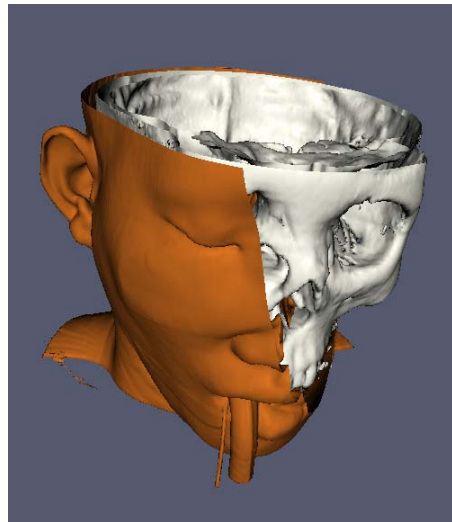
Subsample Rate: 8 pixels

# Color Depth LOD

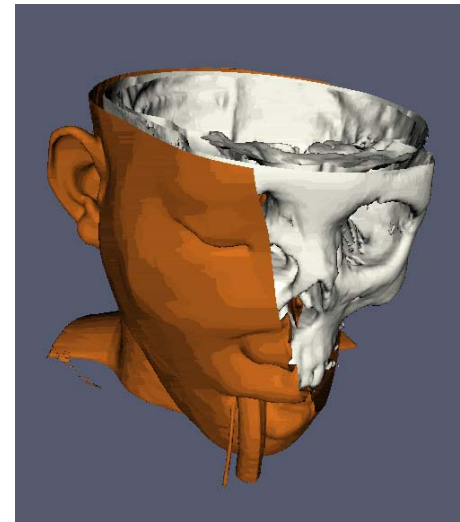
- Squirt is used to transfer images from server to client.
- Squirt is a run length encoder optimized for images.
- Run lengths improved by masking out some color bits.



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

---

- Amy Henderson Squillacote. *The ParaView Guide*. Kitware, Inc., 2006.
- <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

---

- Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit*. Kitware, Inc., fourth edition, 2006.
- Kitware Inc. *The VTK User's Guide*. Kitware, Inc., 2006.
- Jasmin Blanchette and Mark Summerfield. *C++ GUI Programming with Qt 4*. Prentice Hall, 2006.



# Further Reading

## Parallel VTK Topics

---

- James Ahrens, Charles Law, Will Schroeder, Ken Martin, and Michael Papka. “A Parallel Approach for Efficiently Visualizing Extremely Large, Time-Varying Datasets.” Technical Report #LAUR-00-1620, Los Alamos National Laboratory, 2000.
- James Ahrens, Kristi Brislawn, Ken Martin, Berk Geveci, C. Charles Law, and Michael Papka. “Large-Scale Data Visualization Using Parallel Data Streaming.” *IEEE Computer Graphics and Applications*, 21(4): 34–41, July/August 2001.
- Andy Cedilnik, Berk Geveci, Kenneth Moreland, James Ahrens, and Jean Farve. “Remote Large Data Visualization in the ParaView Framework.” *Eurographics Parallel Graphics and Visualization 2006*, pg. 163–170, May 2006.





# Further Reading

## Advanced Pipeline Execution

---

- James P. Ahrens, Nehal Desai, Patrick S. McCormic, Ken Martin, and Jonathan Woodring. “A Modular, Extensible Visualization System Architecture for Culled, Prioritized Data Streaming.” *Visualization and Data Analysis 2007, Proceedings of SPIE-IS&T Electronic Imaging*, pg 64950I-1–12, January 2007.
- John Biddiscombe, Berk Geveci, Ken Martin, Kenneth Moreland, and David Thompson. “Time Dependent Processing in a Parallel Pipeline Architecture.” *IEEE Visualization 2007*. October 2007.



# Further Reading

## Parallel Rendering

---

- Kenneth Moreland, Brian Wylie, and Constantine Pavlakos. “Sort-Last Parallel Rendering for Viewing Extremely Large Data Sets on Tile Displays.” *Proceedings of IEEE 2001 Symposium on Parallel and Large-Data Visualization and Graphics*, pg. 85–92, October 2001.
- Kenneth Moreland and David Thompson. “From Cluster to Wall with VTK.” *Proceedings of IEEE 2003 Symposium on Parallel and Large-Data Visualization and Graphics*, pg. 25–31, October 2003.
- Kenneth Moreland, Lisa Avila, and Lee Ann Fisk. “Parallel Unstructured Volume Rendering in ParaView.” *Visualization and Data Analysis 2007, Proceedings of SPIE-IS&T Electronic Imaging*, pg. 64950F-1–12, January 2007.