Cornell University
Center for Advanced Computing

Introduction to Remote Visualization

Drew Dolgert

Thanks to:
Kelly Gaither – TACC
Drew Dolgert – Cornell
Adam Kubach, Jeff Conner – ASU

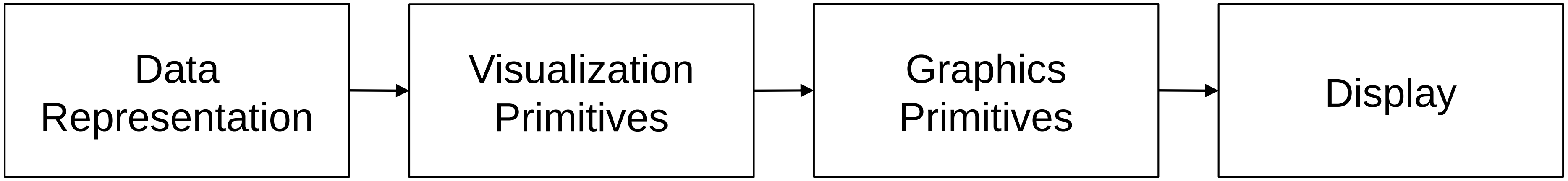
Agenda and Timeline

- 9:00 Overview of Computer Graphics and Scientific Visualization
- 9:30 Lab: Local Paraview
- 10:15 Break
- 10:30 Remote Visualization
- 10:45 Lab: Remote Paraview
- 11:15 Parallel Visualization
- 11:30 Lab: Parallel Paraview

Scientific Visualization

- 3 Goals
  - Insight
  - Validation
  - Communication

Getting from Data to Insight

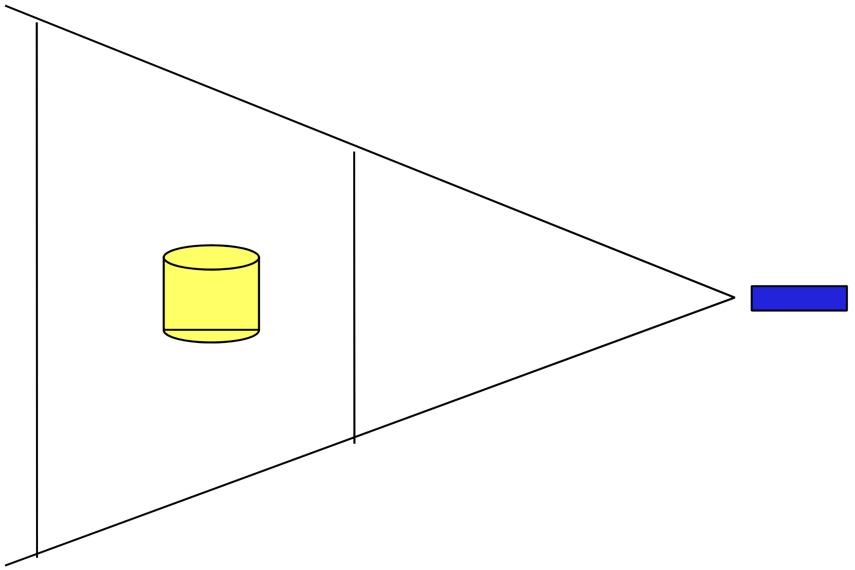


Computer Graphics Pipeline

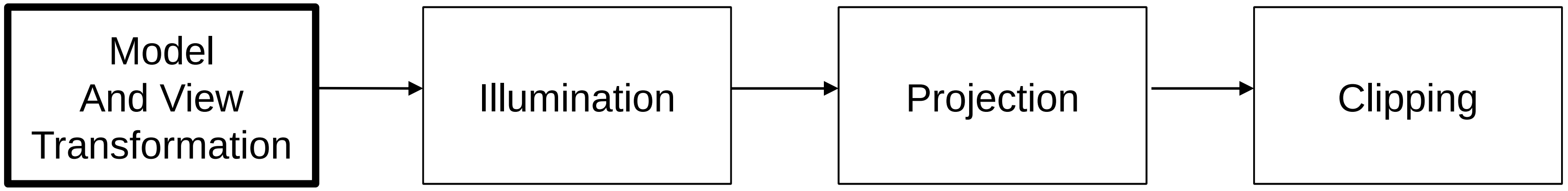
$$\begin{tikzcd}\boxed{\text{Application}} \arrow[r] & \boxed{\text{Geometry}} \arrow[r] & \boxed{\text{Rasterization}} \arrow[r] & \boxed{\text{Display}}\end{tikzcd}$$

•Reads data and maps to geometric primitives.

•May reduce number of primitives or do view frustum culling to reduce data sent to next stage.



Inside Geometry Stage

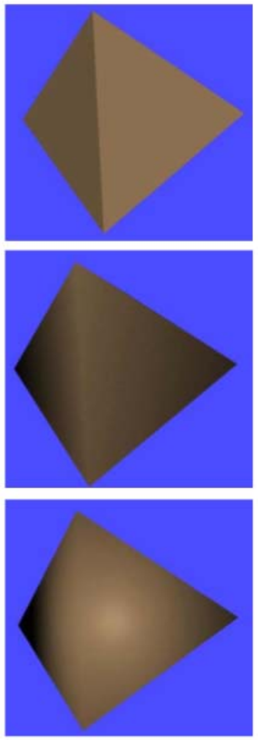


- Model transformation
  - Starts with data described in (x,y,z) coordinates. Data is transformed from it’s local space to world space.
- View transformation
  - Change coordinate system such that the eye is at 0,0,0 and looking in the direction of the negative z-axis. This is called eye space.

Inside Geometry Stage

$$\begin{tikzcd}
\boxed{\text{Model And View Transformation}} \arrow[r] & \boxed{\textbf{Illumination}} \arrow[r] & \boxed{\text{Projection}} \arrow[r] & \boxed{\text{Clipping}}
\end{tikzcd}$$

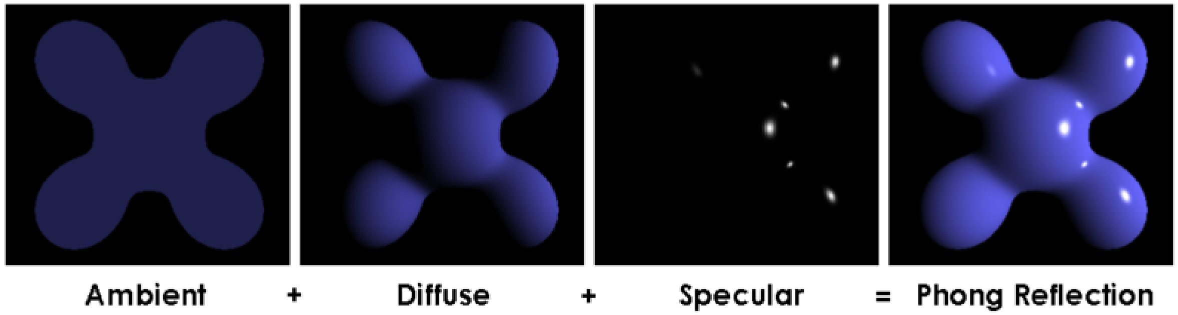
- Approximate how light interacts with objects
- Flat Shading
    - Compute light interaction per polygon
    - Whole polygon has the same color
- Gouraud shading
    - Compute light interaction per vertex
    - Interpolate the colors
- Phong shading
    - Interpolate the normals per pixel



Inside Geometry Stage

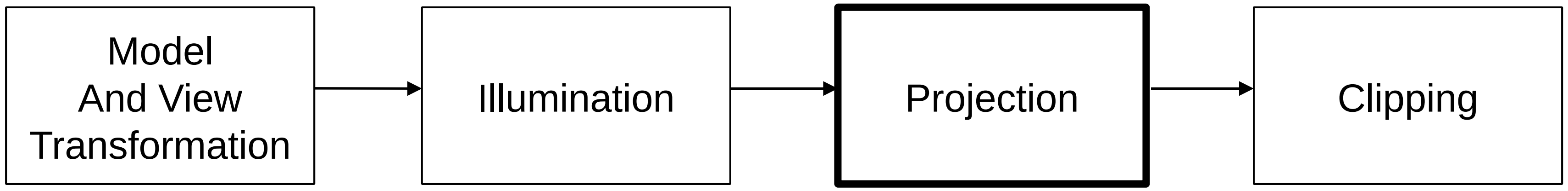
$$\begin{tikzcd}
\boxed{\text{Model And View Transformation}} \arrow[r] & \boxed{\text{Illumination}} \arrow[r] & \boxed{\text{Projection}} \arrow[r] & \boxed{\text{Clipping}}
\end{tikzcd}$$

- Ambient
  - Indirect light – for example, light that reflects off walls
- Diffuse
  - Light scattered in multiple directions (view-independent)
- Specular
  - Makes surfaces shiny by creating highlights (view-dependent)



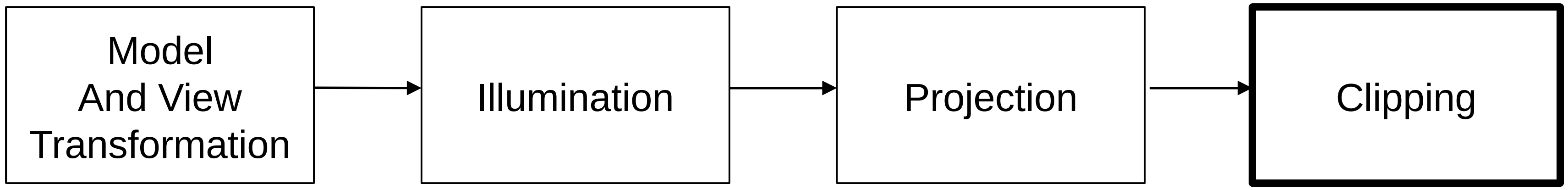
Source
www.wikipedia.org

Inside Geometry Stage



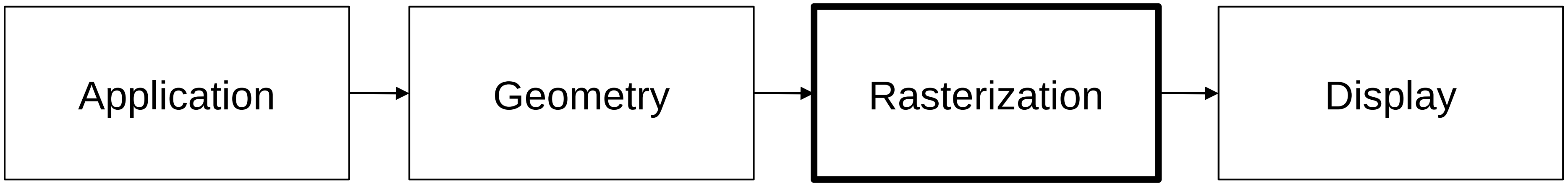
- Transforms the viewing volume into cube with extreme points at (-1,-1,-1) and (1,1,1). Projection is usually either:
  - Orthographic
    - » Parallel lines remain parallel
  - Perspective
    - » Objects that are farther away appear smaller. This mimics how we perceive objects.

Inside Geometry Stage



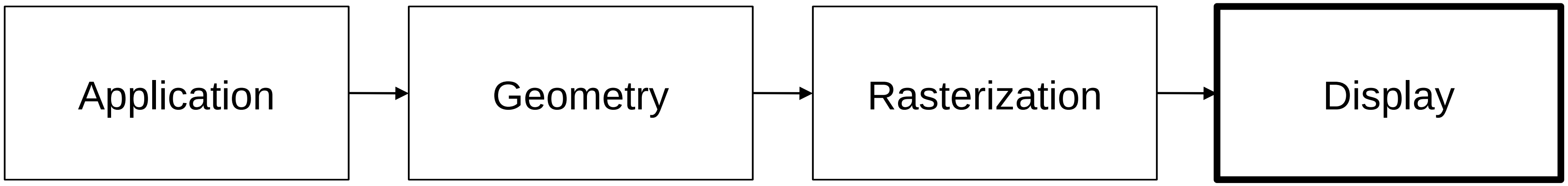
- Any values outside the unit cube are not sent to the rasterizer stage.
- Values that are sent to rasterizer are in window coordinates.

Computer Graphics Pipeline

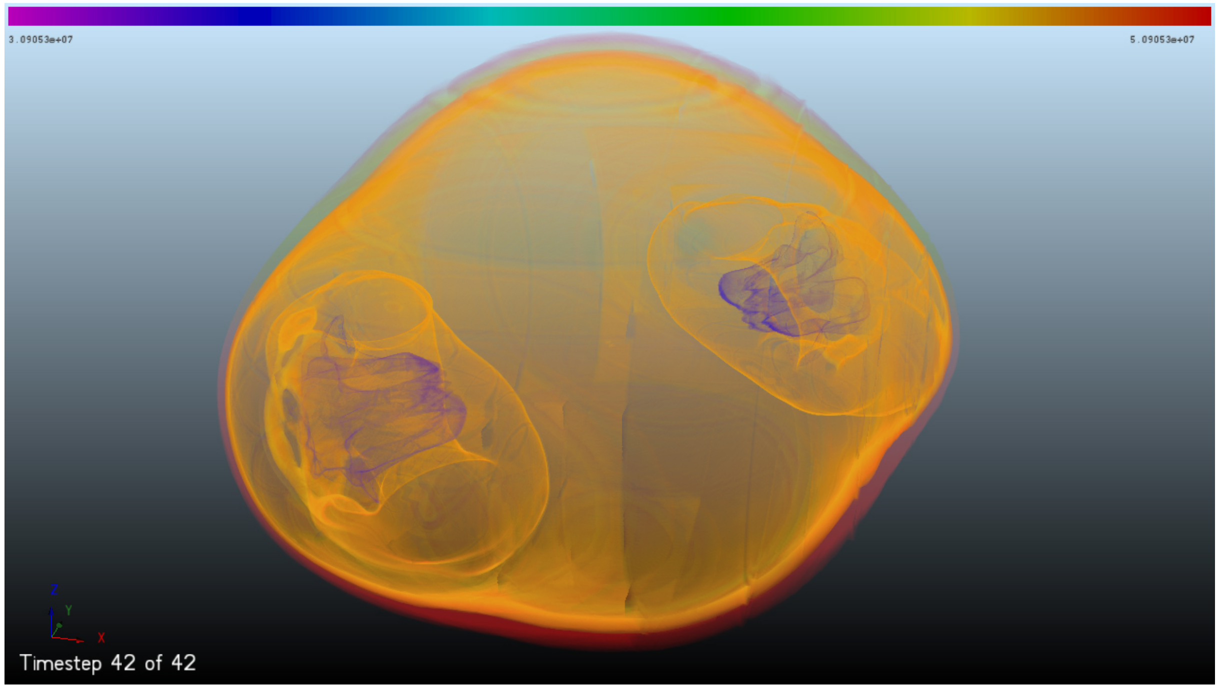


- Convert objects into fragments (future pixels).
- Stage also resolves visibility
  - Z values of pixels are compared before being written to display (depth test)
  - Errors in depth testing can occur due to a lack of precision in the depth buffer (Z-fighting)
    - Changing near and far planes of viewing volume can help

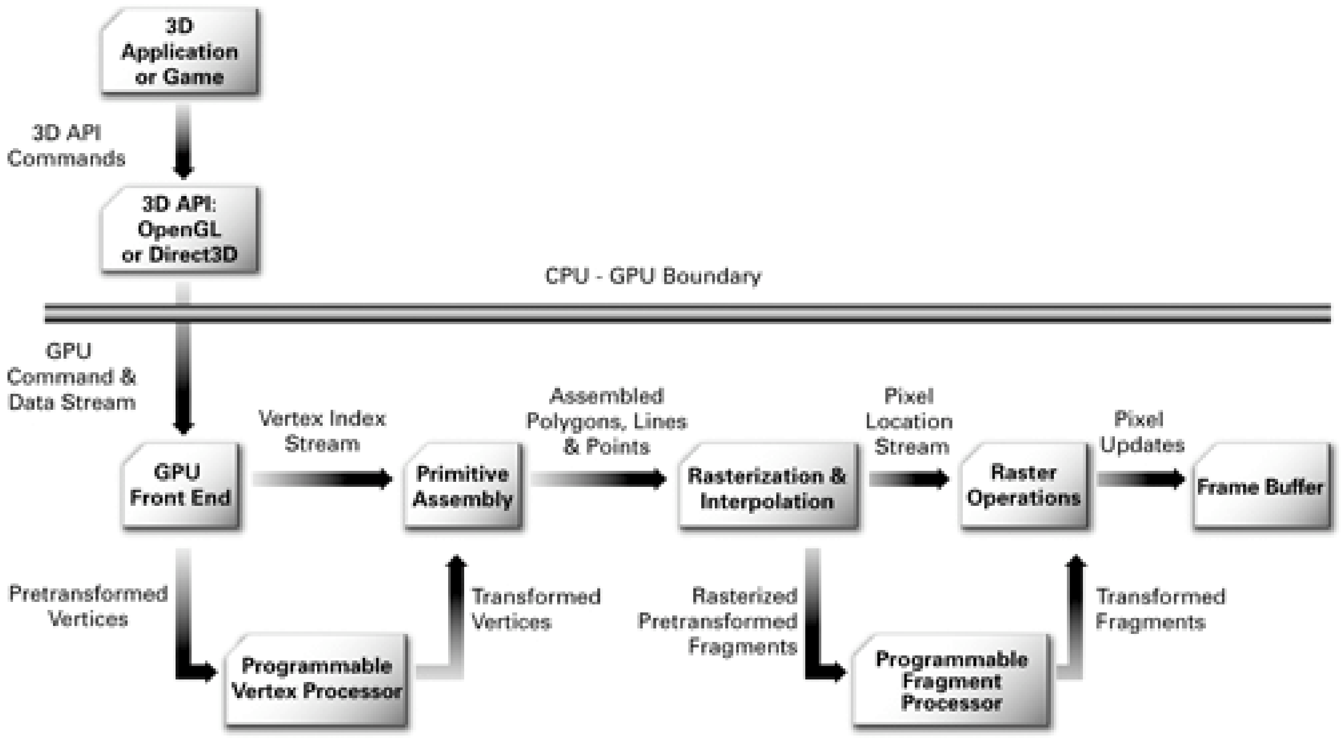
Computer Graphics Pipeline



- Pixels are the result

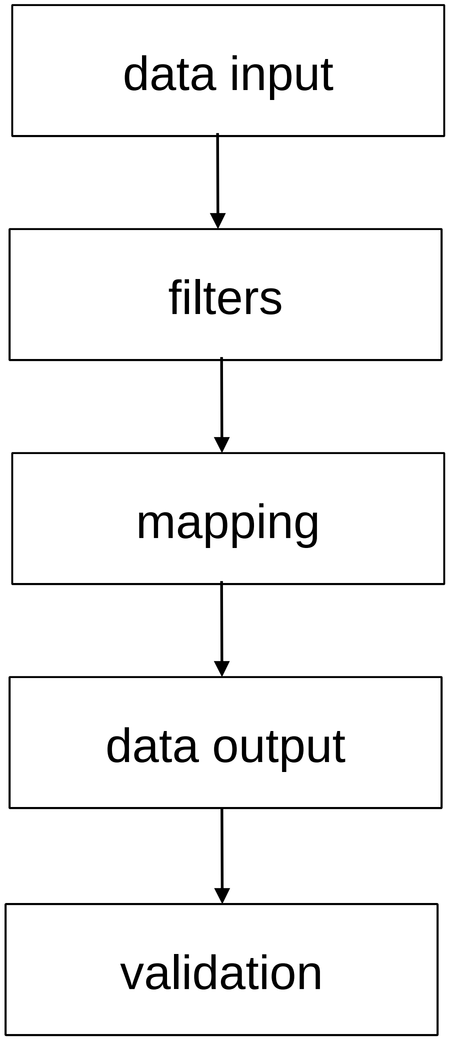


OpenGL Pipeline



Source: CG Tutorial

Scientific Visualization Process



Read data into tool

Pre-process data

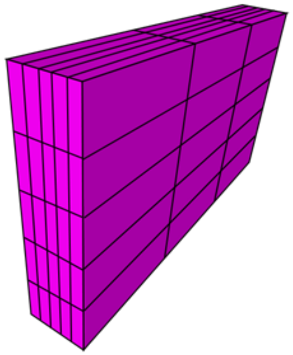
Map data to graphics primitives (points, lines, polygons)

Graphics pipeline

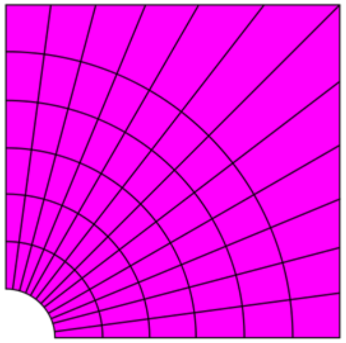
Verify that the result is a reasonable approximation of the raw data.

Grid Types of Input Data

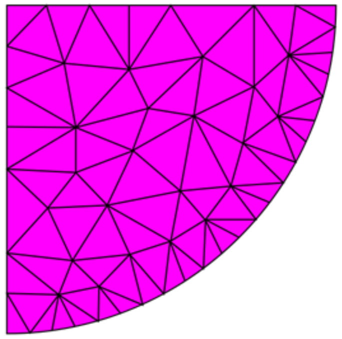
- Structured
  - Rectilinear



  - Curvilinear



- Unstructured



Images from www.wikipedia.org

Types of Values Present in Input Data

- Scalar – single value at either the node or center of the cell.
- Vector - 3-tuple typically indicating both magnitude and direction at either the node or the center of the cell.
- Tensor – system of equations that can be solved to find various physical components.

Visualization Techniques

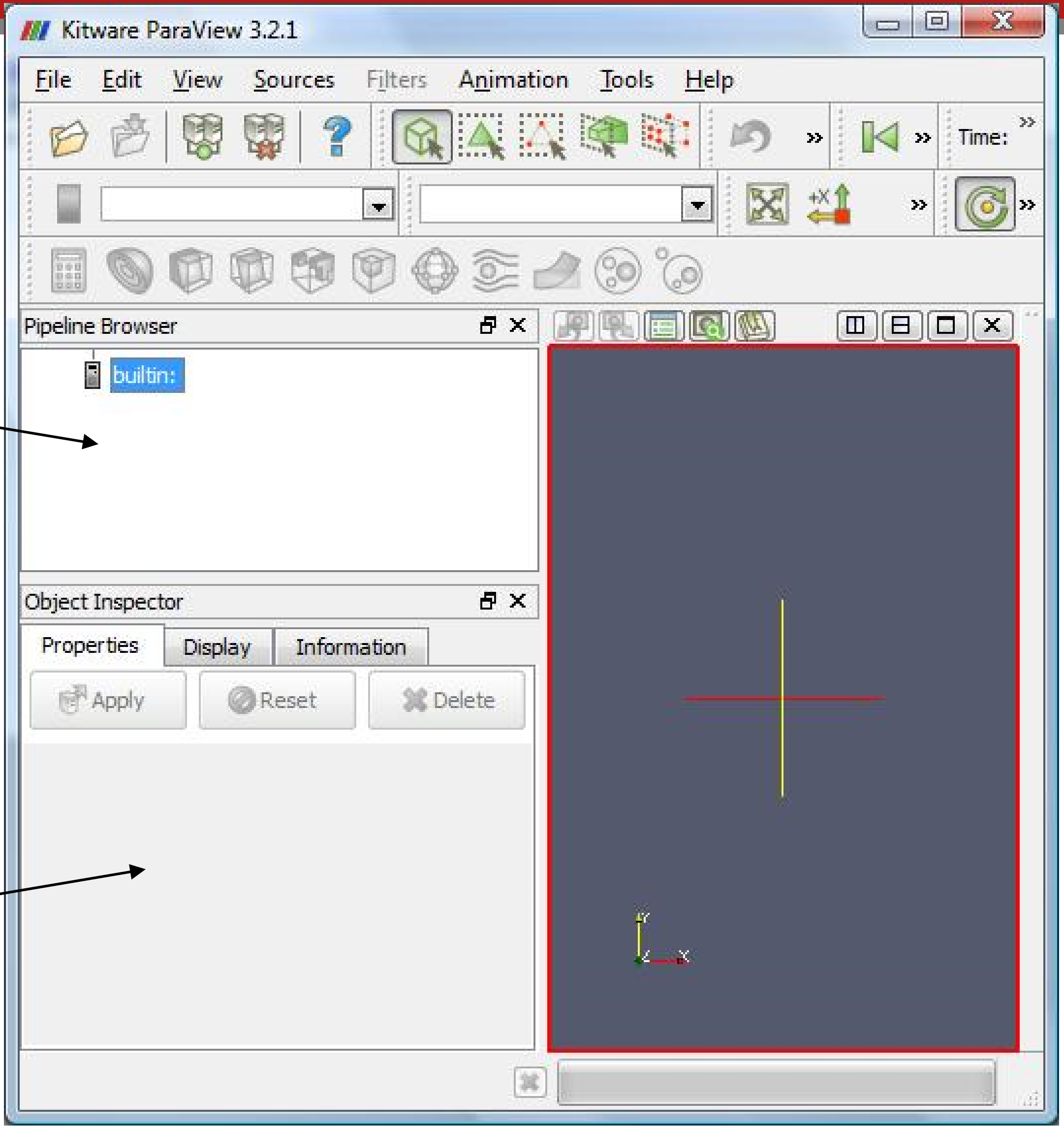
- Surface – Extracts polygonal versions of calculated components
- Volume – Calculates how light interacts with contents of grid
- Glyph – Use symbols to represent values or states within a field of information

Visualizing with Paraview

- Popular visualization tool
  - Built on VTK and Qt
  - Handles many formats
  - Multiple visualization techniques
  - Open source
  - Plugins

Using Paraview Locally

- Download:
  - http://www.paraview.org/paraview/resources/software.html
- Sample data:
  - http://www.tacc.utexas.edu/~kelly/TRAINING/RectGrid2.vtk
- Goal is to get familiar with Paraview before using remote systems.



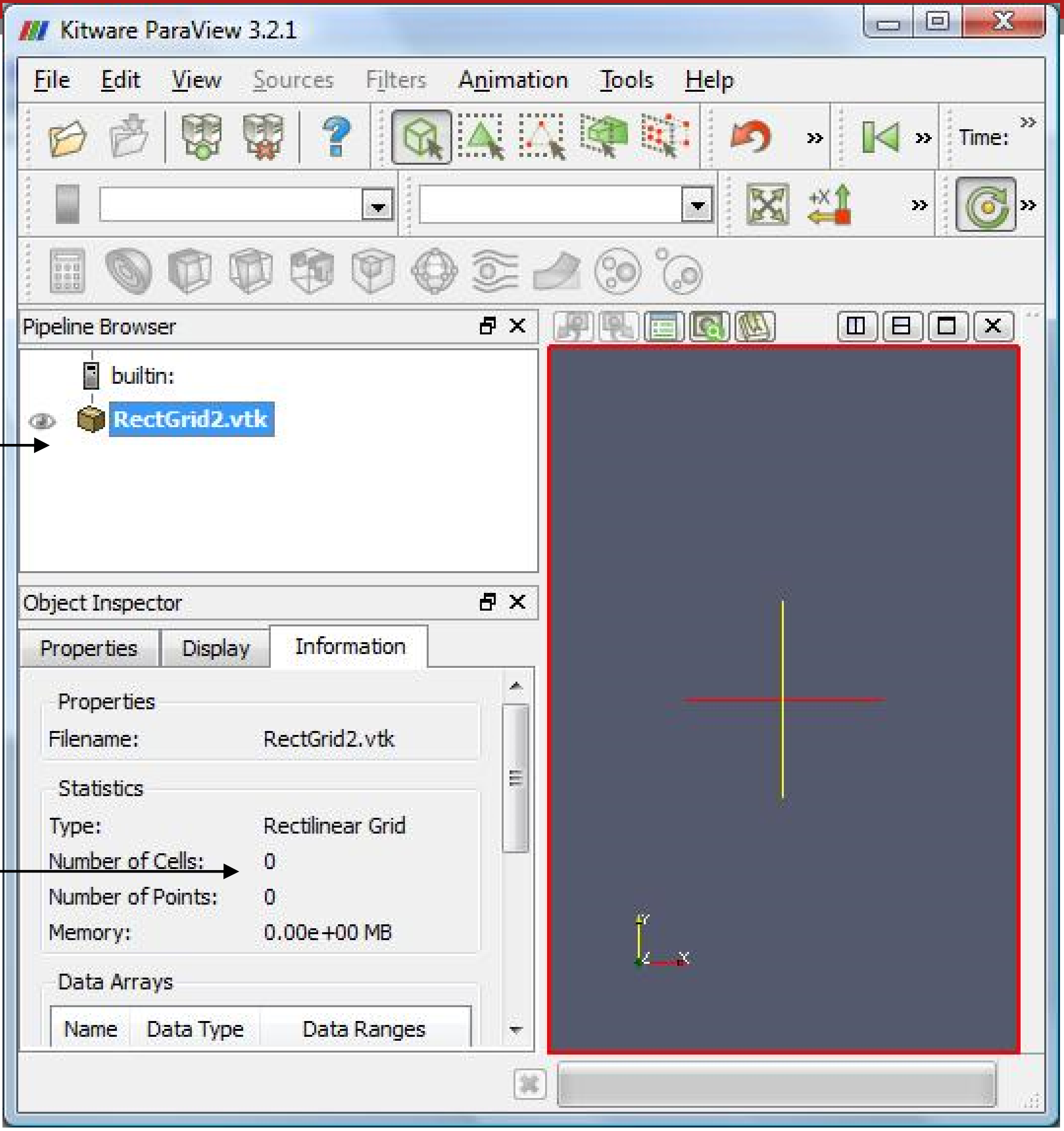
Pipeline browser

Object inspector

Load RectGrid2.vtk using File->open

Your file at the top of the pipeline.

It thinks there are no cells or points in the file.



Properties

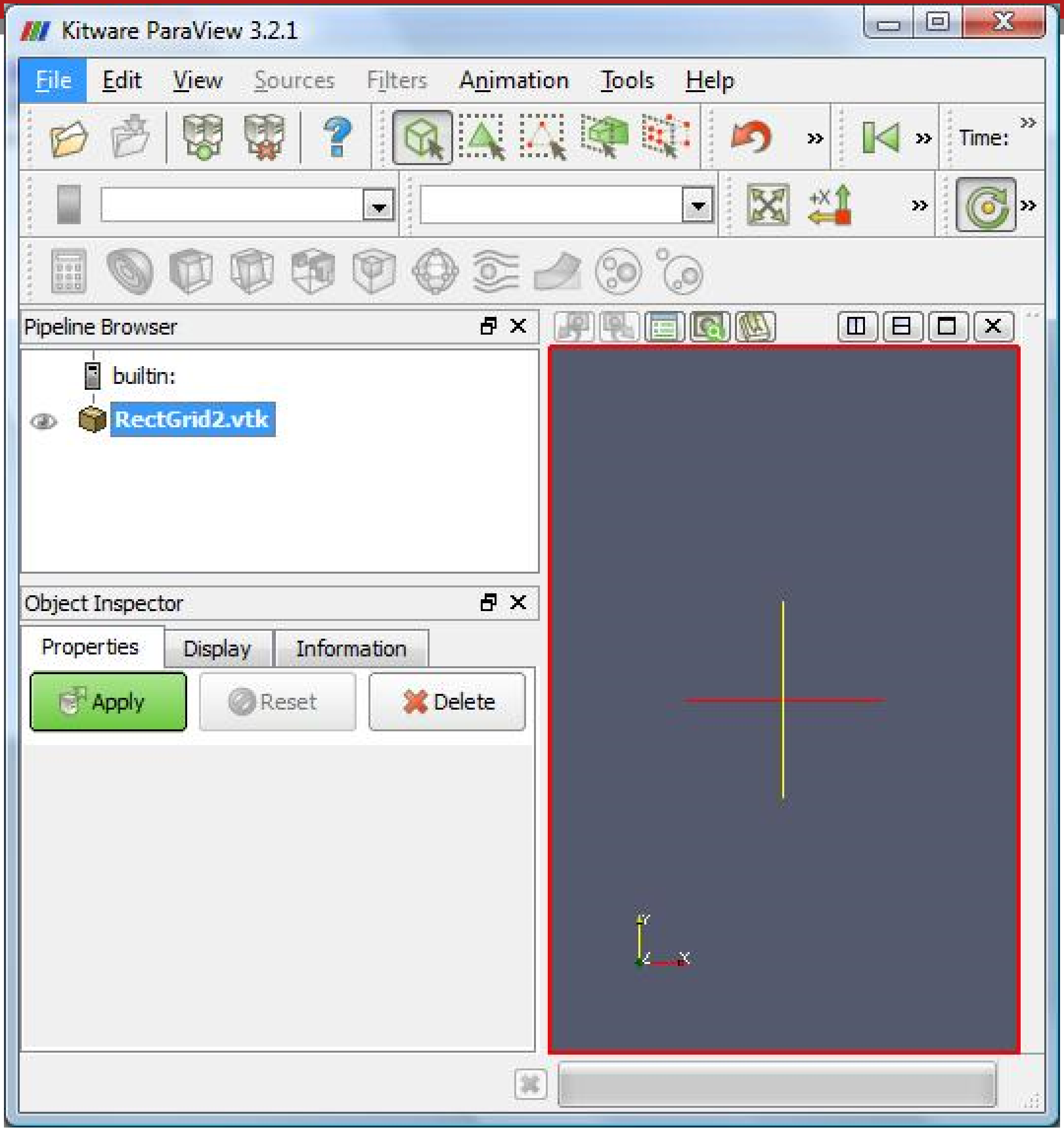
Filename: RectGrid2.vtk

Statistics

Type:	Rectilinear Grid
Number of Cells:	0
Number of Points:	0
Memory:	0.00e+00 MB

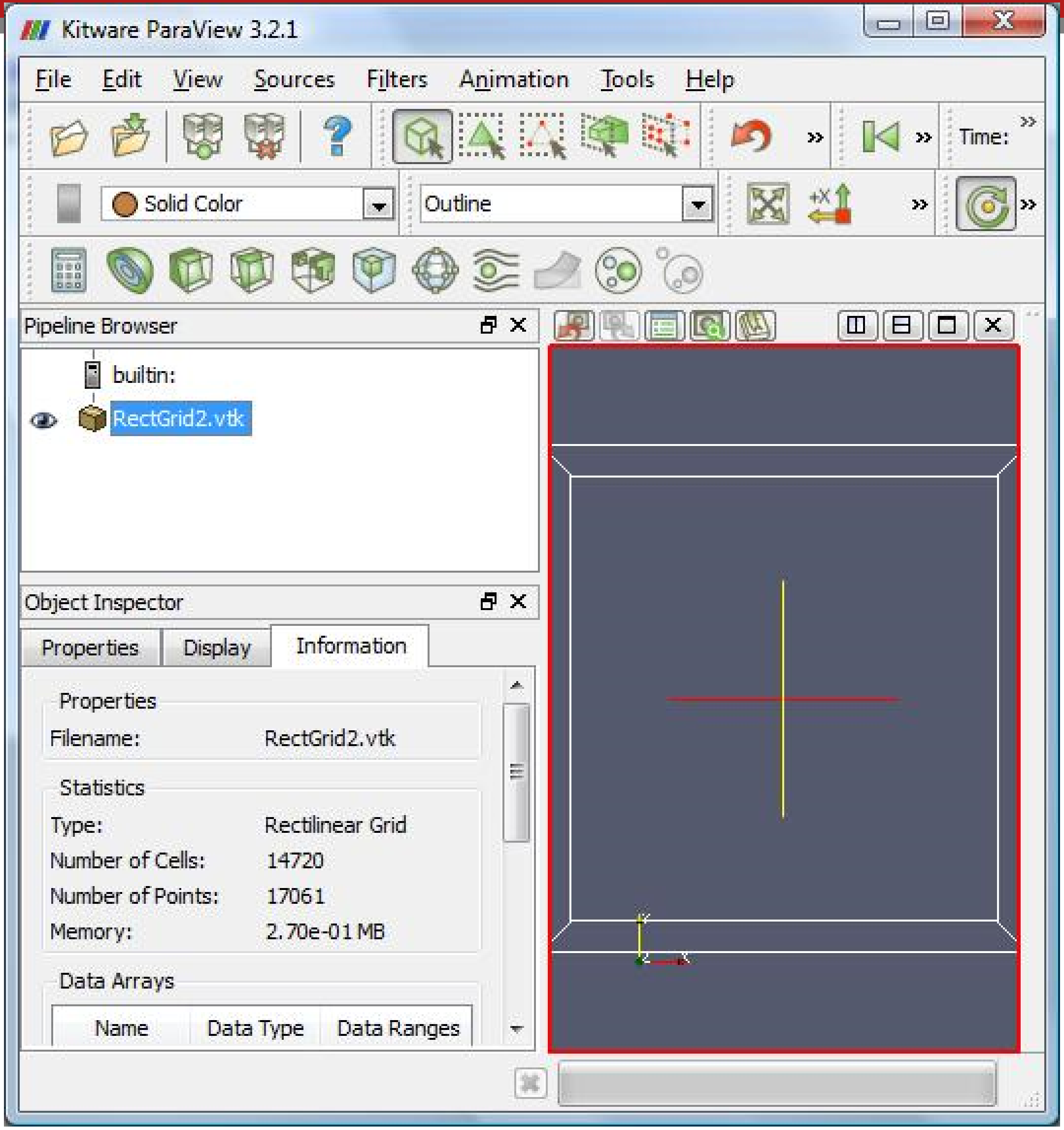
Data Arrays

Name	Data Type	Data Ranges



Hit “Apply” to load the file.

Confirm data loaded by looking at information tab.



Properties

Filename:	RectGrid2.vtk

Statistics

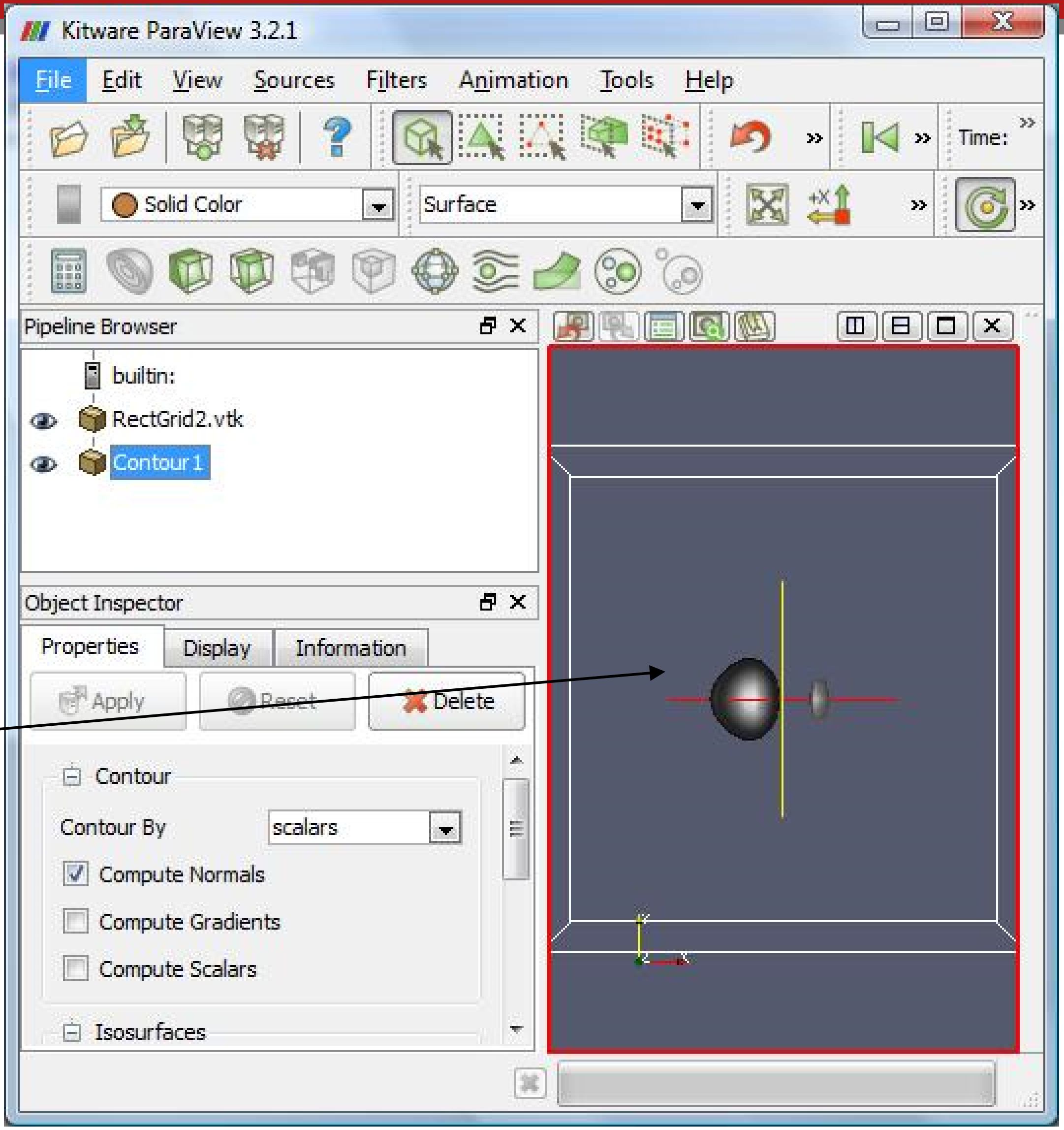
Type:	Rectilinear Grid
Number of Cells:	14720
Number of Points:	17061
Memory:	2.70e-01 MB

Data Arrays

Name	Data Type	Data Ranges

1. Select dataset
2. Find Contour filter in the Filters menu.
3. Hit Apply, as usual.

Click and drag. Try ctrl, shift, middle-click, right-click.

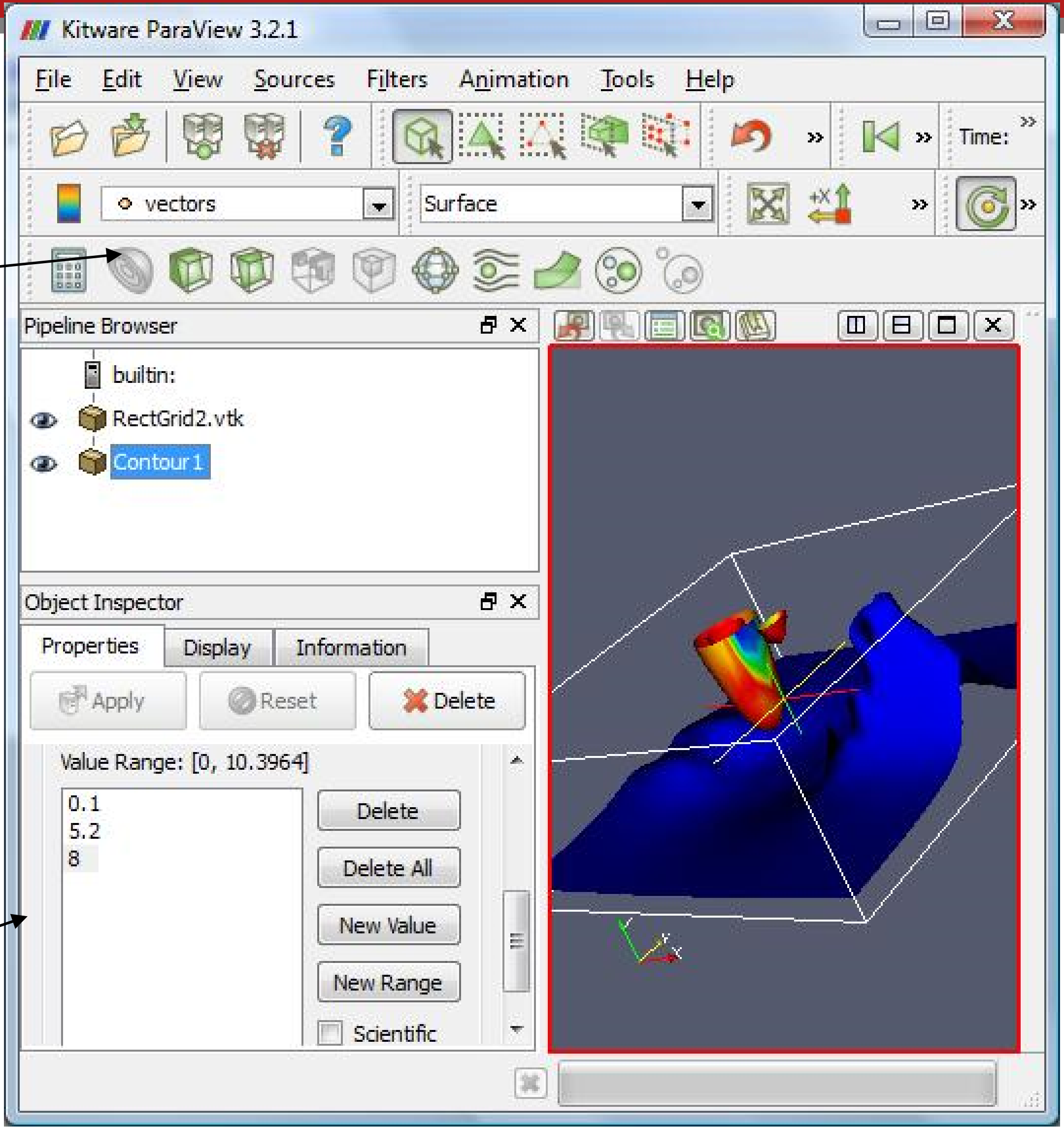


Contour

Contour By: scalars

- [✓] Compute Normals
- [ ] Compute Gradients
- [ ] Compute Scalars

Isosurfaces



Value Range: [0, 10.3964]

```
0.1
5.2
8
```

Add colors with the drop-down list box. The choices come from fields of the dataset.

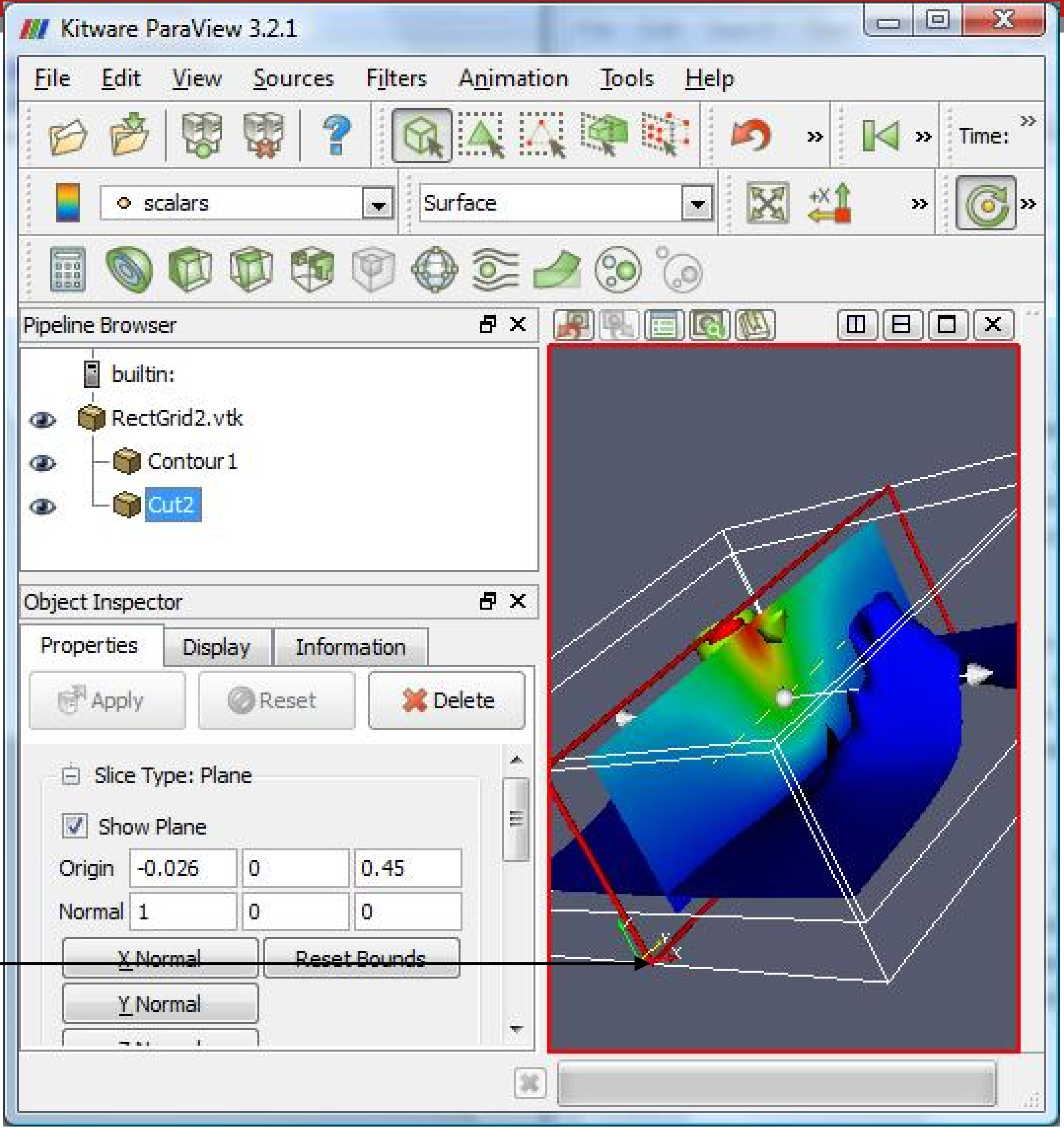
Change contours

1. Select RectGrid2.vtk
2. Add Slice filter.
3. Hit apply, again.

or...
1. Select Contour1
2. Add Slice filter.
3. Apply, apply.

What is the difference?

Click and drag. Try Ctrl, Shift, right-click.

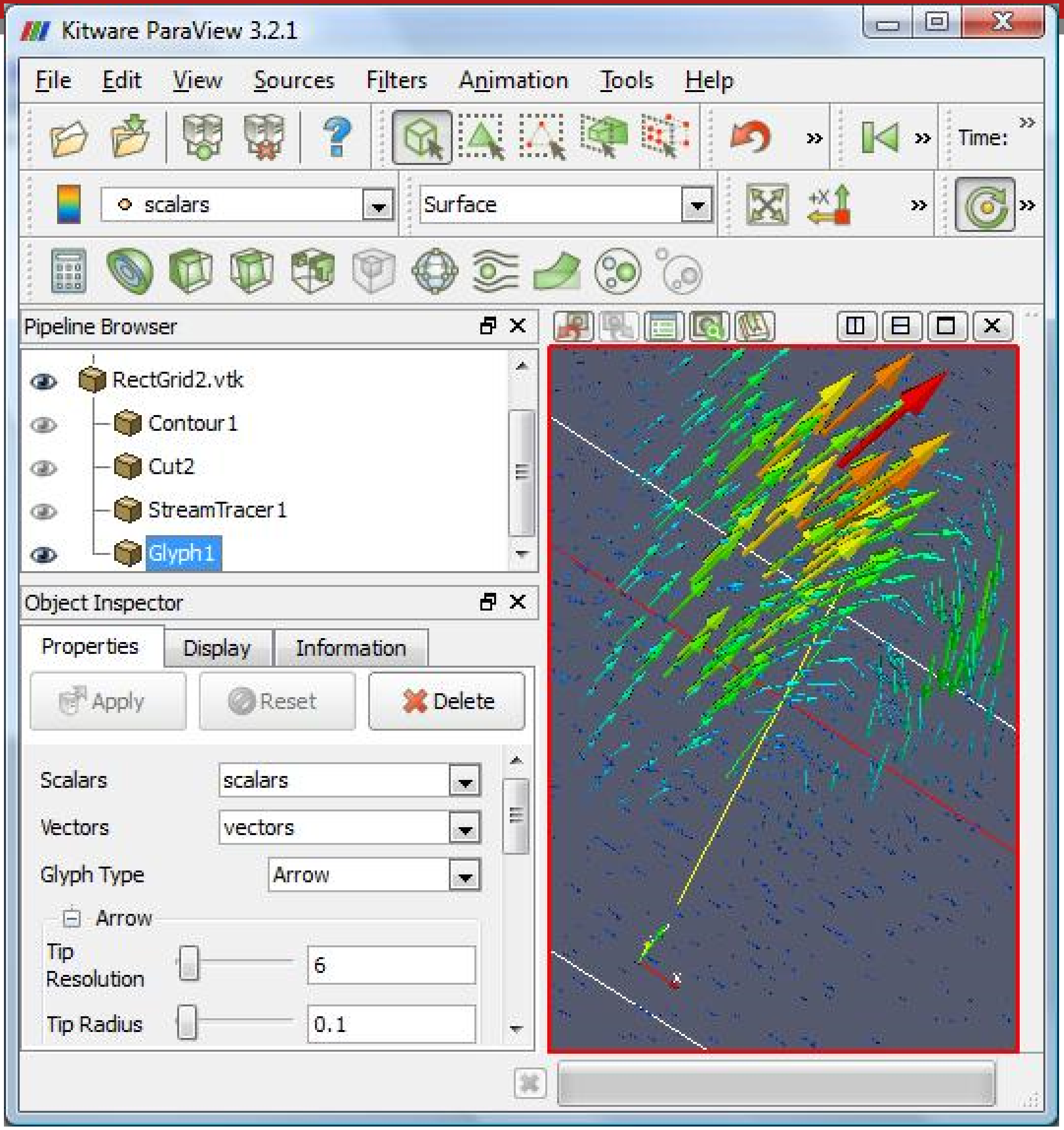


Slice Type: Plane

Show Plane: ✓

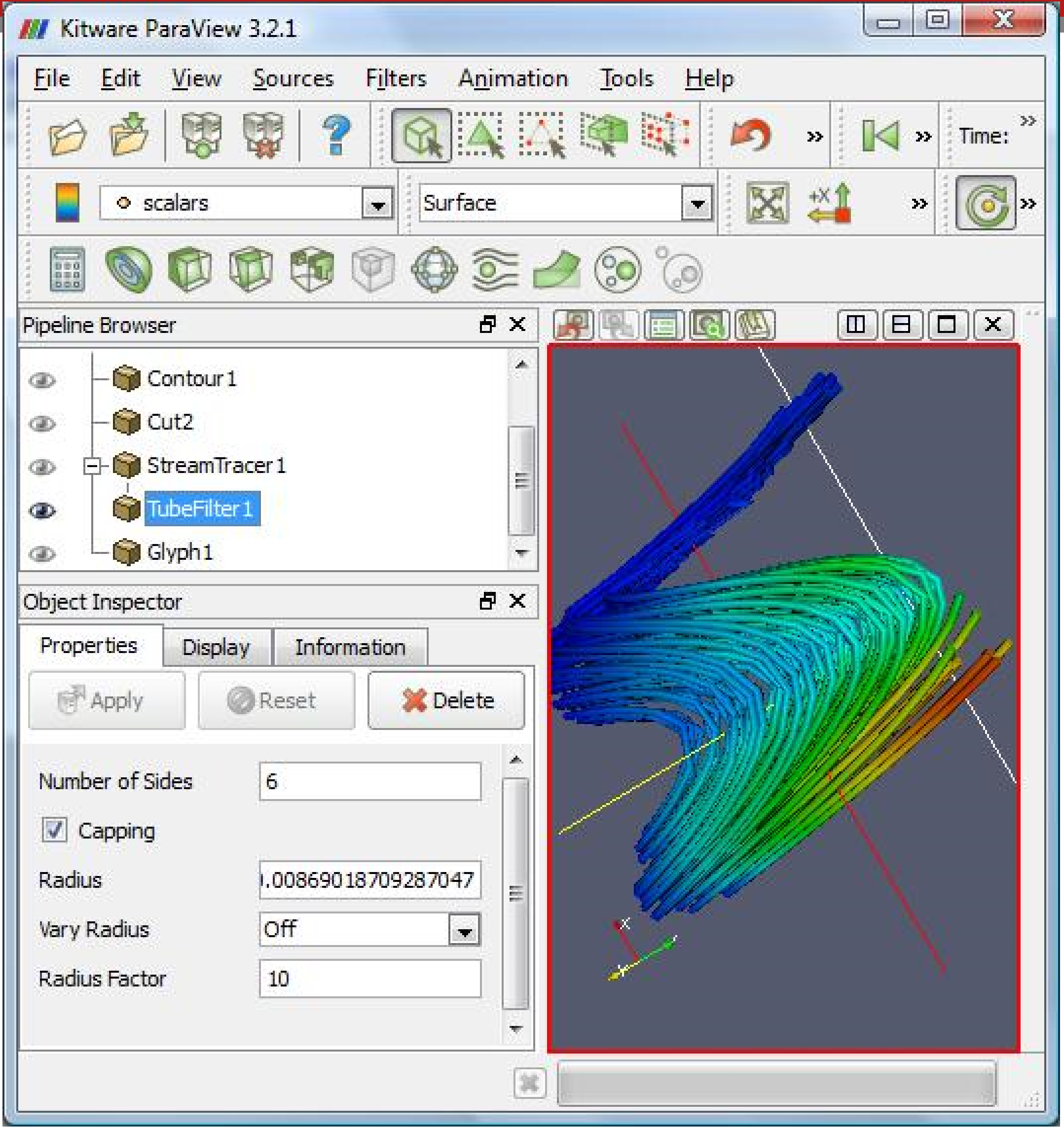
Origin	-0.026	0	0.45
Normal	1	0	0

Glyph filter.
Play with the glyph type options.



Scalars: scalars
Vectors: vectors
Glyph Type: Arrow
Arrow
Tip Resolution: 6
Tip Radius: 0.1

This time, add the tube filter to the StreamTracer, not to RectGrid2.

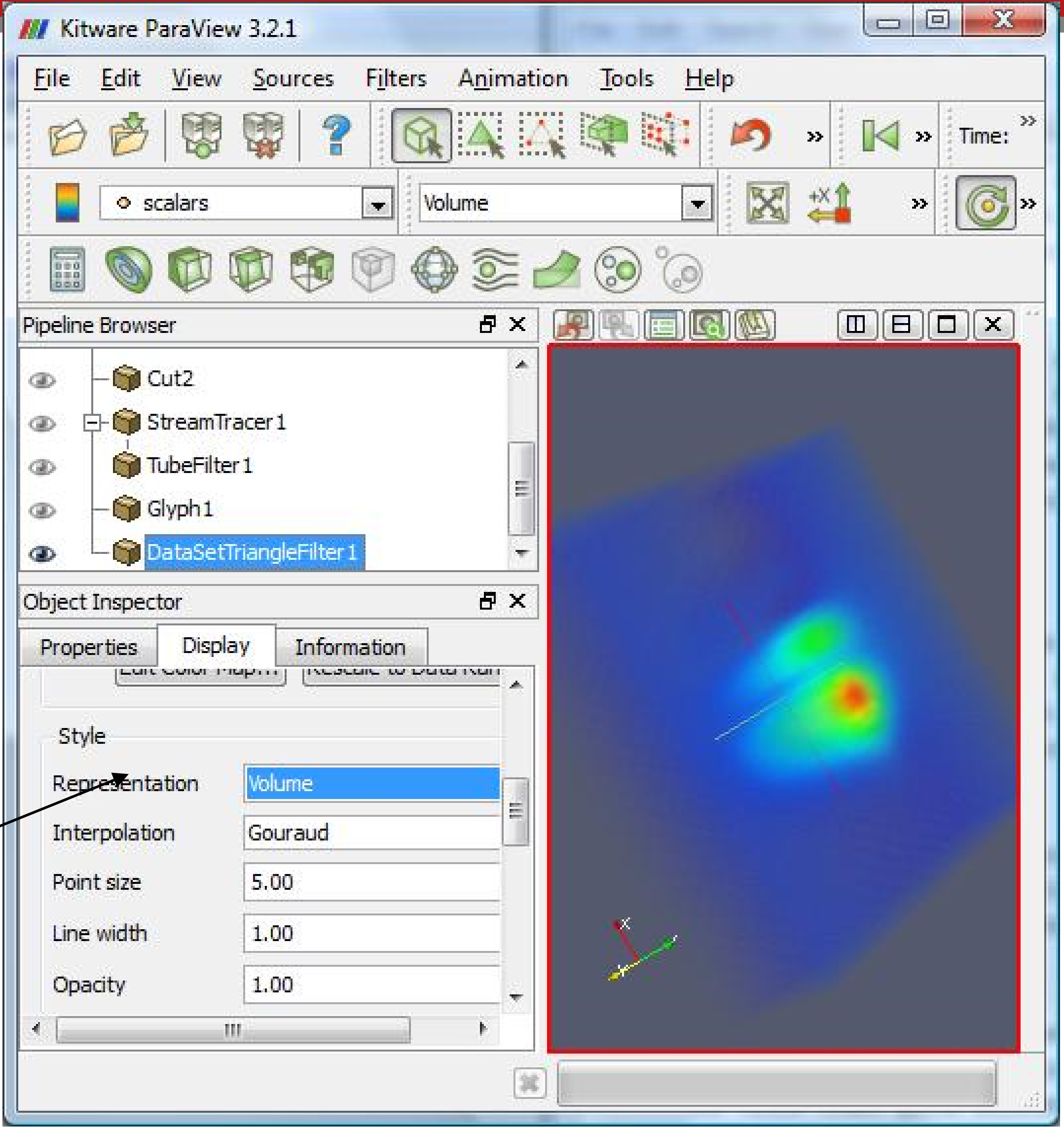


Number of Sides 6
Capping ✓
Radius).00869018709287047
Vary Radius Off
Radius Factor 10

Volume Rendering

First, add "tetrahedalize" filter.
Select Display tab.
Find the Style section.
Change representation.

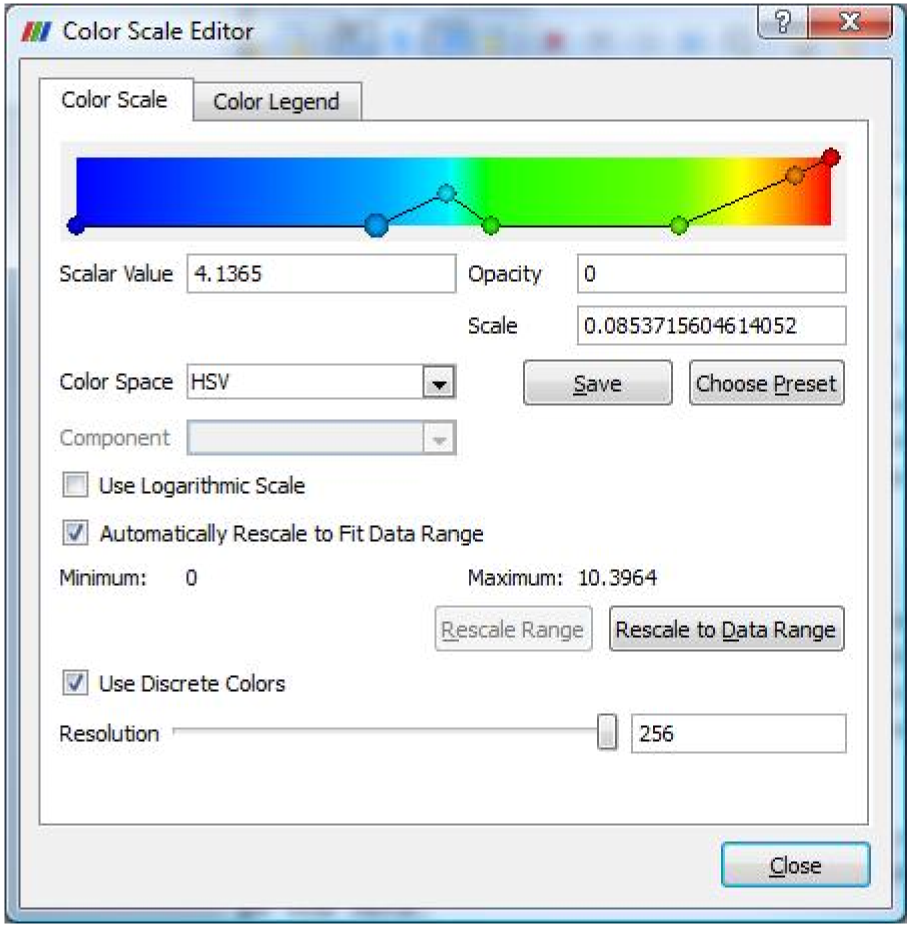
The colormap is just above
the Style section if you scroll up.



Style

Representation	Volume
Interpolation	Gouraud
Point size	5.00
Line width	1.00
Opacity	1.00

For volume rendering, the y axis of the line determines opacity.


Color Scale Editor

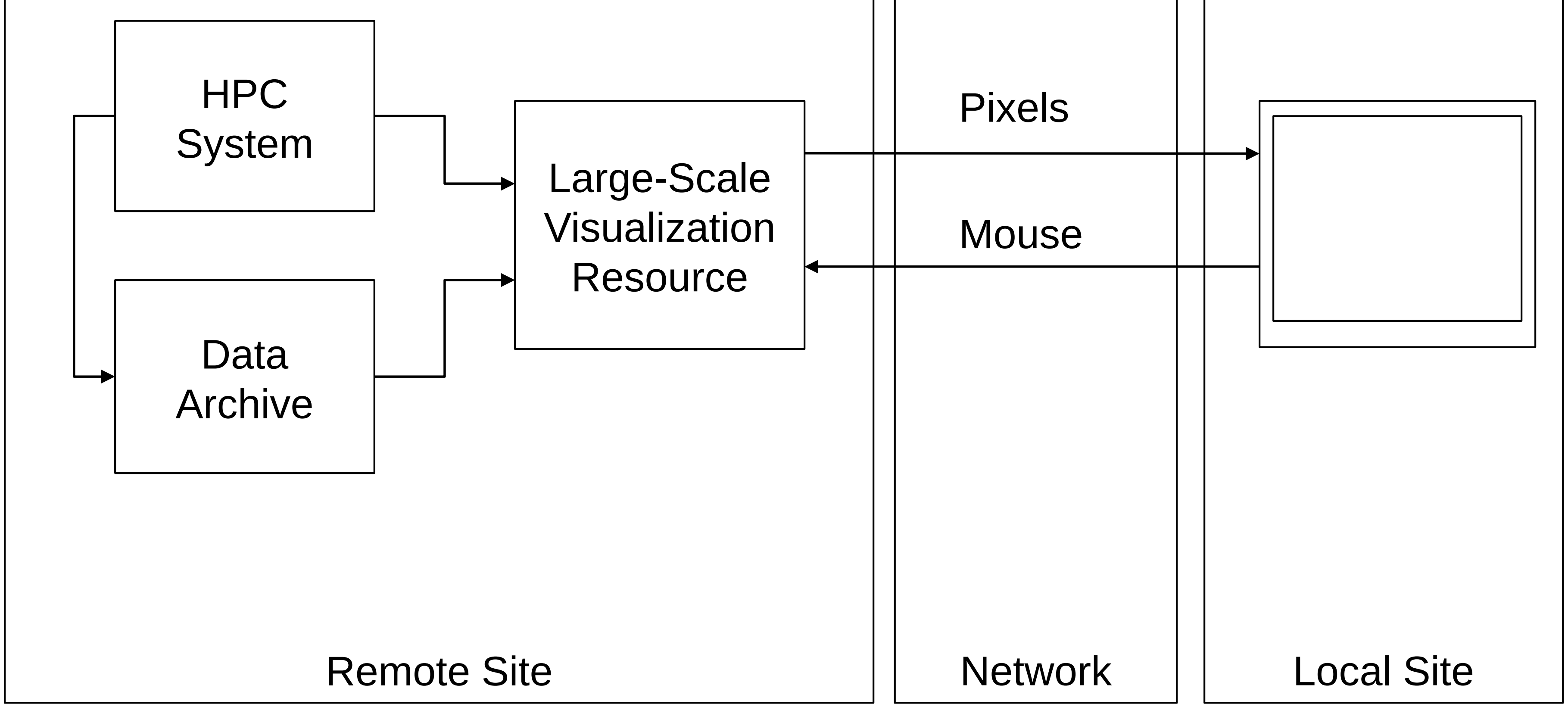
Color Scale | Color Legend

Scalar Value: 4.1365
Opacity: 0
Scale: 0.0853715604614052
Color Space: HSV
Component:
Use Logarithmic Scale [ ]
Automatically Rescale to Fit Data Range [✓]
Minimum: 0
Maximum: 10.3964
Use Discrete Colors [✓]
Resolution: 256

Remote Visualization

- Motivation:
  - Data too big to move
  - Local workstation not powerful enough to display
- Challenges:
  - Bandwidth
    - 1280 x 1024 pixels of 24 bits at 12 times a second = 360 MBps
  - Latency of network and GPU

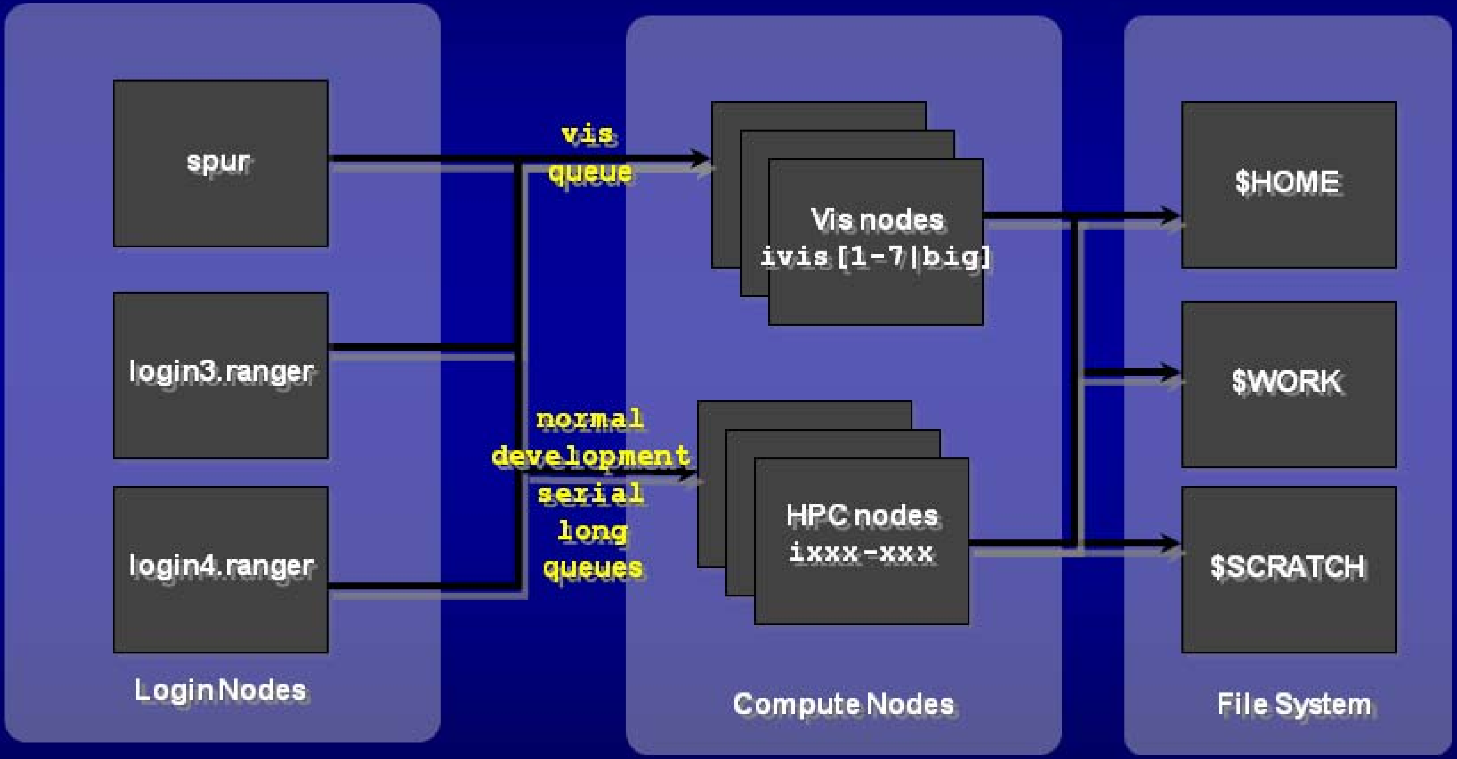
Remote Visualization Model



TACC TeraGrid Visualization Hardware

- Spur
  - 8 node visualization cluster (+ login node)
  - Each node contains 16 cores, at least 128 GB of RAM and 4 NVIDIA Quadro GX 5600 GPUs
    - Total: 128 Cores, 1 TB memory 32 GPUs

Spur / Ranger topology



TACC Visualization Software

- Applications
  - Parallel Paraview
  - Parallel Visit
  - Amira
  - IDL
- Toolkits
  - OpenGL (Mesa on Ranger)
  - The Visualization Toolkit (VTK)

Remove Paraview Lab

- VNC client
  - TurboVNC: http://www.virtualgl.org/Downloads/TurboVNC
- Login with putty, a secure-shell client, or ssh
  - Putty:
    - http://www.chiark.greenend.org.uk/~sgtatham/putty/
- Account is train1xx. It will work for a week.
- Copy files with “cp ~train100/* .”

Remote Paraview

- ssh username@spur.tacc.utexas.edu
- Login greeting on spur:

```
--------------------- Project balances for user train400 -----------------------
| Name              Avail SUs     Expires |                                     |
| 20091012DATA           5000             |                                     |
------------------------ Disk quotas for user train400 -------------------------
| Disk             Usage (GB)     Limit     %Used   File Usage       Limit   %Used |
| /share                  0.0         6      0.02           36      100000    0.04 |
-------------------------------------------------------------------------------
```

Your account identifier, used with the -A option on job scripts.

Special for Tutorial: Share VNC

- Four people share one 128MB 16-way server.
- One person, the **lead**, submits the job script, gets a node.
- Same person connects with VNC, starts 3 more VNC servers.
- Other 3 connect to those VNC servers, using the account of the first person.

- For next few slides, the lead works through with me.
- Then the rest of each team follows through connecting to the VNC servers started by the lead.

- Ready?

Remote Paraview

- # You will vnc to spur, so you need to set a password.
- # Everybody use the password “kermit” so there is no thought later.
- spur% vncpasswd
- # Submit a job to gain access to a visualization node running vncserver.
- spur% qsub job.vnc -geometry 1440x900
- # Check to see if the job ran and wrote ~/vncserver.out
- spur% showq -u
- spur% qstat
- spur% ls vncserver.out

Find the latest here:
http://services.tacc.utexas.edu/index.php/spur-user-guide

How We Edited the Job Script

- Copy it. Then add -A 20090312HPC and -pe 4way. And change runtime to go the whole day.

```
spur% cp /share/sge/default/pe_scripts/job.vnc job.vnc
spur% vi job.vnc

#$ -V                                  # Inherit the submission environment
#$ -A 20090312HPC
#$ -cwd                                # Start job in submission dir
#$ -N vncserver                        # Job name
#$ -j y                                # Combine stderr and stdout into stdout
#$ -o $HOME/$JOB_NAME.out              # Name of the output file
#$ -pe 4way 16                         # Request 1 Vis node
#$ -q vis                              # Queue name
#$ -l h_rt=6:00:00                     # runtime (hh:mm:ss) - 4 hours
```

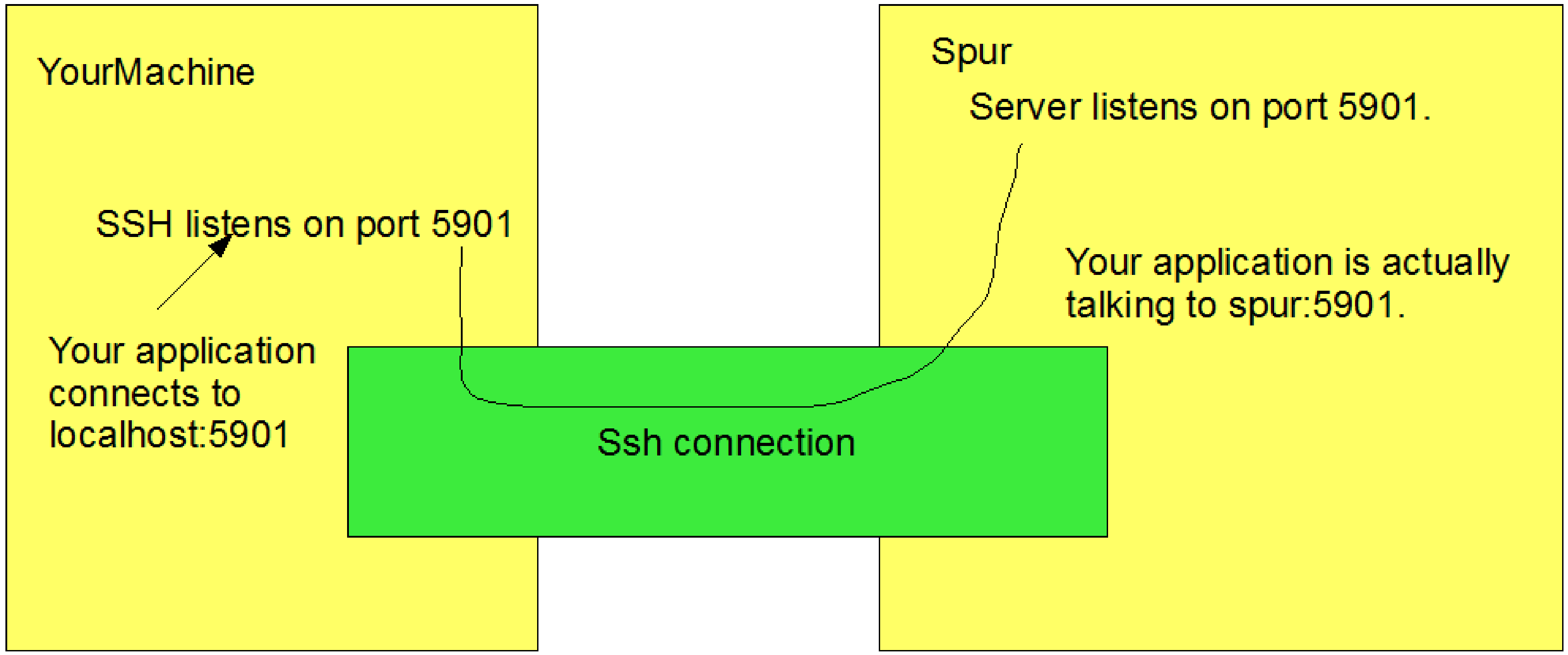
In vi, type "i" to edit. Then hit the escape key to return to *command mode.*
Save the file with ":w<return>" and quit with ":q<return>". Quit without saving with ":q!".

Contents of vncserver.out

```
vis1.ranger.tacc.utexas.edu
job execution at: Wed Oct 22 19:08:34 CDT 2008
got VNC display vis1.ranger.tacc.utexas.edu:1
VNC display number is 1
local (compute node) VNC port is 5901
got spur vnc port 5903
setting up vnc port forwarding from the head node
Your VNC server is now running!
To connect via VNC client:  SSH tunnel port 5903 to spur.tacc.utexas.edu:5903
                            Then connect to localhost:5903
```

Port to use for ssh tunneling

Secure Shell Tunneling


YourMachine
SSH listens on port 5901
Your application connects to localhost:5901
Ssh connection
Spur
Server listens on port 5901.
Your application is actually talking to spur:5901.

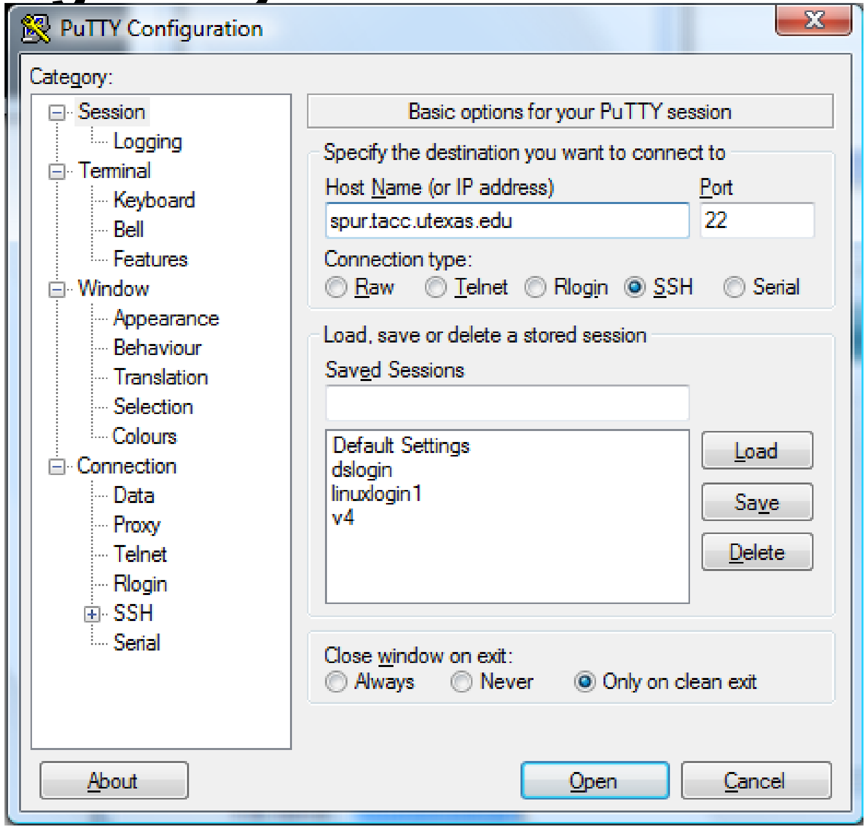
SSH tunnel to Spur

- VNC is not secure, use ssh to encrypt connection
- Vncserver.out contains the port
- On UNIX based machine:

  ssh –g -L 59xx:spur.tacc.utexas.edu:59yy username@spur.tacc.utexas.edu

Tunneling on Windows using Putty

- Type hostname



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): spur.tacc.utexas.edu

Port: 22

Connection type: Raw Telnet Rlogin [✓ SSH] Serial

Load, save or delete a stored session

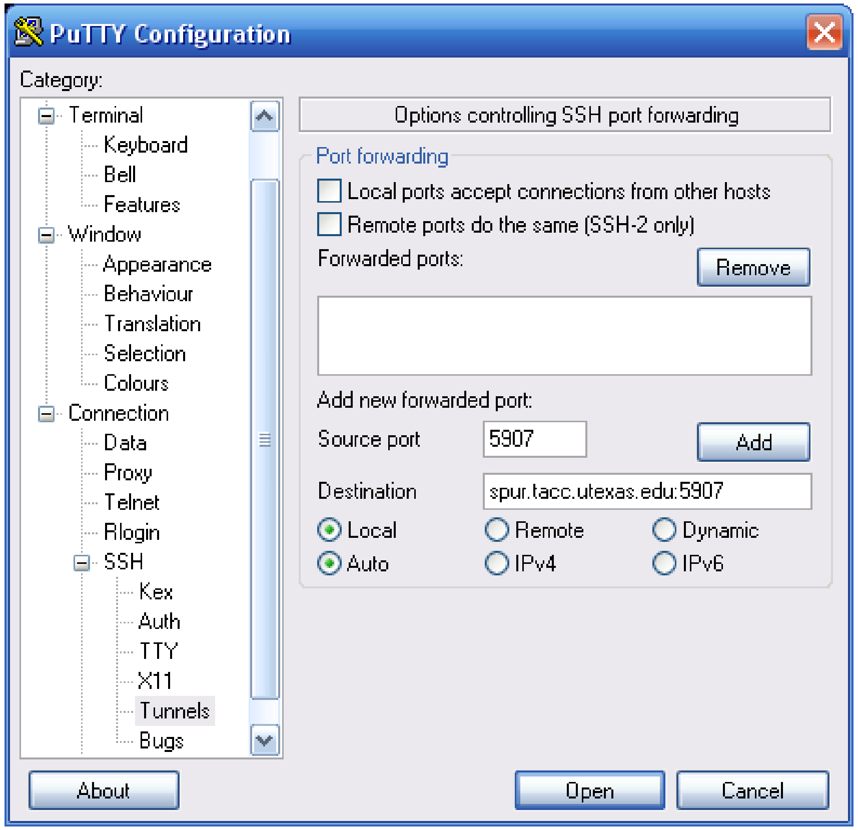
Saved Sessions

Default Settings
dslogin
linuxlogin1
v4

Close window on exit: Always Never [✓ Only on clean exit]

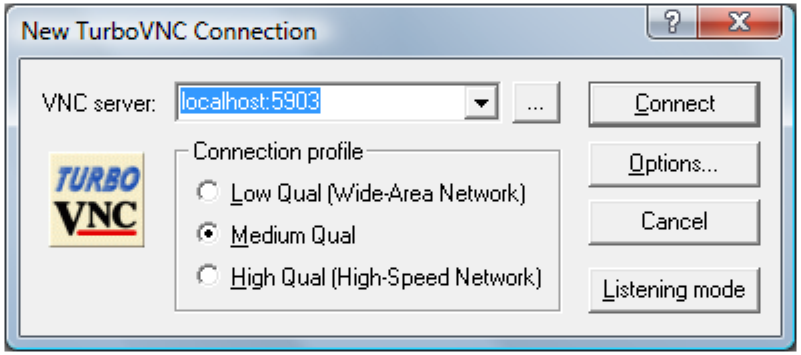
Tunneling on Windows using Putty

- Add spur.tacc.utexas.edu:port



Local ports accept connections from other hosts [ ]
Remote ports do the same (SSH-2 only) [ ]
Source port: 5907
Destination: spur.tacc.utexas.edu:5907
[✓ Local] Remote Dynamic
[✓ Auto] IPv4 IPv6

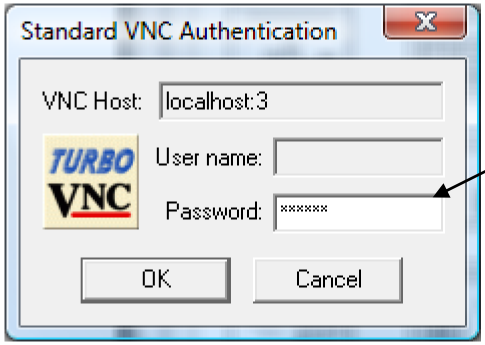
Connecting with VNC



New TurboVNC Connection

VNC server: localhost:5903

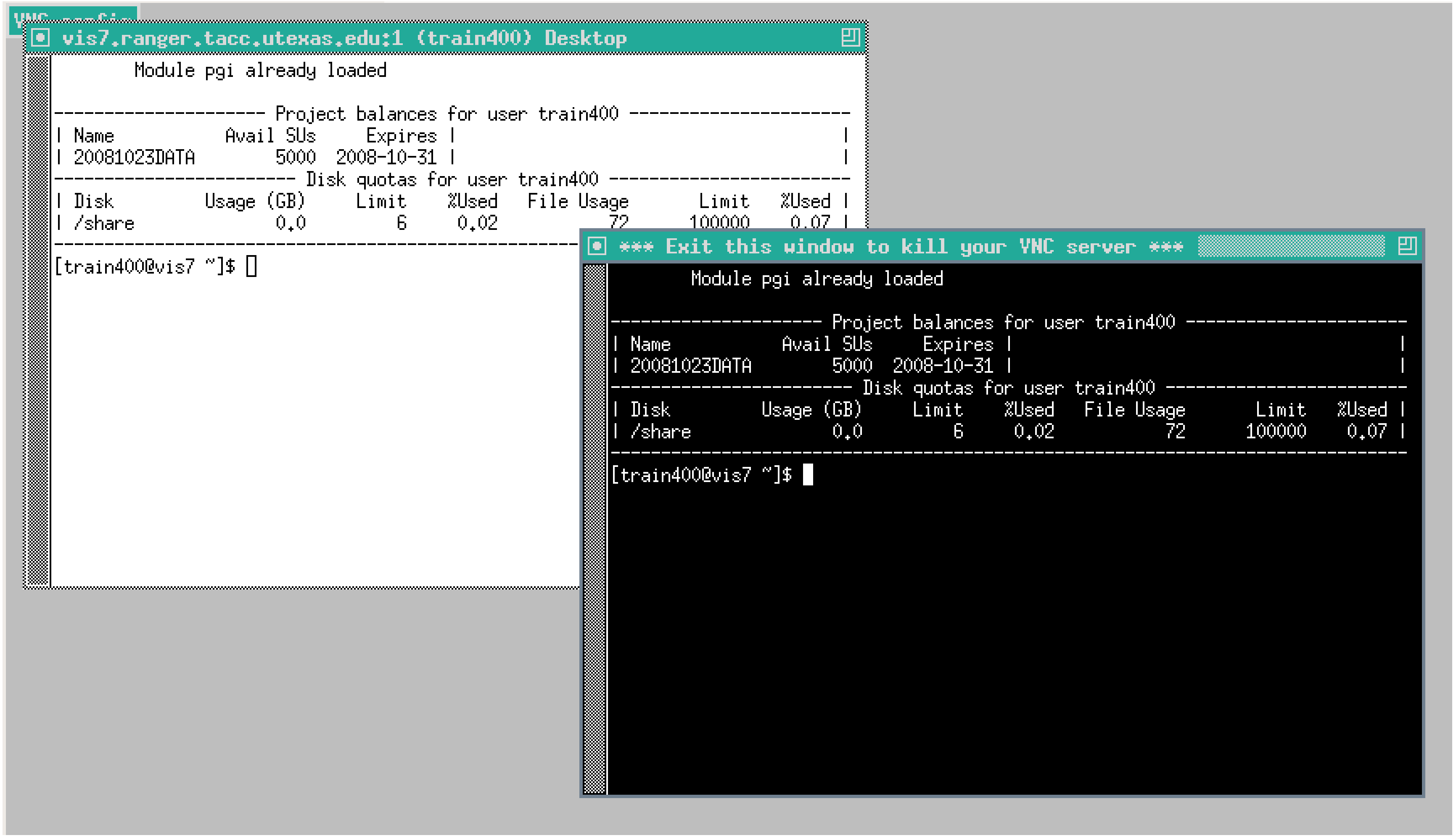
Connection profile
- Low Qual (Wide-Area Network)
- [✓ Medium Qual]
- High Qual (High-Speed Network)



Standard VNC Authentication

VNC Host: localhost:3
User name:
Password: ******

This is that vncpasswd.



vis7.ranger.tacc.utexas.edu:1 (train400) Desktop

```
        Module pgi already loaded

--------------------- Project balances for user train400 ----------------------
| Name            Avail SUs      Expires |                                      |
| 20081023DATA         5000  2008-10-31 |                                      |
------------------------ Disk quotas for user train400 ------------------------
| Disk          Usage (GB)     Limit    %Used   File Usage       Limit   %Used |
| /share               0.0         6     0.02           72      100000    0.07 |
-------------------------------------------------------------------------------
[train400@vis7 ~]$ 
```

*** Exit this window to kill your VNC server ***

```
        Module pgi already loaded

--------------------- Project balances for user train400 ----------------------
| Name            Avail SUs      Expires |                                      |
| 20081023DATA         5000  2008-10-31 |                                      |
------------------------ Disk quotas for user train400 ------------------------
| Disk          Usage (GB)     Limit    %Used   File Usage       Limit   %Used |
| /share               0.0         6     0.02           72      100000    0.07 |
-------------------------------------------------------------------------------
[train400@vis7 ~]$ 
```

In the VNC terminal

- Don't exit the black window until you're done.
- If you submitted the job for this group of four
  - start three vncservers. They will start on 5902, 5903, 5904.
  - Type “vncserver” three times.
  - If you exit the black window, everybody quits. Start a new xterm with “xterm&”. Then click to tell it where to place the xterm. Work there.
  - And, later, you can kill display with vncserver –kill :2
- If you are one of the teammates
  - set up a tunnel to the right port: vis4:5903 becomes spur:5943. Use your friend's password: muppet.
  - You will be in under their account.

In the VNC terminal

- Don't exit the black window until you're done.
- Spur% module delete mvapich mvapich2
- Spur% module swap pgi intel
- Spur% module load openmpi/1.3
- spur% module load vis
- spur% module load paraview
- “vglrun” is used for all OpenGL programs on VNC.
- spur% vglrun paraview
- Play with RectGrid2.vtk.

Parallel Visualization

- Three types of parallelism to think about:
  - Task parallelism – passing results to 1 process for rendering

	Timesteps					
Processes		1	2	3	4	5
	1		Read file 1	Isosurface 1	Cut Plane 1	
	2			Read file 2	Streamlines 2	Render
	3	Read file 3	Triangulate 3	Decimate 3	Glyph 3	

Parallel Visualization

- Three types of parallelism to think about:
  - Pipeline parallelism – useful when processes have access to separate resources or when an operation requires many steps

		Timesteps				
Processes		1	2	3	4	5
	1	Read file 1	Read file 2	Read file 3		
	2		Isosurface 1	Isosurface 2	Isosurface 3	
	3			Render 1	Render 2	Render 3

Parallel Visualization

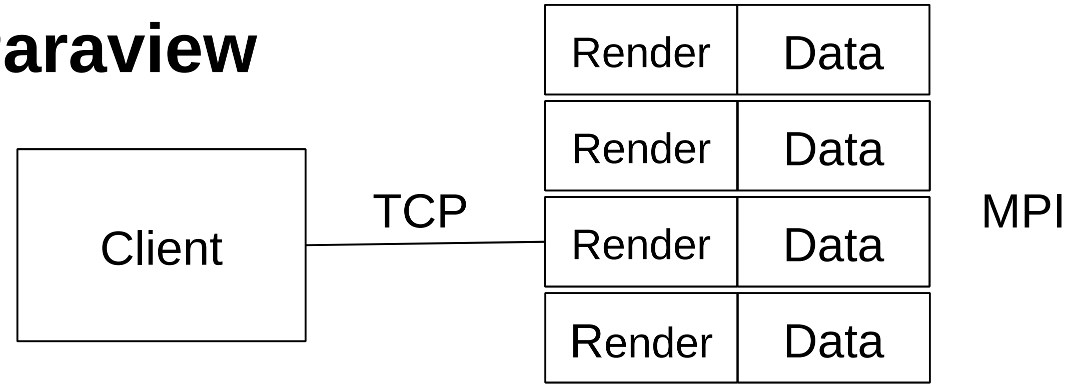
- Three types of parallelism to think about:
  - Data parallelism – data set is partitioned between the processes and all process execute same operations on the data. Scales well as long as the data and operations can be decomposed.

	Timesteps			
Processe s		1	2	3
	1	Read partition 1	Isosurface partition 1	Render partition 1
	2	Read partition 2	Isosurface partition 2	Render partition 2
	3	Read partition 3	Isosurface partition 3	Render partition 3

Parallel Paraview

- Paraview has three main logical components:
  - Client server responsible for user interface of the application
  - Data server reads and processes data sets to create final geometric models. Each process is told which partition of the data it should load
  - Render server is responsible for rendering the final geometry. The render server can run in parallel if it configured to do so.
- All three can be run in a single process
- Data and render server can be embedded into one process.

Parallel Paraview



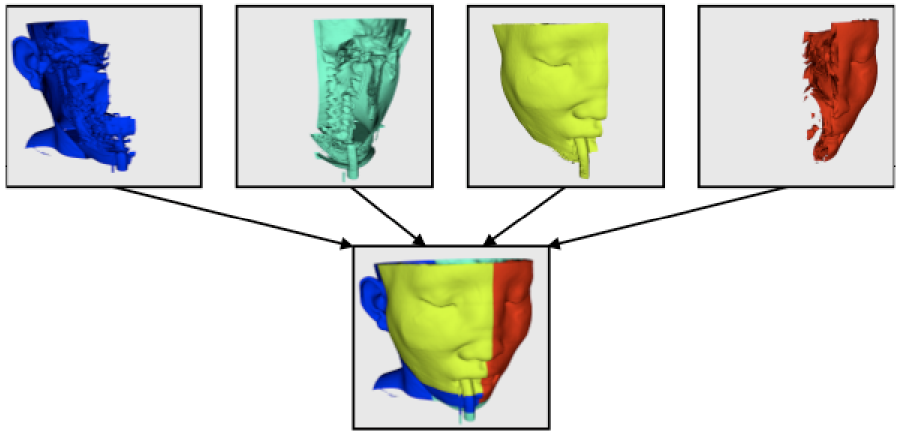


Image from Large Scale Visualization with ParaView Supercomputing 2008 Tutorial

SMP on Spur: Load Paraview Engines

- Quit current paraview session and start again with ampersand.
- ibrun starts MPI tasks with the server (data+render)
- Other teammates use –sp=11112-4 to use different ports.

```
[username@vis5 ~]$ module swap pgi intel
[username@vis5 ~]$ module delete mvapich mvapich2
[username@vis5 ~]$ module load openmpi/1.3
[username@vis5 ~]$ module load vis
[username@vis5 ~]$ module load paraview
[username@vis5 ~]$ vglrun paraview&
[username@vis5 ~]$ ibrun vglrun pvserver –sp=11111
TACC: Setting up parallel environment for OpenMPI mpirun.
TACC: Setup complete. Running job script.
TACC: starting parallel tasks...
Listen on port: 11111
Waiting for client...
```

SMP on Spur: Tell Paraview to Use the Engines

- Click the "Connect" button, or select File -> Connect
- Click "Add Server"
- Enter a "Name", e.g. "manual launch"
- Click "Configure"
- For "Startup Type", select "Manual"
- Click "Save"
- Select the name of your server configuration, and click "Connect"
- In the xterm where you launched ParaView server, you should see "Client connected."

SMP on Spur: Is It Running?

- Try running “top” in an xterm in the VNC desktop.
- Compare behavior with the desktop version you ran.

Parallel Paraview

- ssh username@login3.ranger.tacc.utexas.edu
- vncpasswd
- vncserver # Look at display number it shows
  - login3.ranger.tacc.utexas.edu:3 is port 5903
- Connect with VNC from your terminal.

Starting Paraview Server

- login3% module delete mvapich mvapich2
- login3% module swap pgi intel
- login3% module load openmpi/1.3
- login3% module load vis
- login3% module load mesa
- login3% module load paraview
- qsub parallel_paraview.job

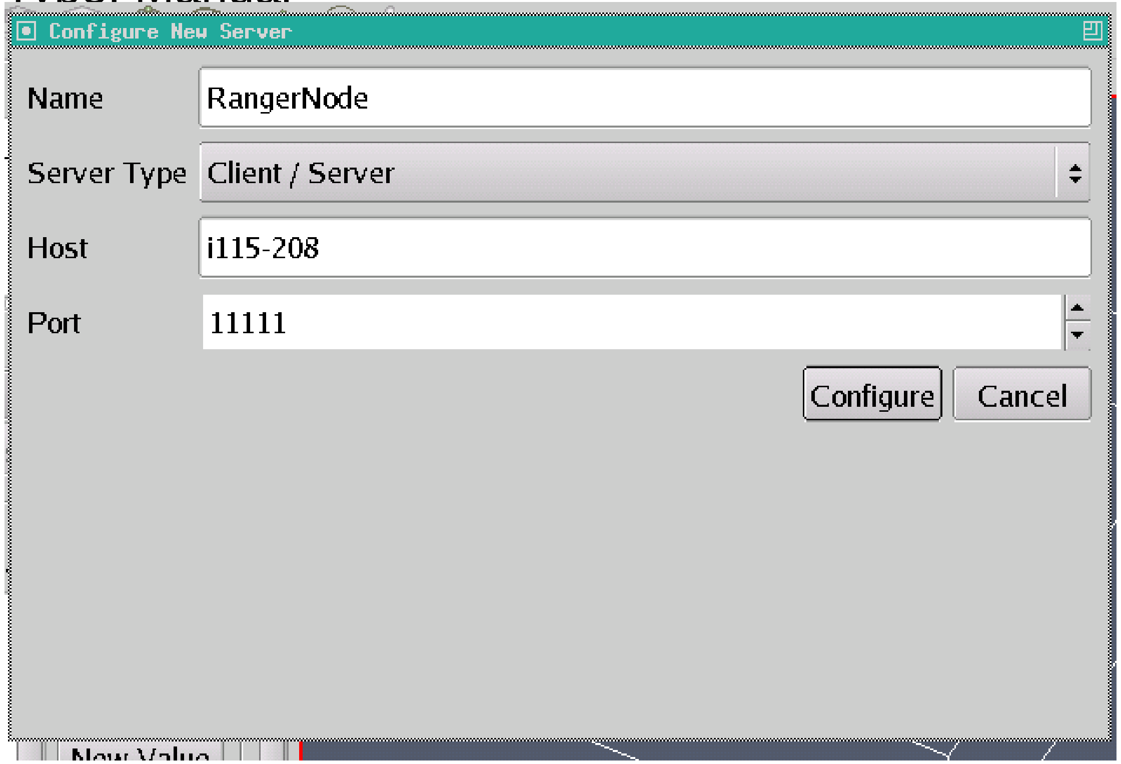
Find Your Ranger Server

- showq -u # to see when it runs
- qstat   # to see what node you got
- i115-406.ranger.tacc.utexas.edu means i115-406 is the node.
- login3% paraview

Connect to Server

In Paraview, go to File->Connect.
Click “Add Server”
Enter the node in “host”.
Click Configure.
Under Configure, select Startup Type: Manual
and click Save.

We choose “manual” because the job we submitted already started the server.


Configure New Server

Name	RangerNode
Server Type	Client / Server
Host	i115-208
Port	11111

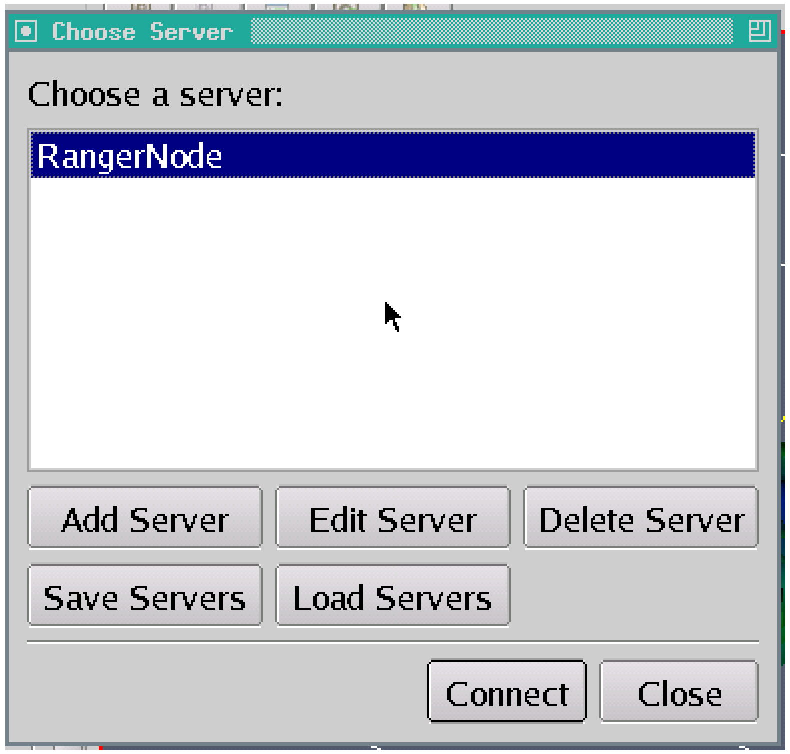
Connect to Server

Select the server and click connect.

Note: You will get a message that says rendering disabled on the server. This is okay.

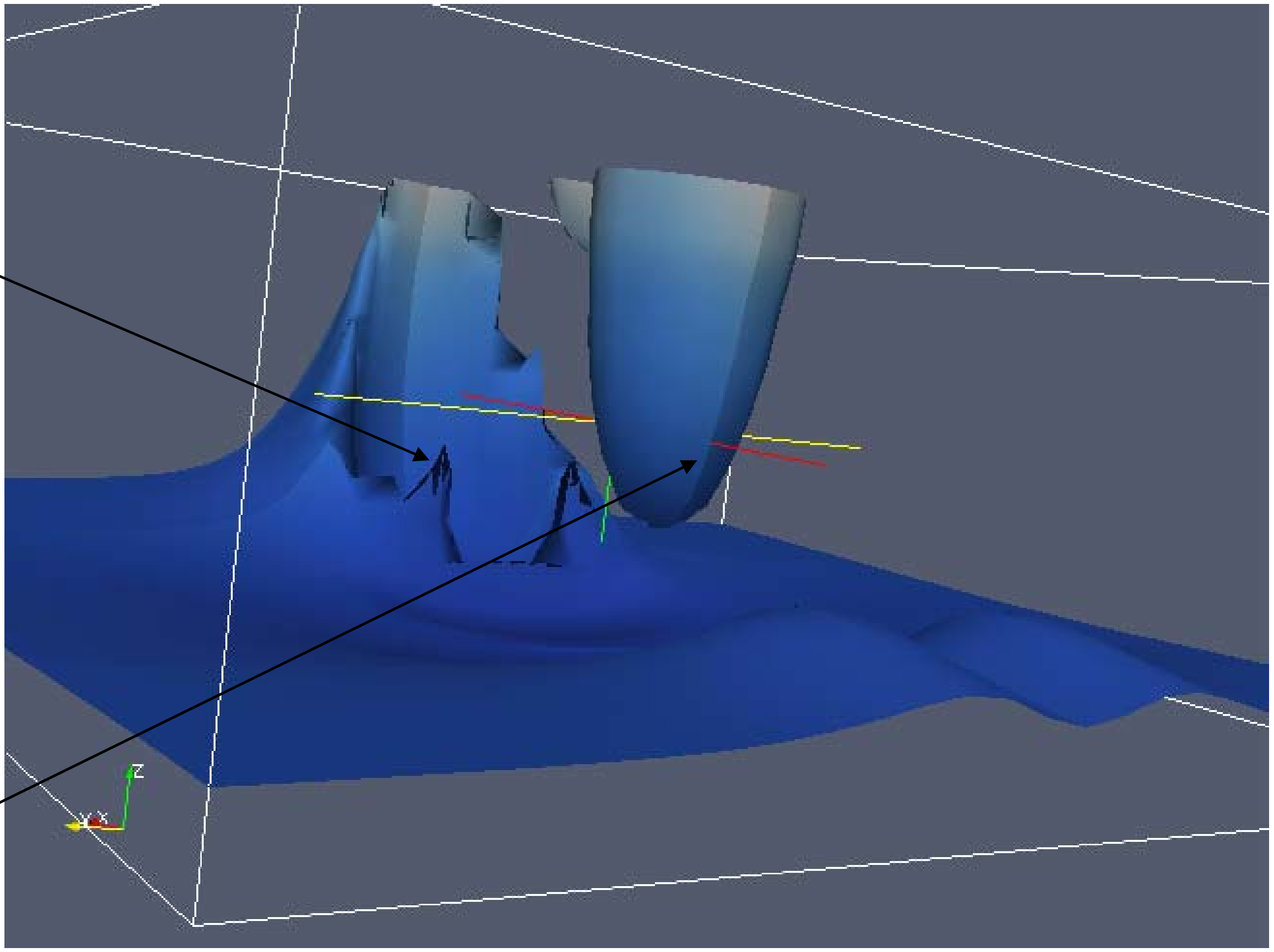
Note: You will need to configure a new server each time the compute node changes, which is likely to be each time you run Paraview.

Open RectGrid2.vtk.



Artifacts

Z-Fighting



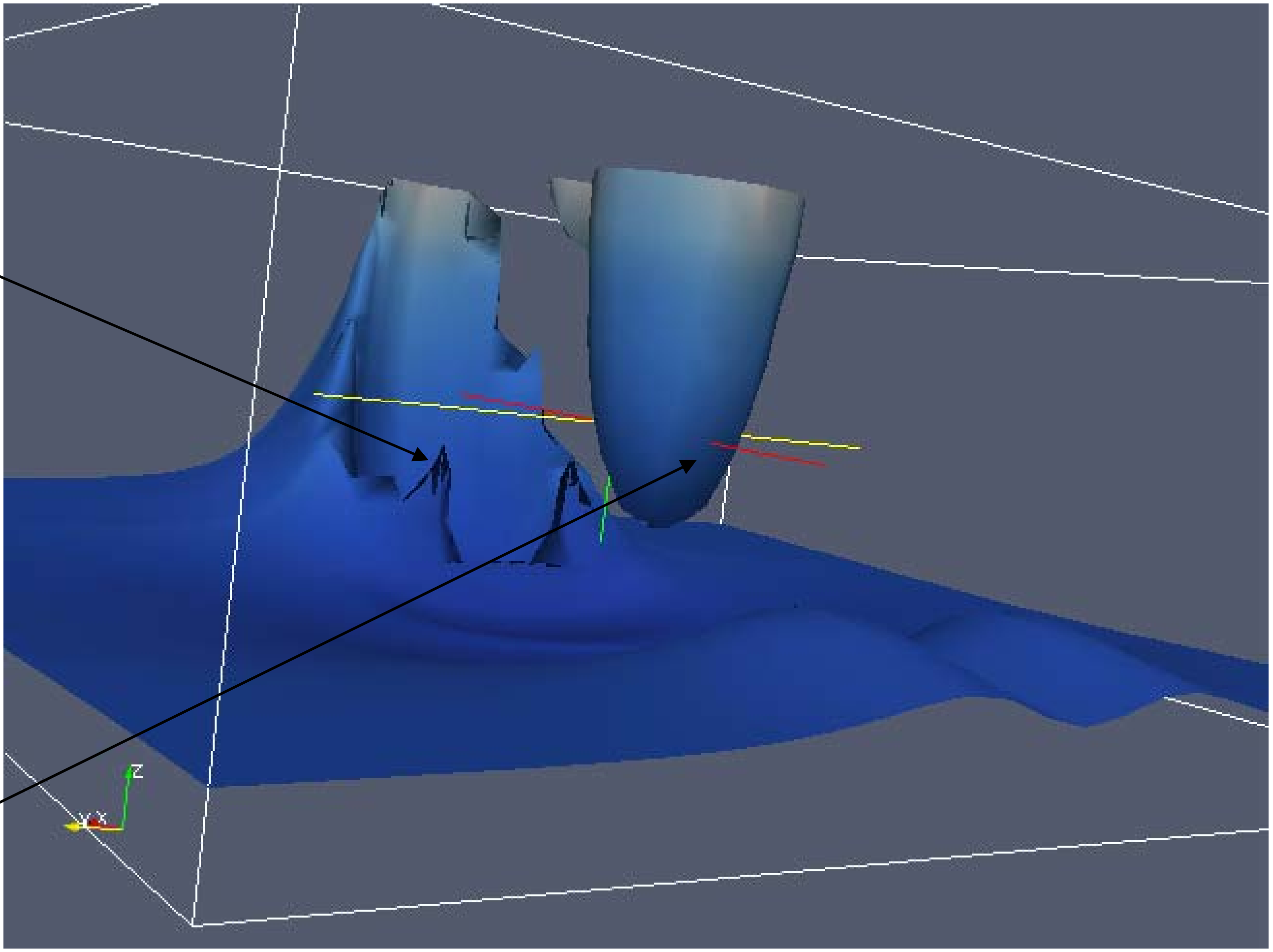
Shading
Disconnect

Add D3 filter

- Short for distributed data decomposition
- Balances unstructured data and creates ghost cells
- Use “Duplicate cells” boundary mode

Artifacts

Z-Fighting



Shading
Disconnect