Advanced Visualization Techniques

Kelly Gaither
Texas Advanced Computing Center

UT/Portugal Summer Institute
Coimbra, Portugal
July 17, 2008

Topics Covered

- Remote and Collaborative Visualization
- EnVision – simplified Interface for Visualization
- Visualizing and Animating a Time Series in Paraview
- VisIt
- Parallel Visualization

Topics Covered

- Remote and Collaborative Visualization
- EnVision – simplified Interface for Visualization
- Visualizing and Animating a Time Series in Paraview
- VisIt
- Parallel Visualization

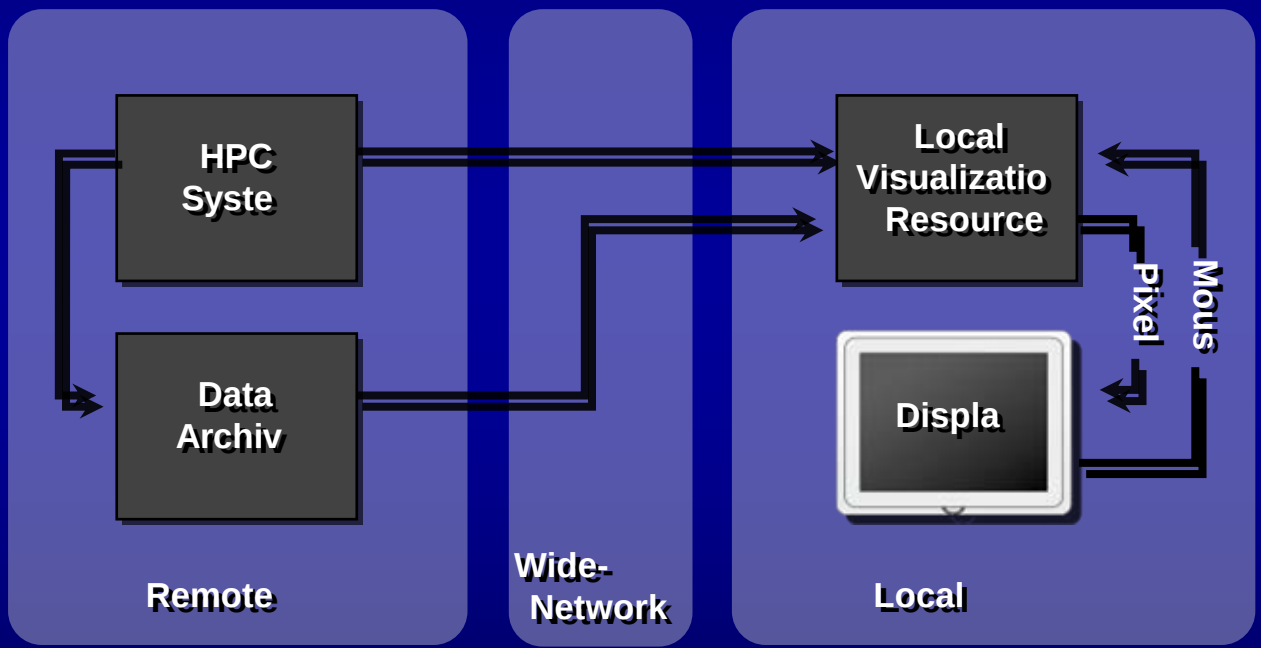
Remote Visualization Motivation

- It is no longer necessarily practical to move tera and peta-scale data from instruments or HPC systems to user's local site
- Certain data is sensitive (e.g. data from industry, government, academia)
- Growth of data has outstripped the graphics capabilities of single GPUs and aggregation of GPUs requires specialized knowledge

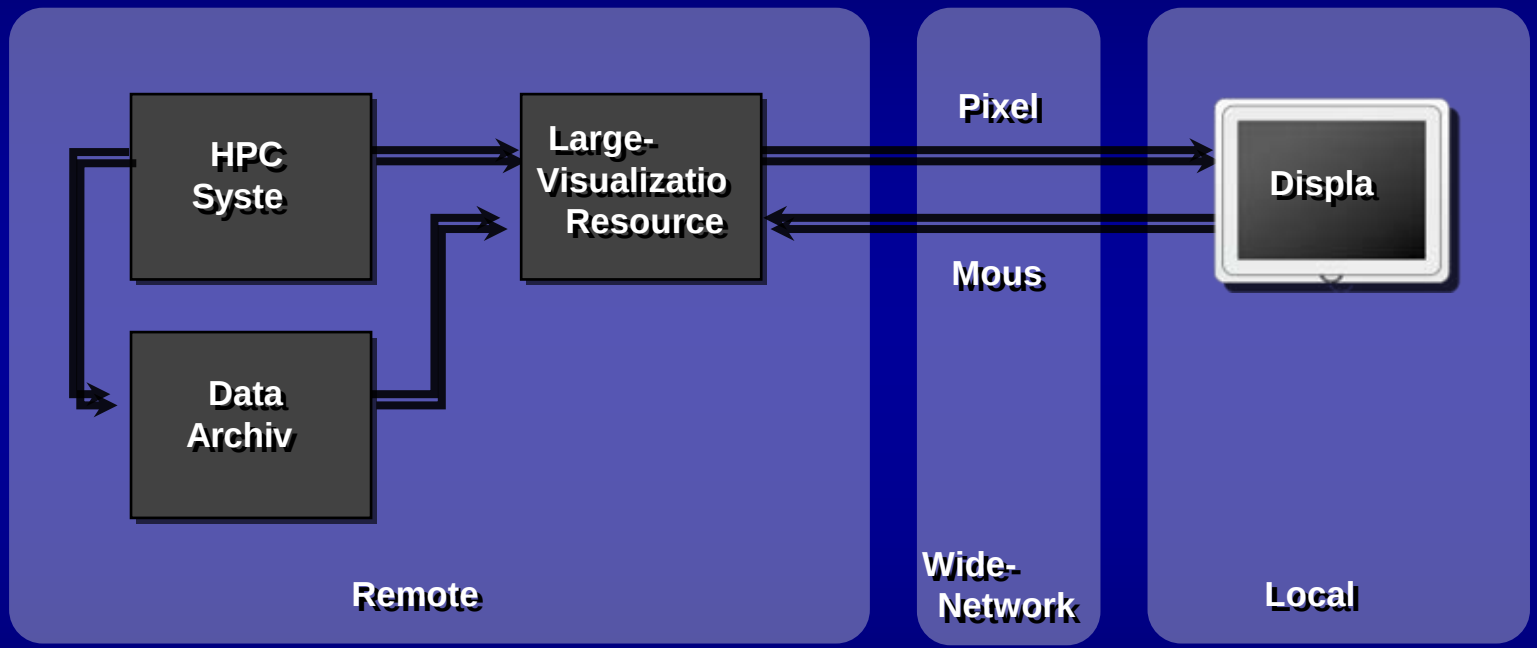
Remote Visualization Challenges

- Latency (wide-area network and GPU read-back)
- Quality of service and the user interface (scheduling, ease of access, and usability)
- Wide-area network bandwidth
  (1280 * 1024 * 12 * 24 = 360 MBps uncompressed)

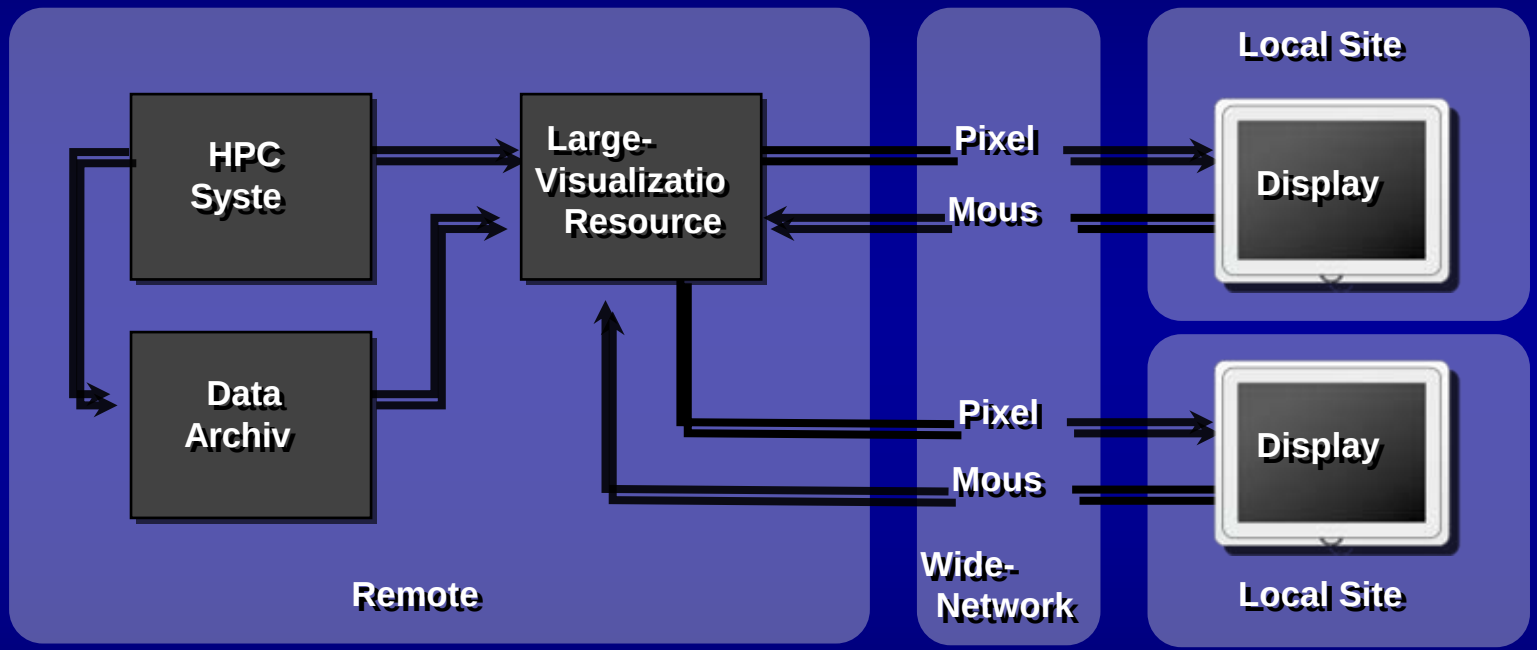
Old Model
(No Remote Capability)



New Model
Remote Capability



New Model
Multi-User and Collaborative Capability



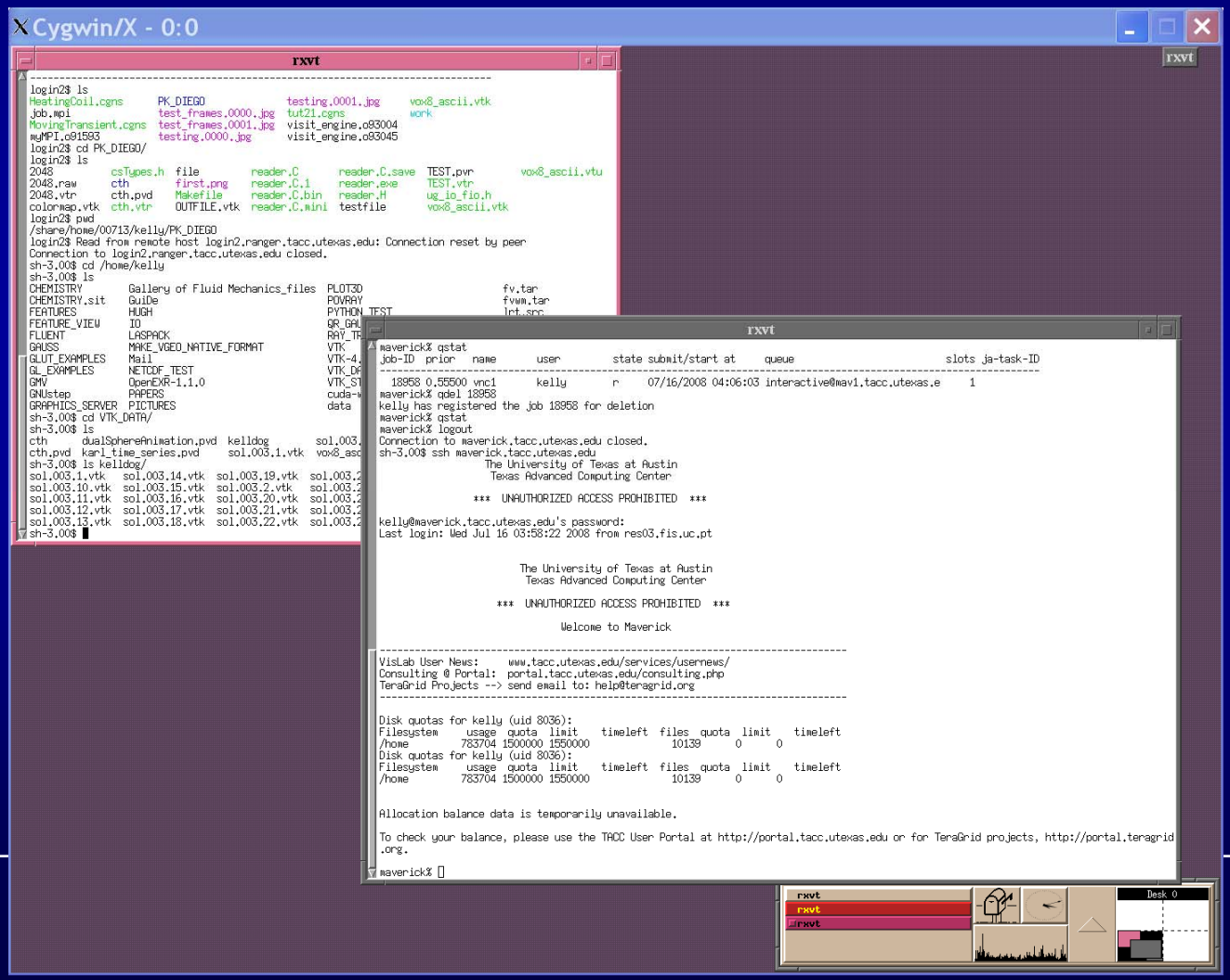
Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 1: Download a vnc client. Recommended option is TurboVNC which can be downloaded from:

http://www.virtualgl.org/Downloads/TurboVNC

Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 2: ssh maverick.tacc.utexas.edu



Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 3: On maverick, set up your vnc password. This only needs to be done once, but is attached to the machine you are currently logged into.

Type ‘vncpasswd’ and hit return. This will prompt you for a password. Type in something that you can easily remember and hit enter. You will then be asked to verify the password.

Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 4: type

‘qsub -v GEOMETRY=1600x1200 -l h_rt=1:00:00 -l gfx=1 /usr/local/qsub/RUN.vnc’

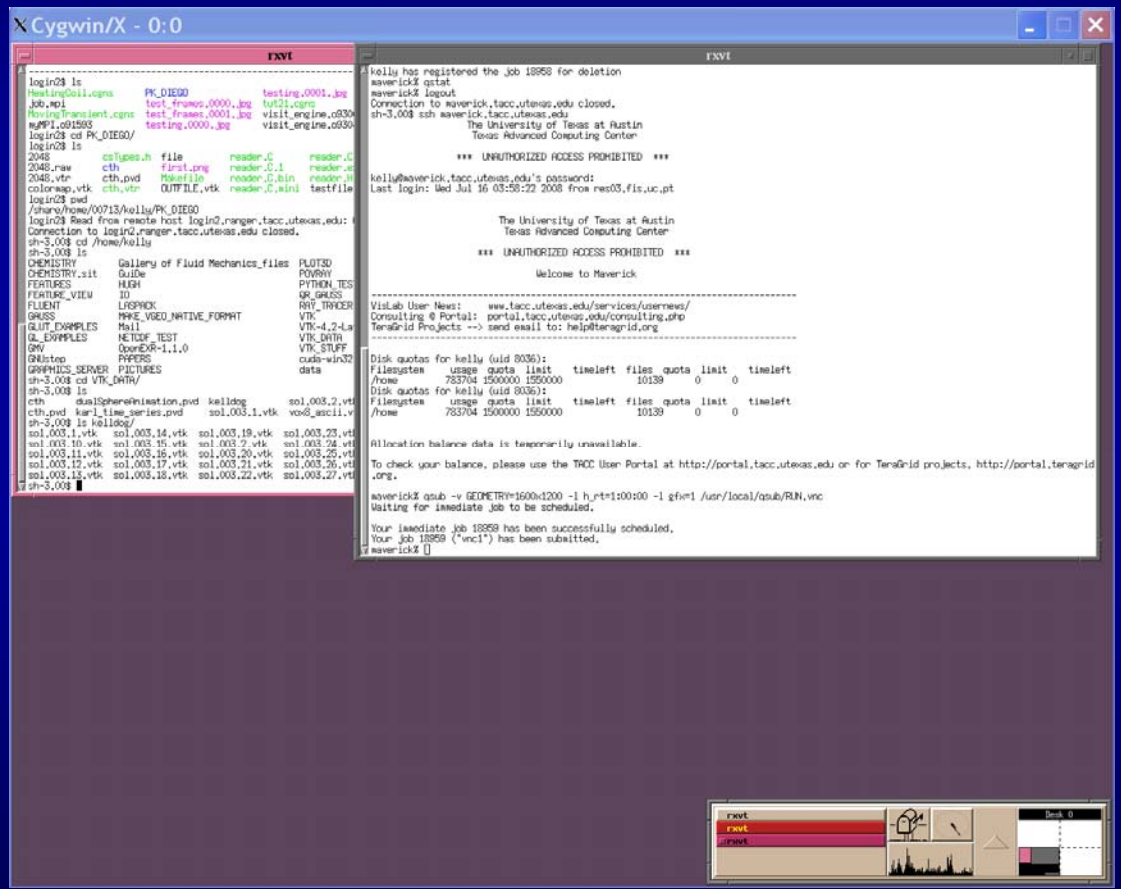
qsub = Method to submit job to the queue

GEOMETRY sets the size of the vnc window

-l h_rt=1:00:00 sets the job run time to 1 hour

-l gfx=1 requests 1 graphics card

/usr/local/qsub/RUN.vnc is the script to start the vnc session



Starting a Remote Visualization Job on Maverick (Lab Exercise)

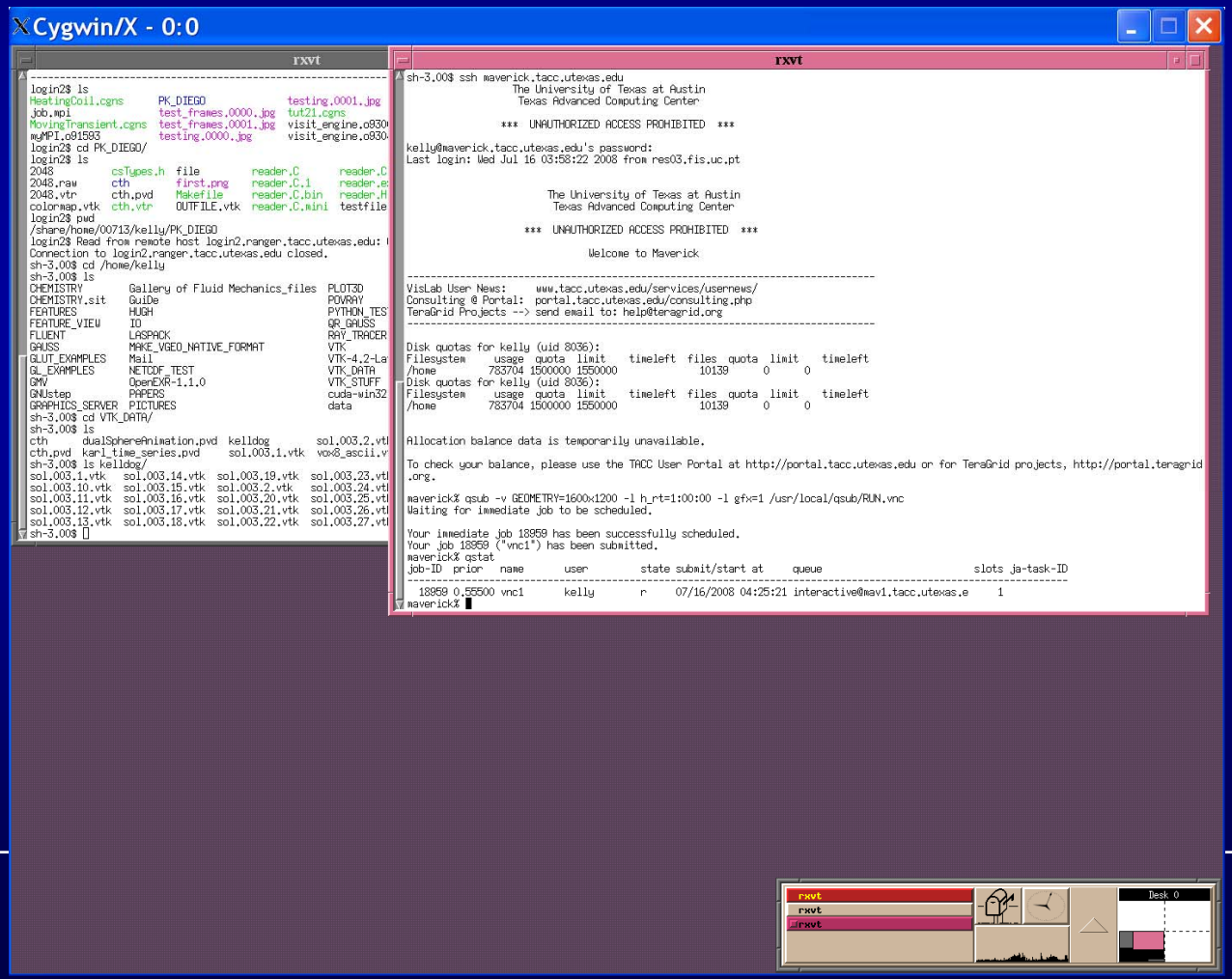
Step 5: type ‘qstat’ to see the status of your job

q = job is still queued

t = job is still being transferred to the SGE manager

e = job has errorred out and possible causes can be found in the file vnc.log in your home directory

r = job is running



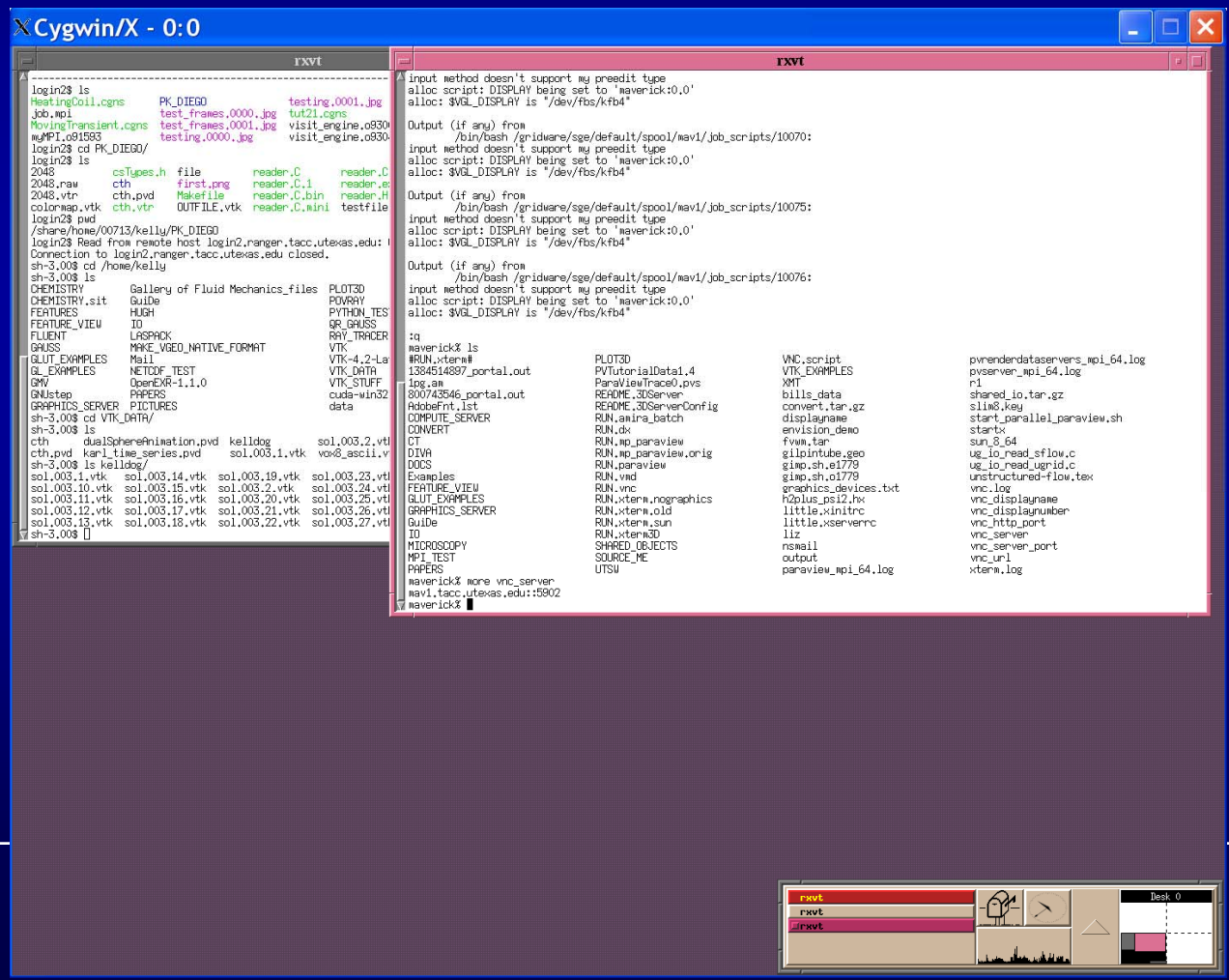
Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 6: Find the display that your job is running on by looking in the file vnc_server in your home directory

You will see a display address, for example:

mav1.tacc.utexas.edu:5902

This tells you to connect to display 2 on mav1.tacc.utexas.edu

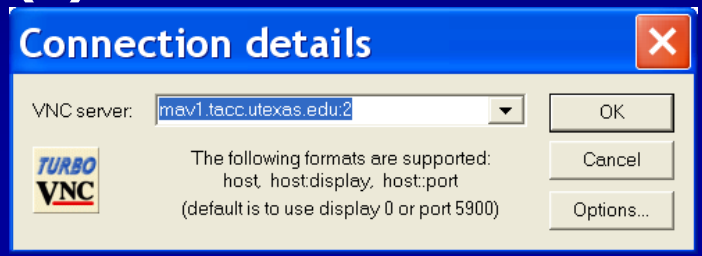


TACC

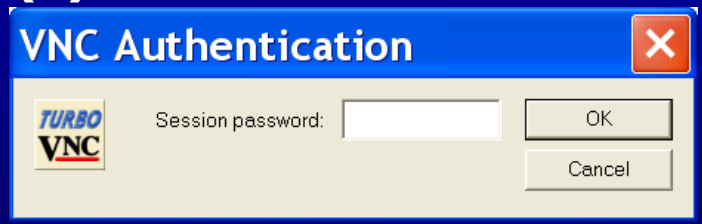
Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 7: Connect to that display by using a vnc client

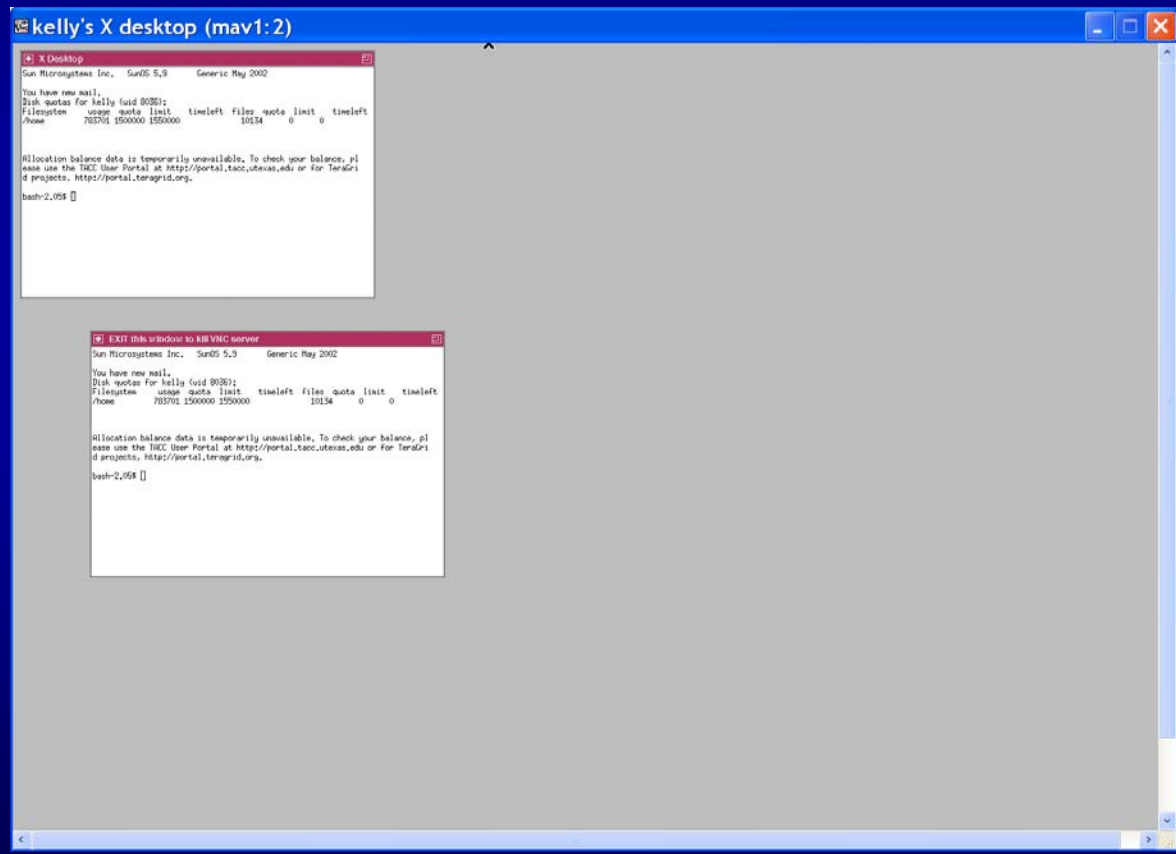
(1)



(2)



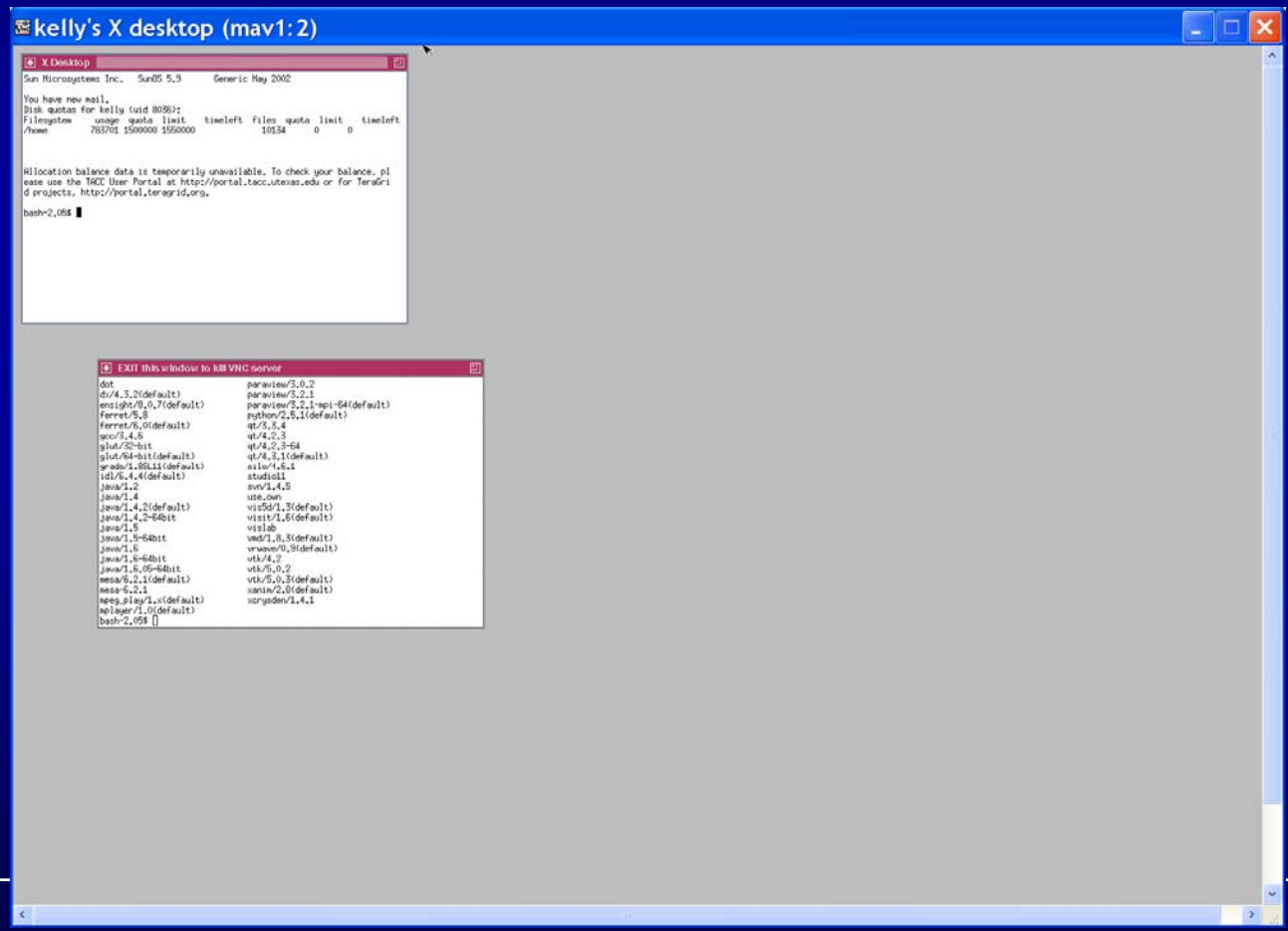
(3)



Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 8: Type ‘module avail’ in the xterm.

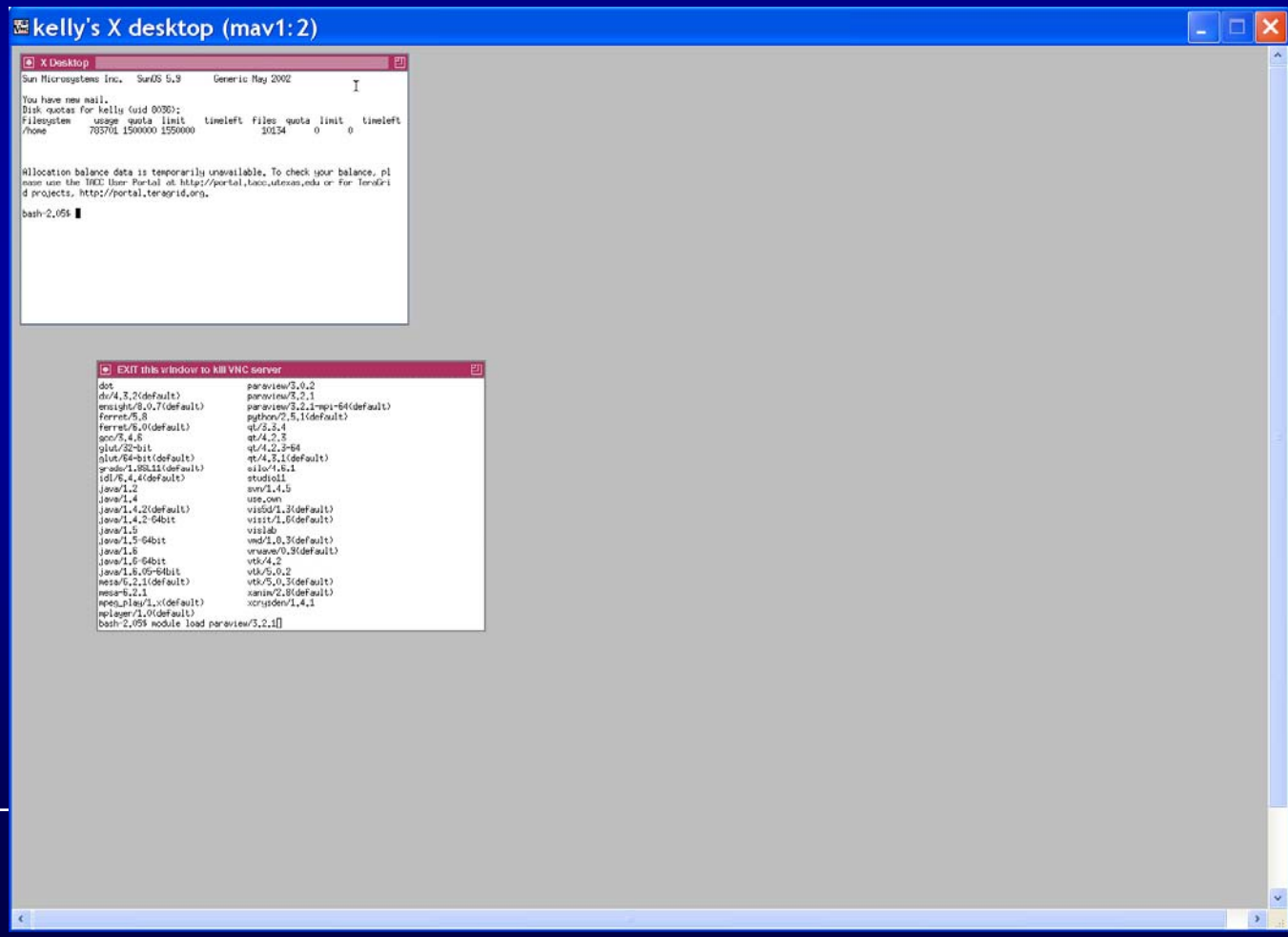
This will tell you which programs are available to load and run.



Starting a Remote Visualization Job on Maverick (Lab Exercise)

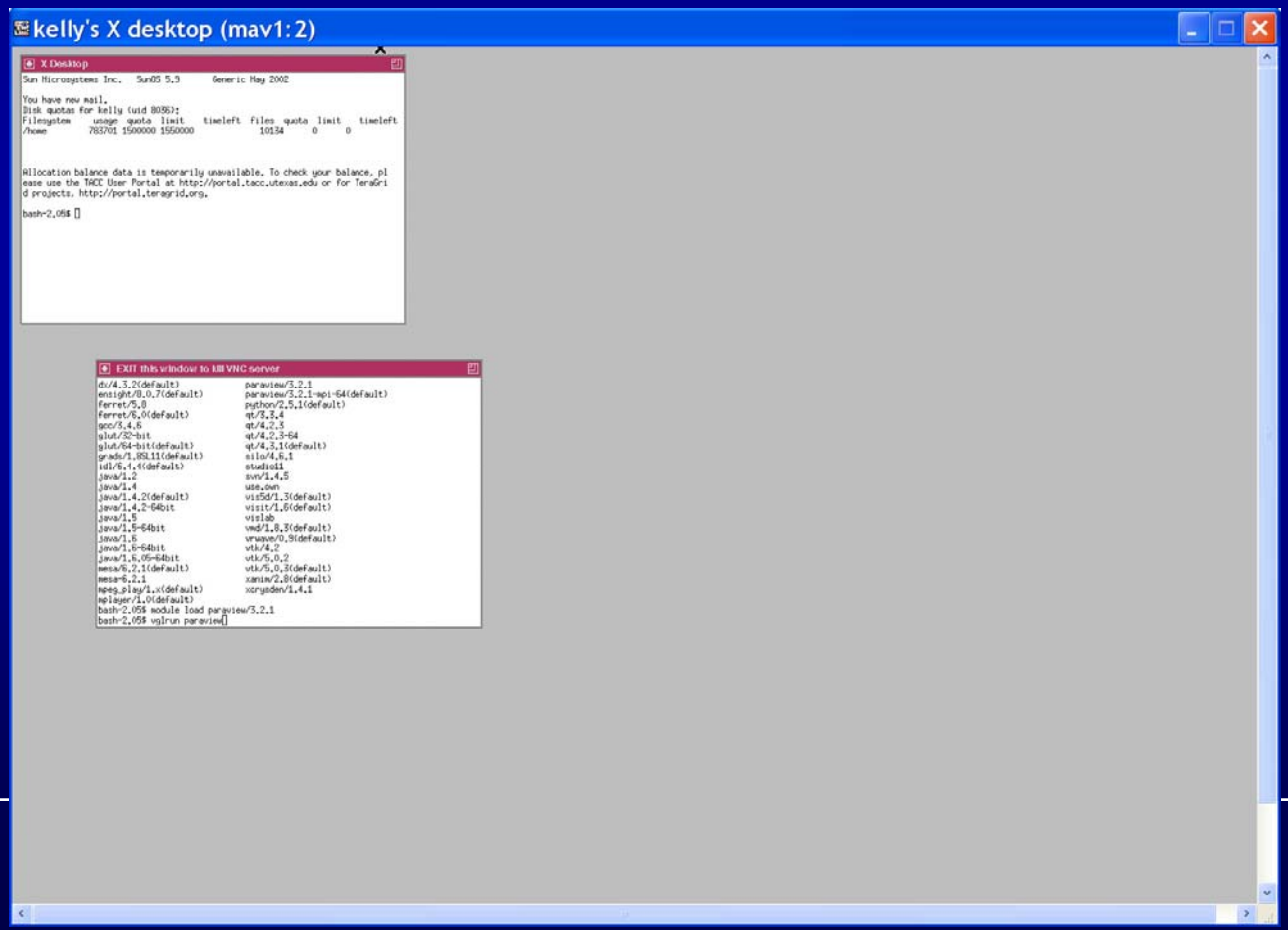
Step 9: Type ‘module load paraview/3.2.1’

This will load paraview 3.2.1 and make it available for running.

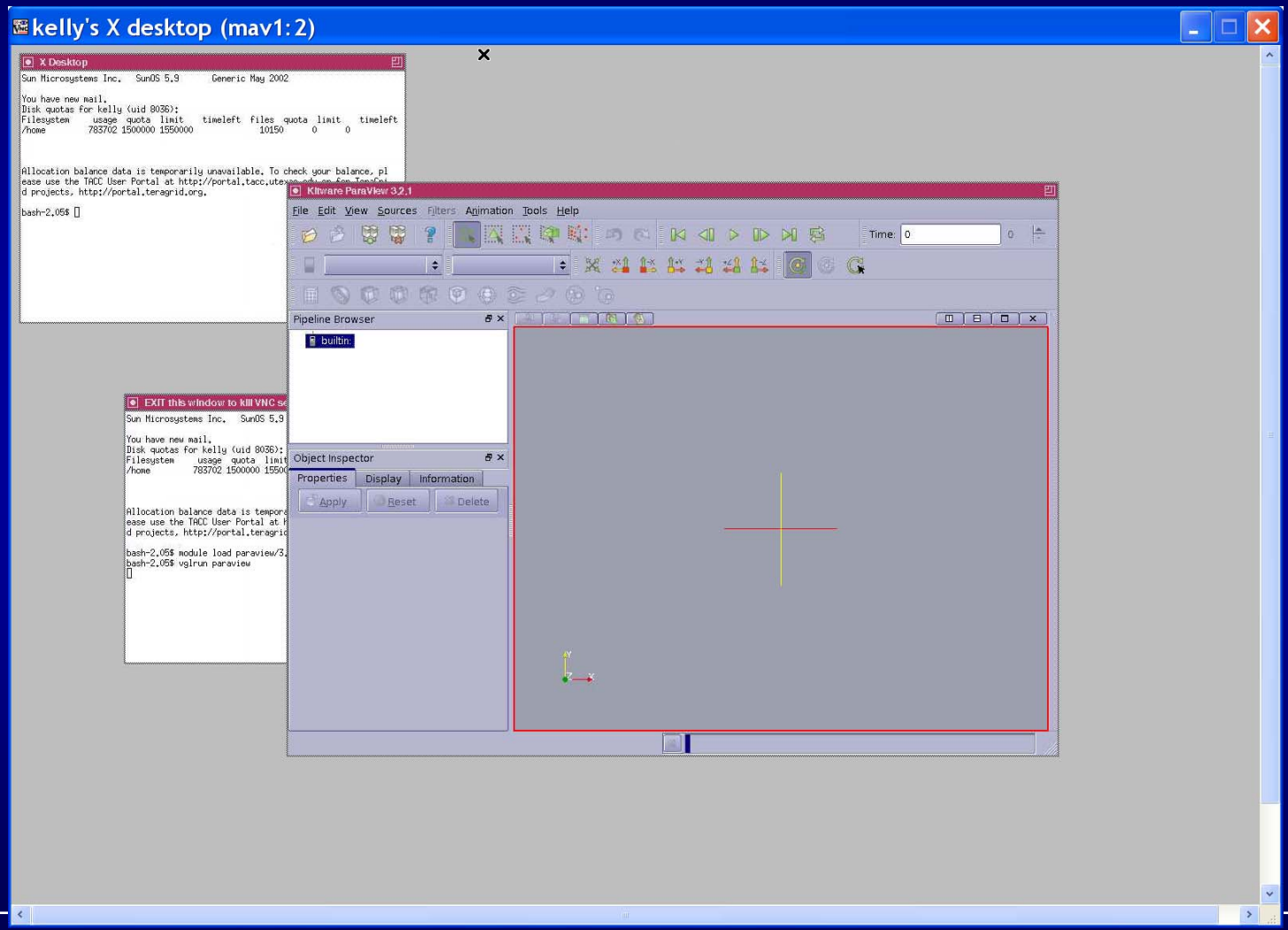


Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 10: Type ‘vglrun paraview’. vglrun is necessary to run any OpenGL based applications through the vnc viewer.

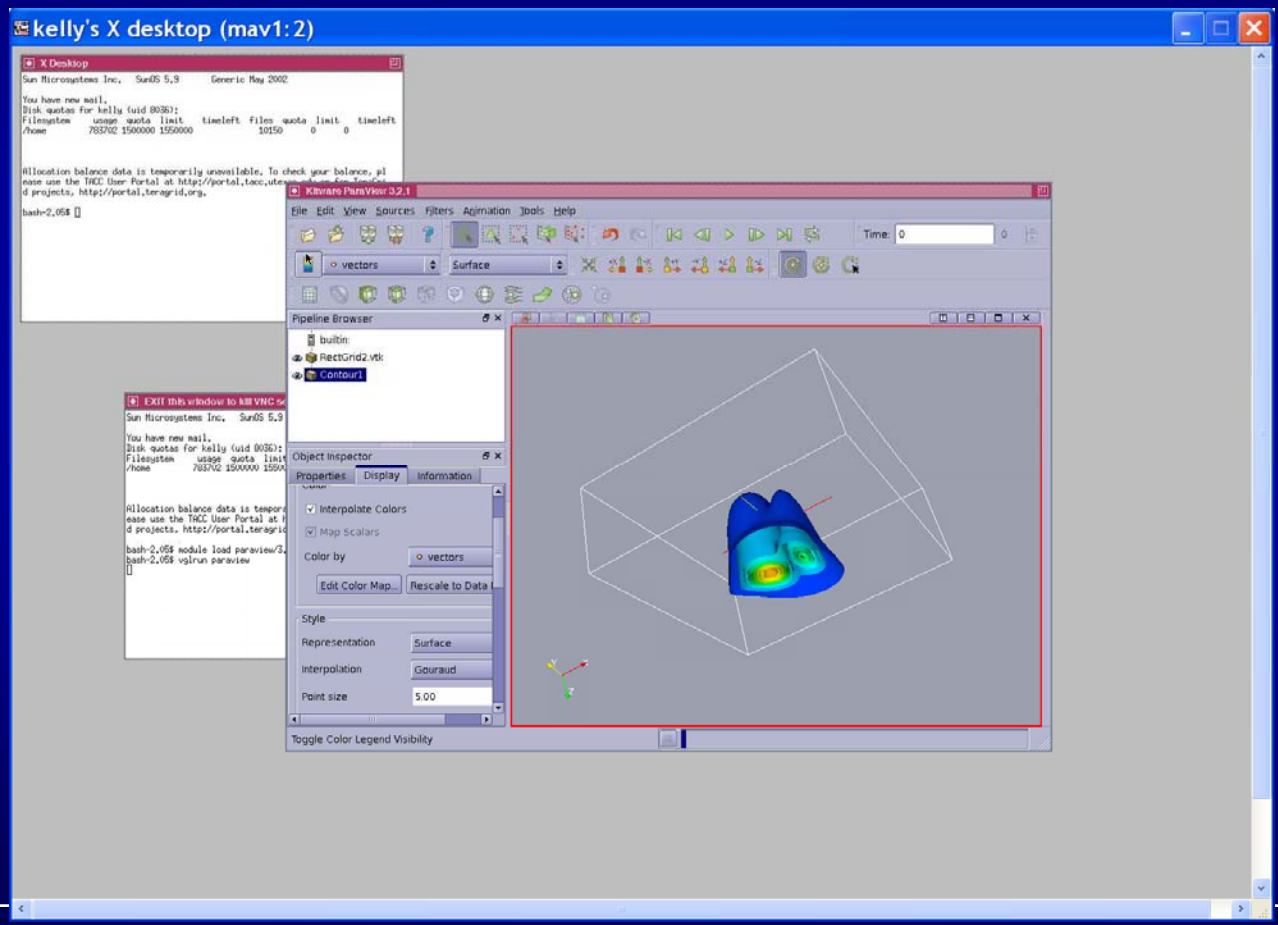


Starting a Remote Visualization Job on Maverick (Lab Exercise)



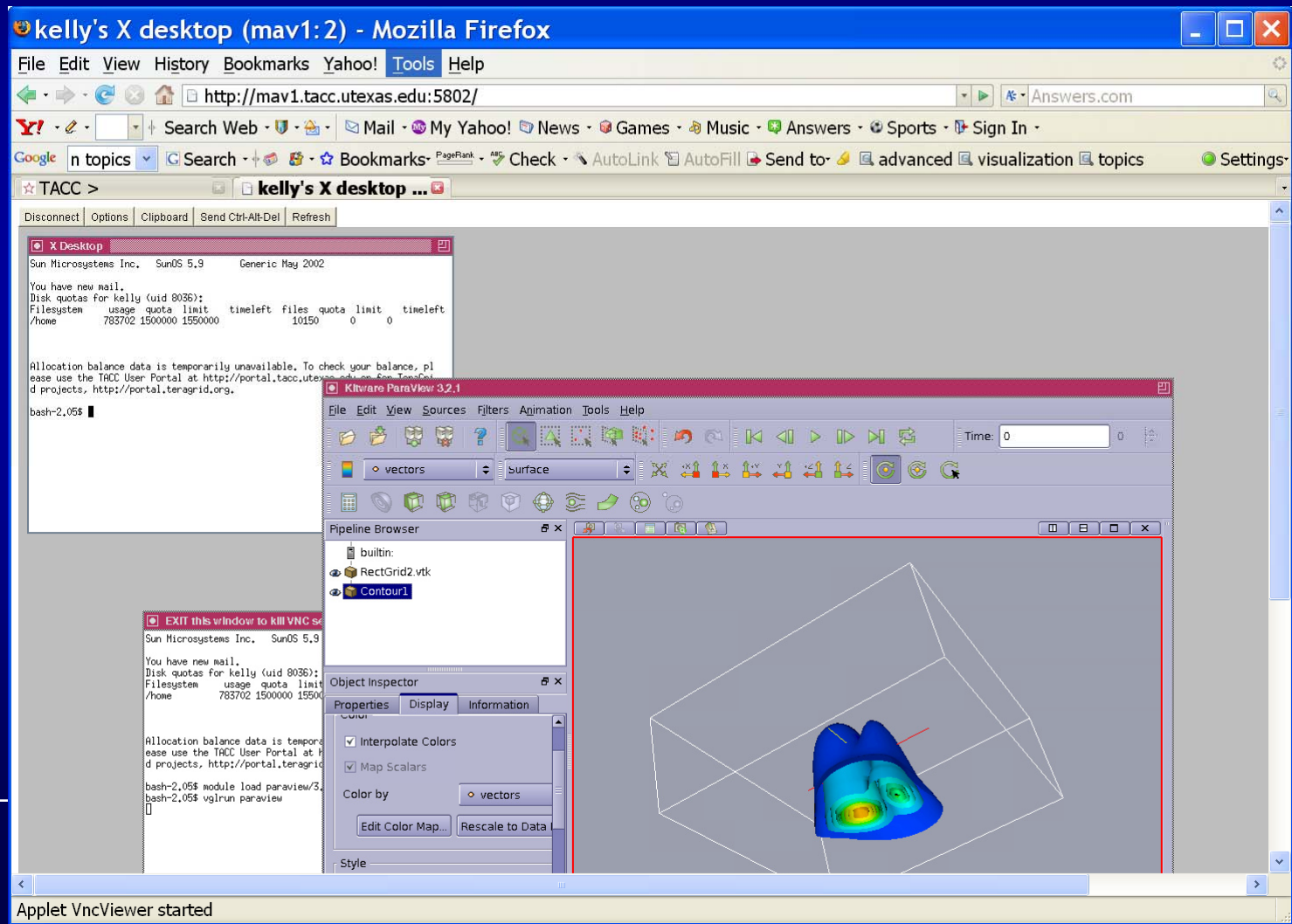
Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 11: Load the RectGrid2.vtk dataset. It can be found in /work/utexas/staff/kelly/PORTUGAL



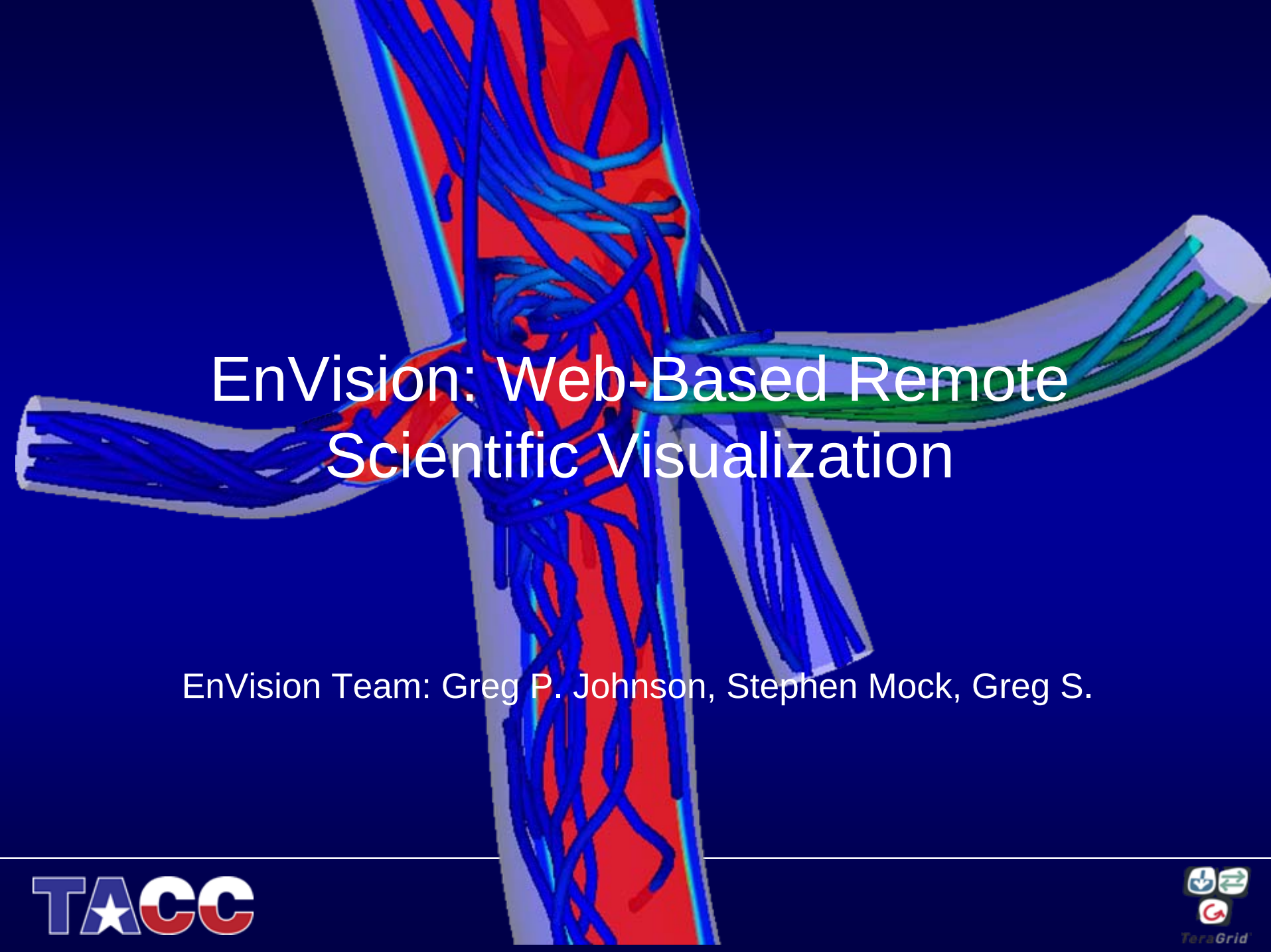
Starting a Remote Visualization Job on Maverick (Lab Exercise)

Step 12: You can also view through a web browser. The web address can be found in vnc_url in your home directory.



Topics Covered

- Remote and Collaborative Visualization
- EnVision – simplified Interface for Visualization
- Visualizing and Animating a Time Series in Paraview
- VisIt
- Parallel Visualization



EnVision: Web-Based Remote Scientific Visualization

EnVision Team: Greg P. Johnson, Stephen Mock, Greg S.

TACC

TeraGrid

Motivation for EnVision

- Most visualization software takes significant time and effort to learn effectively
- EnVision was created with these audiences in mind:
  1. Computational scientists with little to no visualization experience
  2. Users who want to quickly import and visualize their data from anywhere

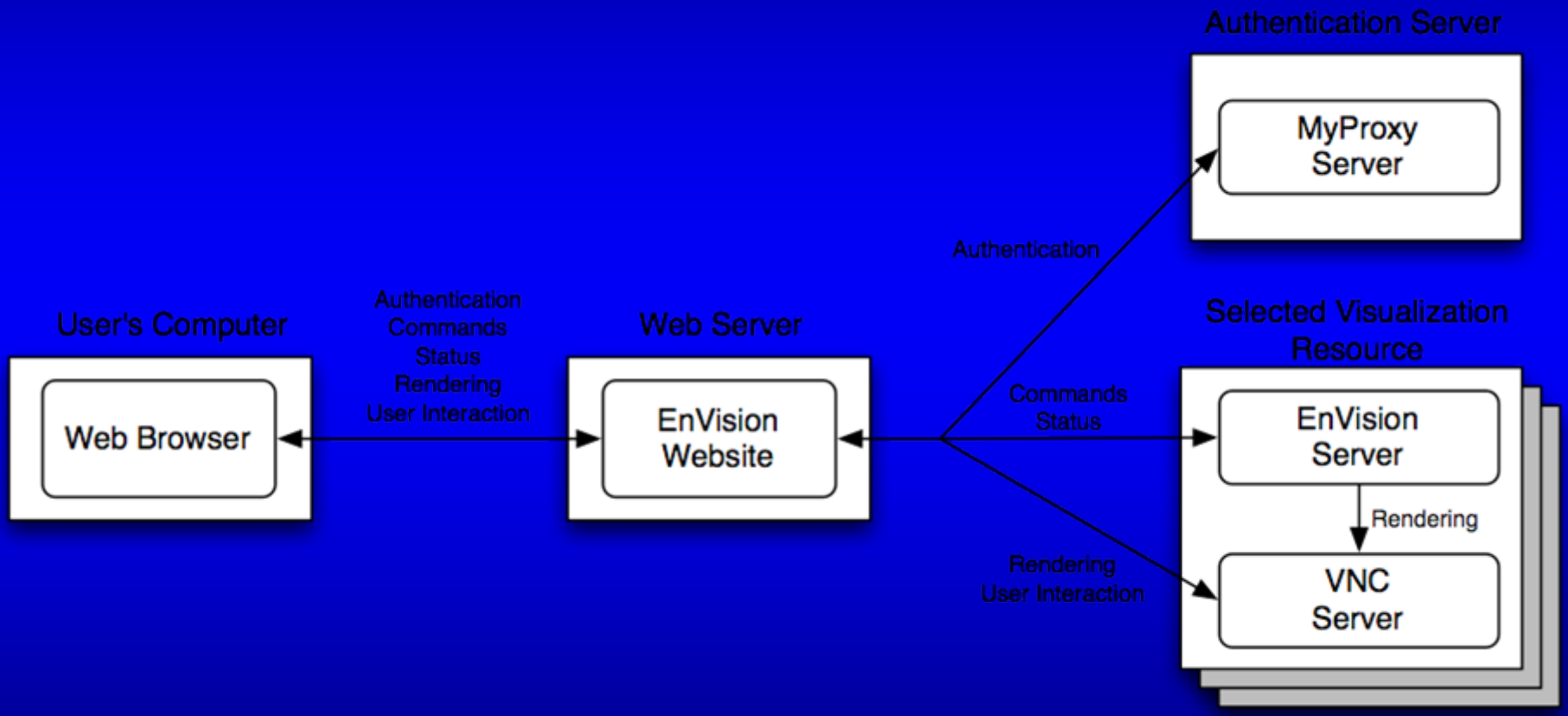
EnVision Goals: Simple, Intuitive, Immediate

- Web-based visualization application with a wizard-style interface
- Does not aim to supersede or replace current visualization applications
- Focus is on simplicity and ease-of-use
- Semi-automates the data importation process
- Presents visualization methods, with examples, that are applicable to their data set

EnVision Guides the User Through the Process

1. Authenticates user, shows visualization resources, and automatically starts a remote visualization job on selected resource
2. Imports data in nearly any format based on a series of intuitive questions
3. Presents user with examples of visualization methods to add
4. Builds the visualization using reasonable defaults for color mapping, etc.
5. Renders the visualization and allows user to interact with it and save snapshots

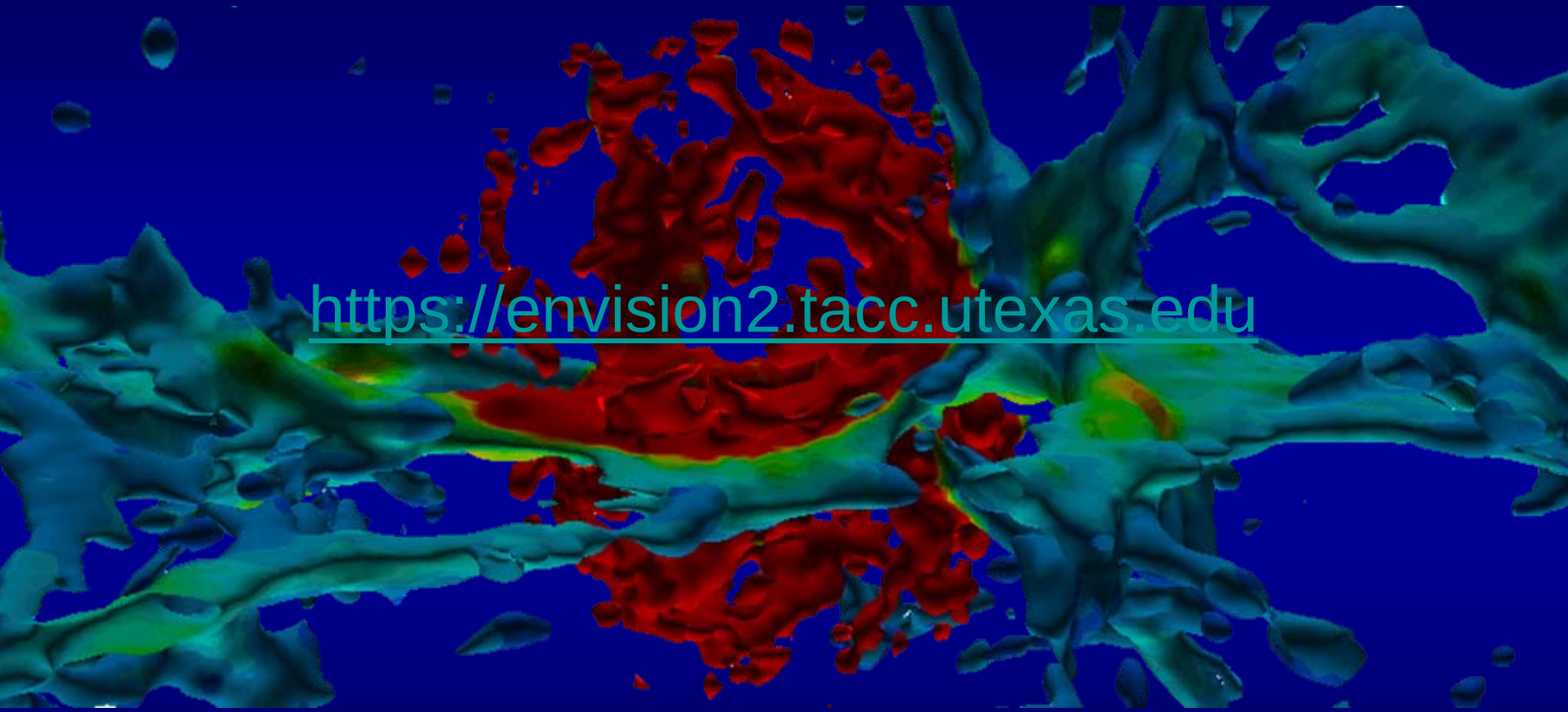
EnVision General Architecture



Current State and Future Plans

- Version 2.0b1 released June 9, 2008
- Current capabilities:
  - TACC Maverick visualization resource
  - supports any VTK formatted data set
  - structured and point arbitrarily formatted data importation through interview
  - basic visualization methods
- Future capabilities include:
  - additional visualization resources
  - additional visualization methods (e.g. volume rendering)
  - collaborative visualization sessions
  - unstructured arbitrarily formatted data importation through interview
  - automated data staging

EnVision Demo



https://envision2.tacc.utexas.edu

EnVision Demo

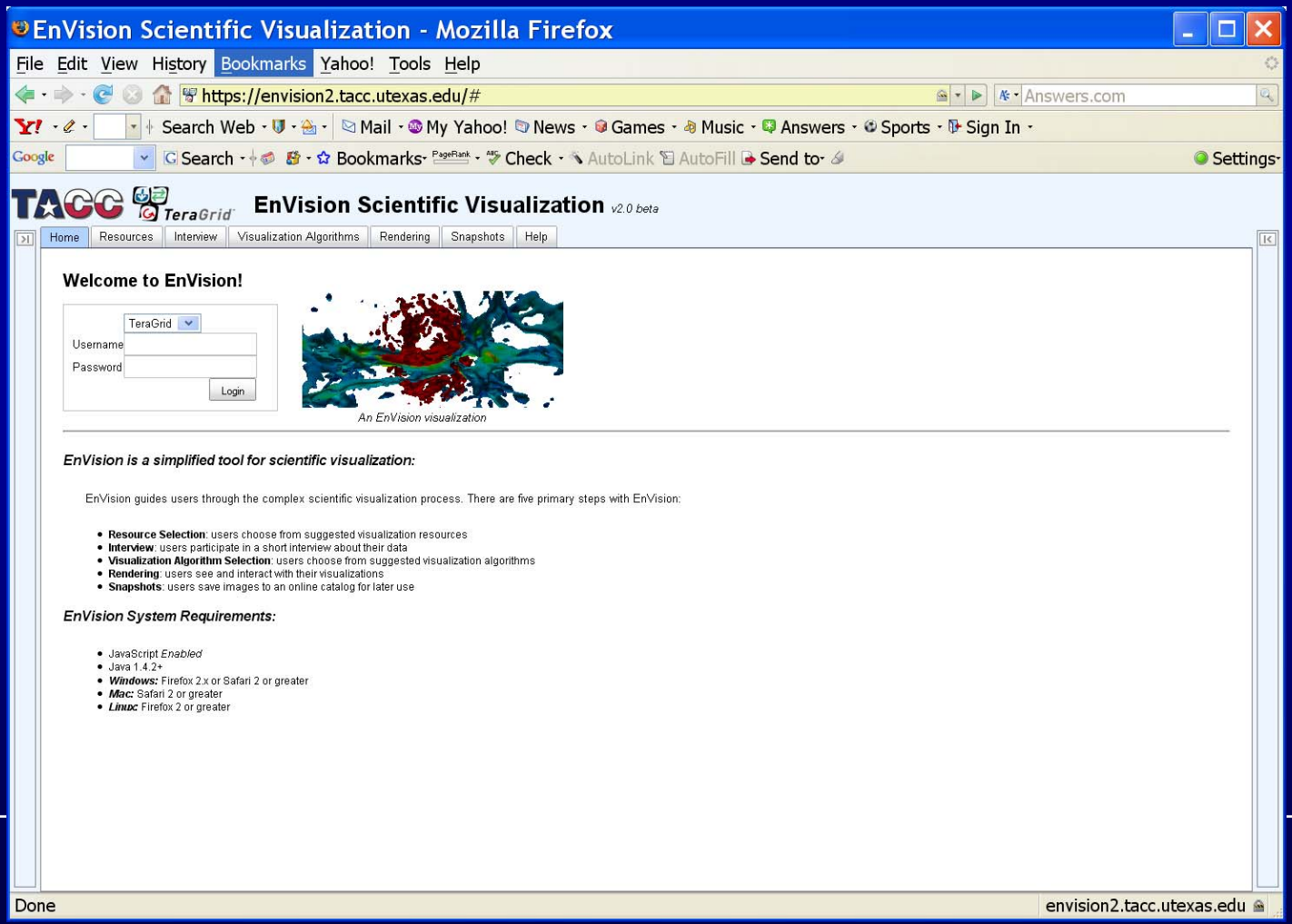
Step 1: Login in to your maverick account and type the following command:

cp –R ~train00/envision_demo ./

Then log out of maverick

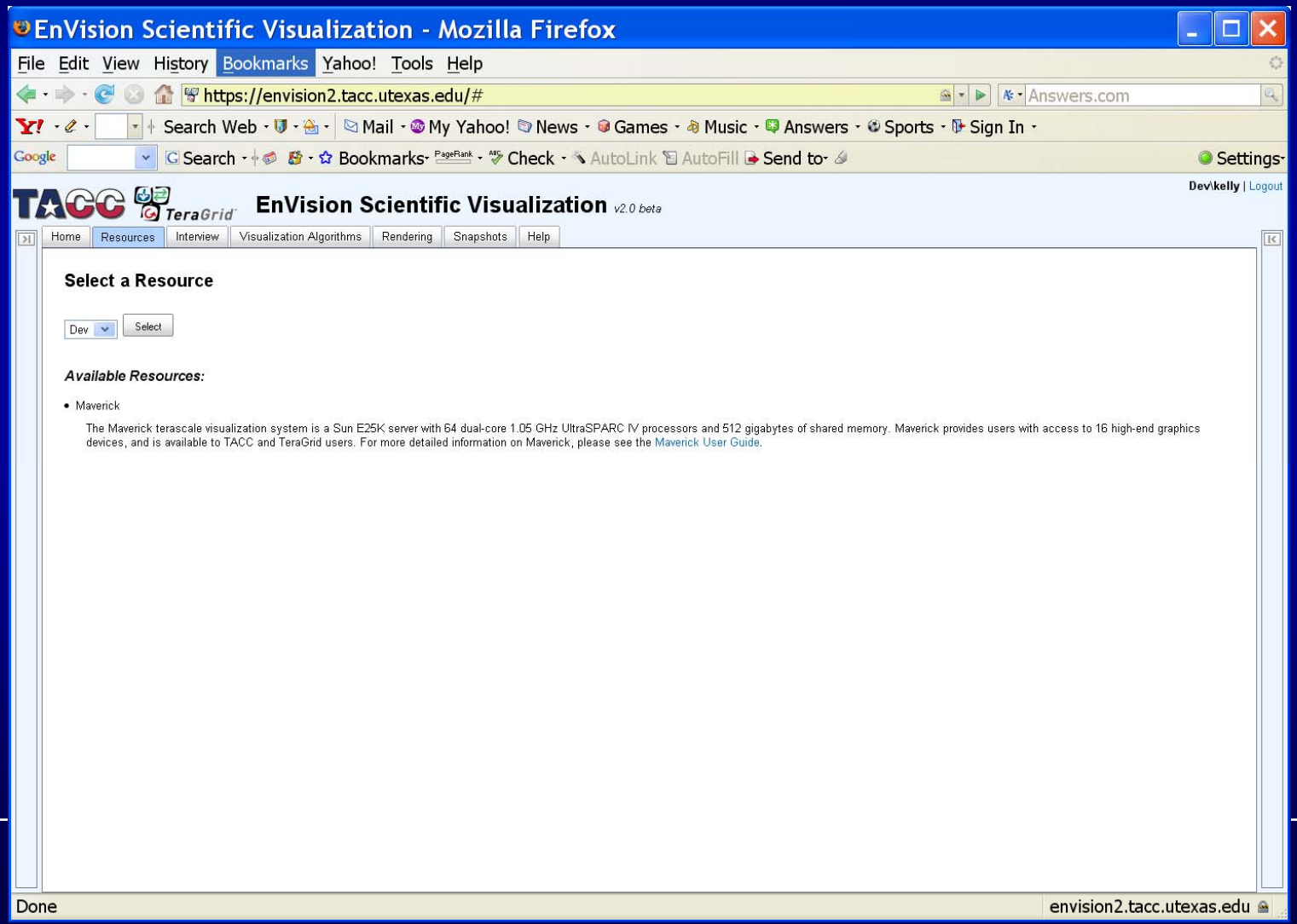
EnVision Demo

Step 2: Login and authenticate: goto http://envision2.tacc.utexas.edu



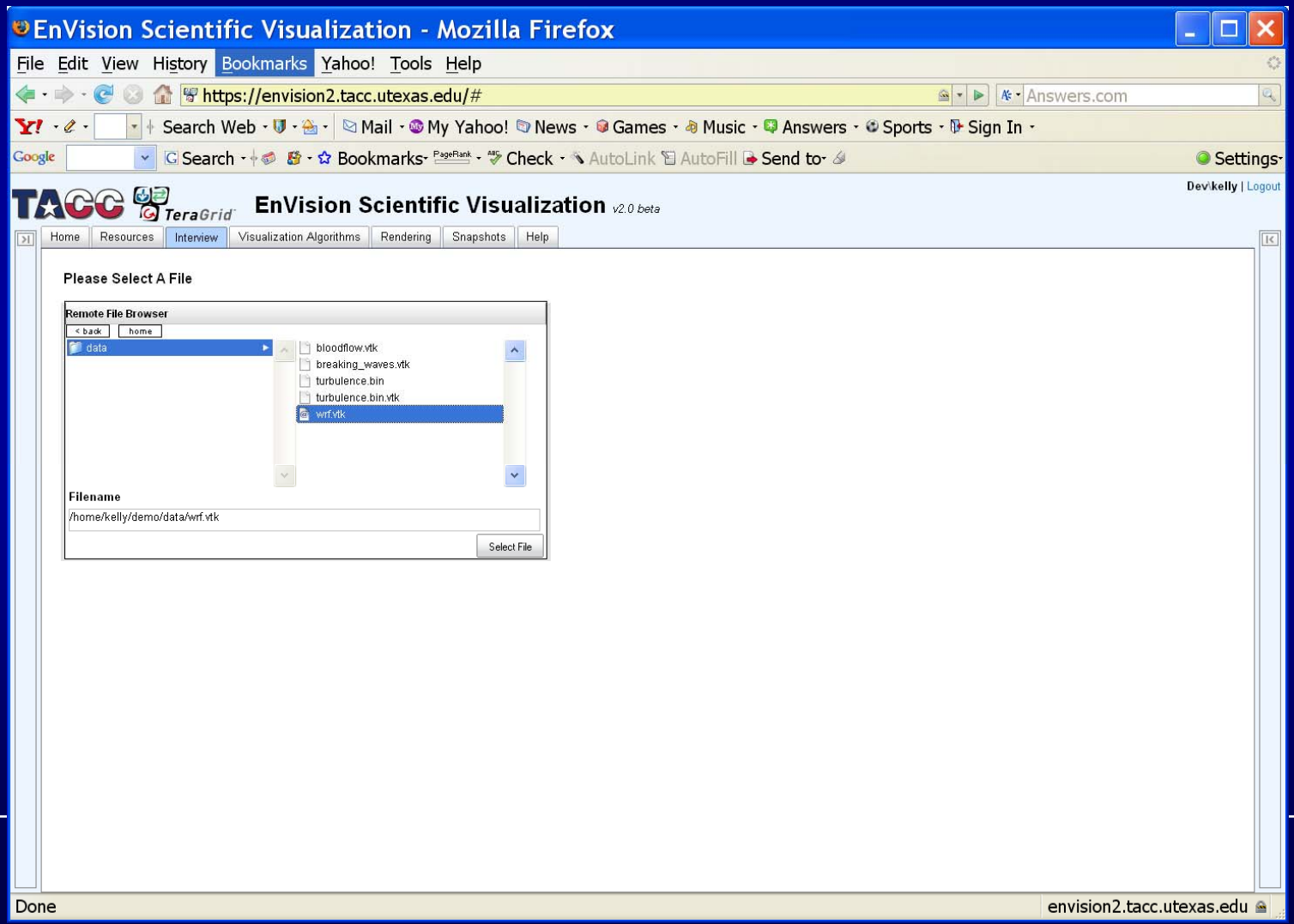
EnVision Demo

Step 3: Select the resource. (Maverick)



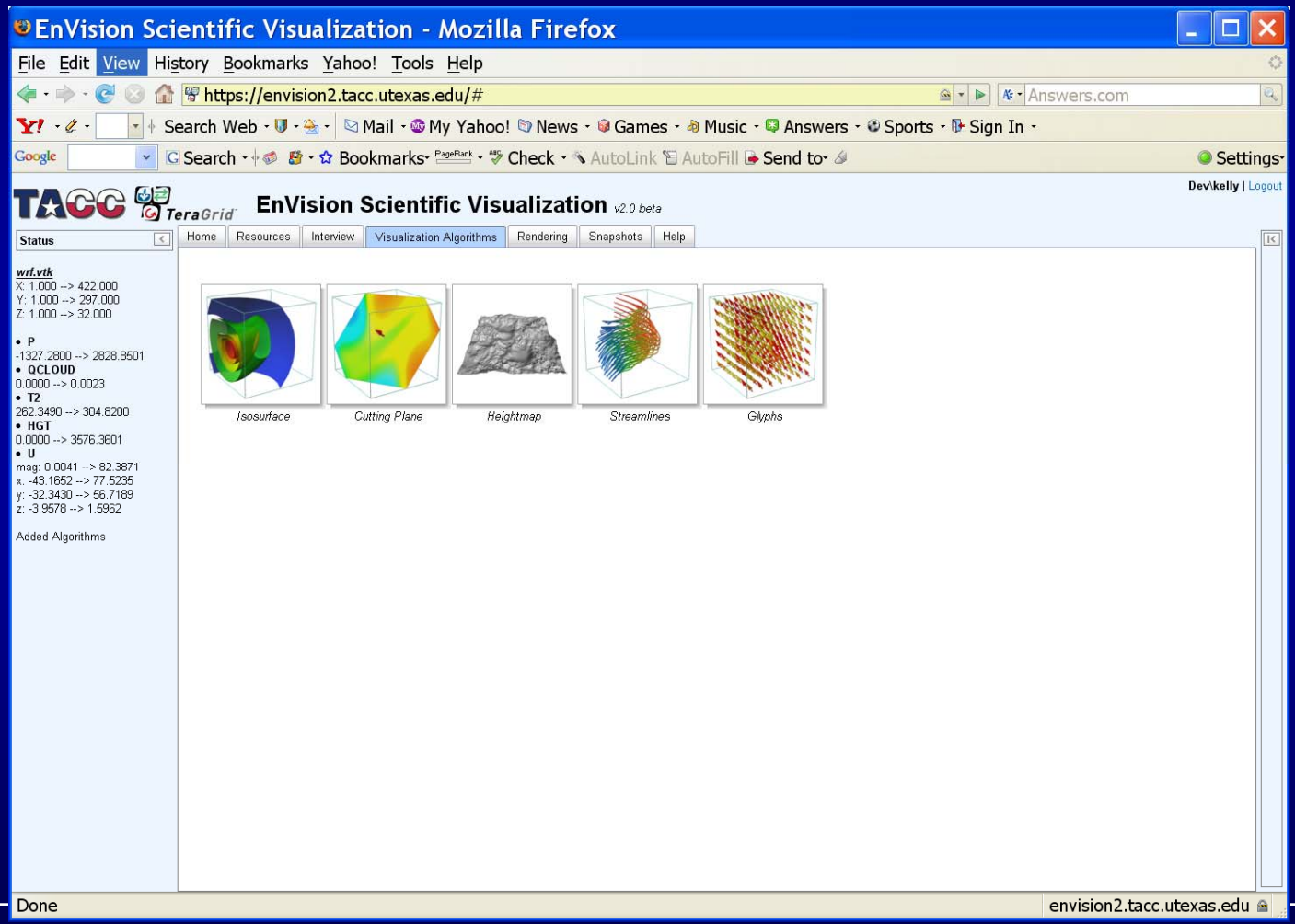
EnVision Demo

Step 4: Load the data.



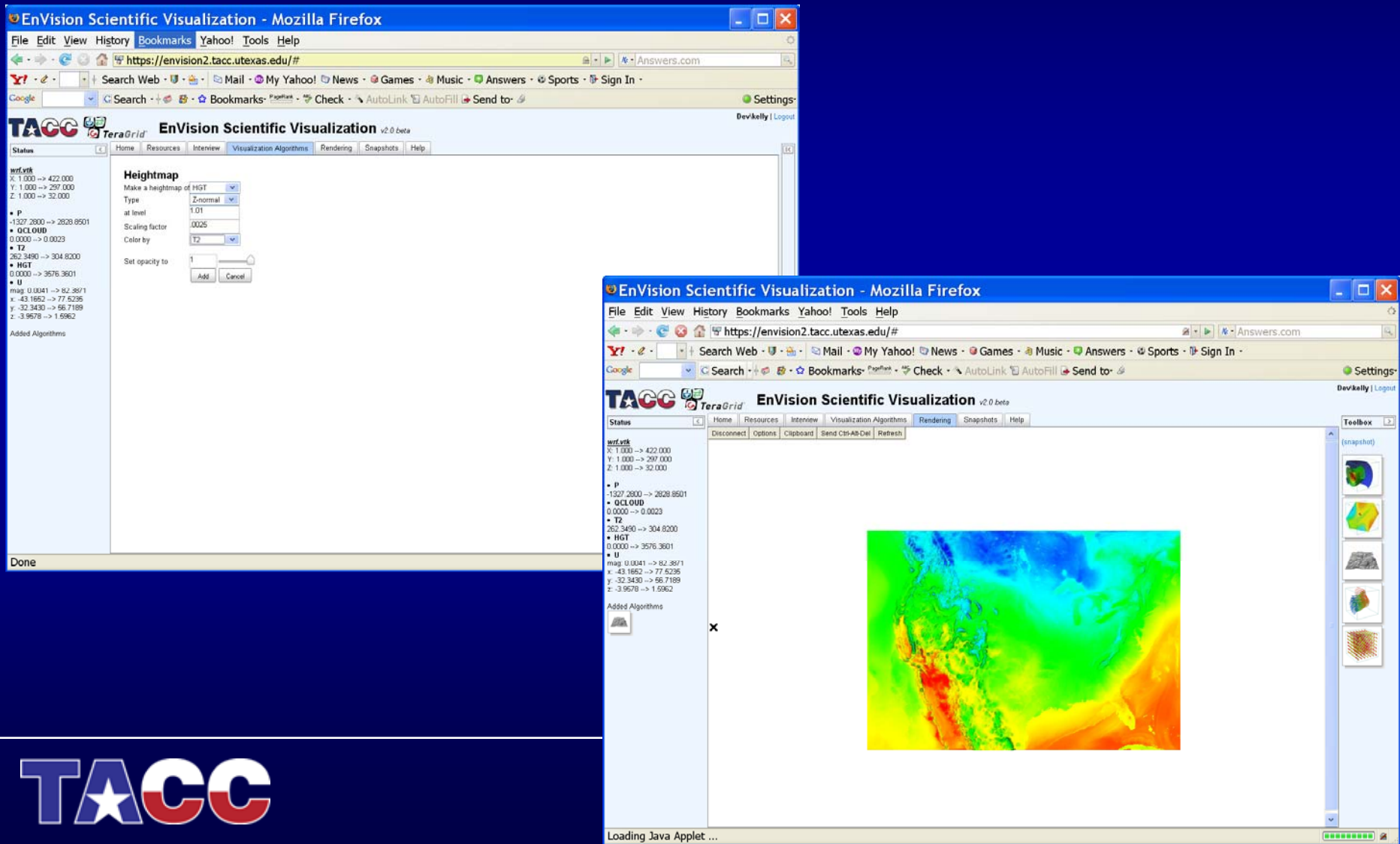
EnVision Demo

Step 5: Select the type of visualization you want to see.



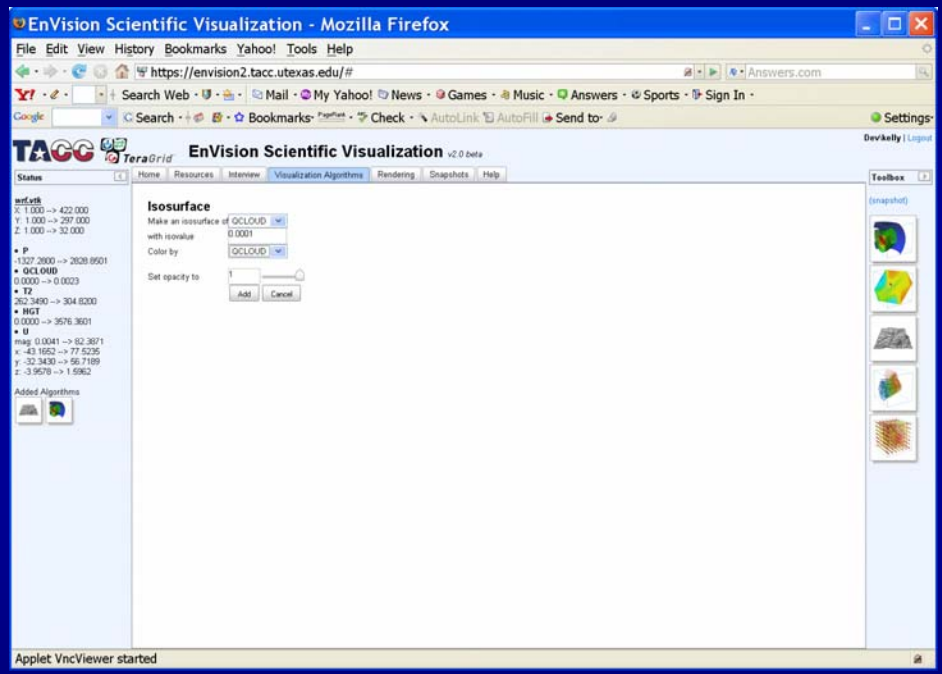
EnVision Demo

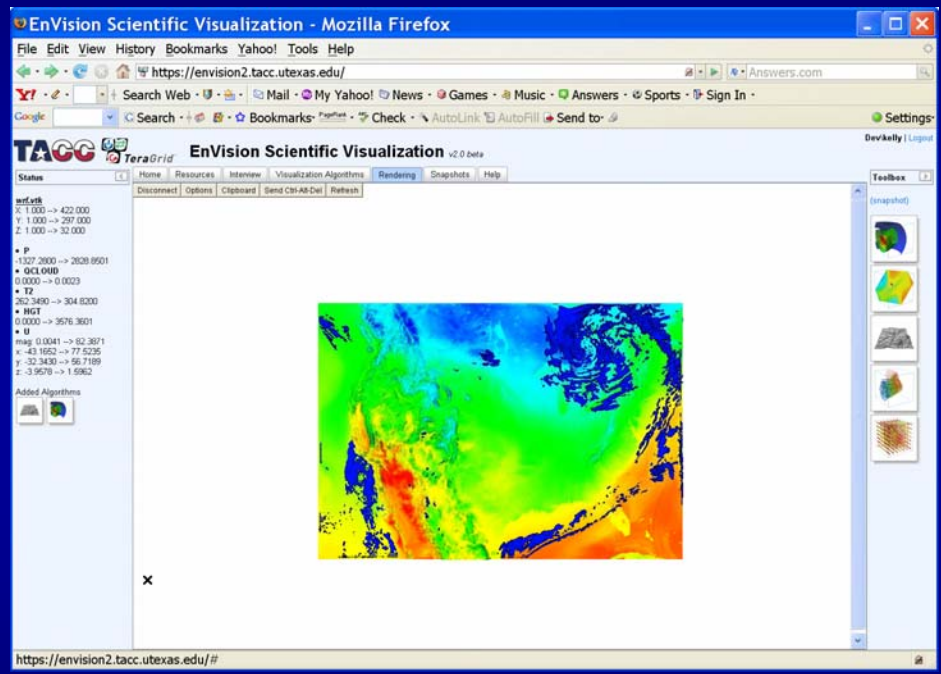
Step 6: Create a heightmap.



EnVision Demo

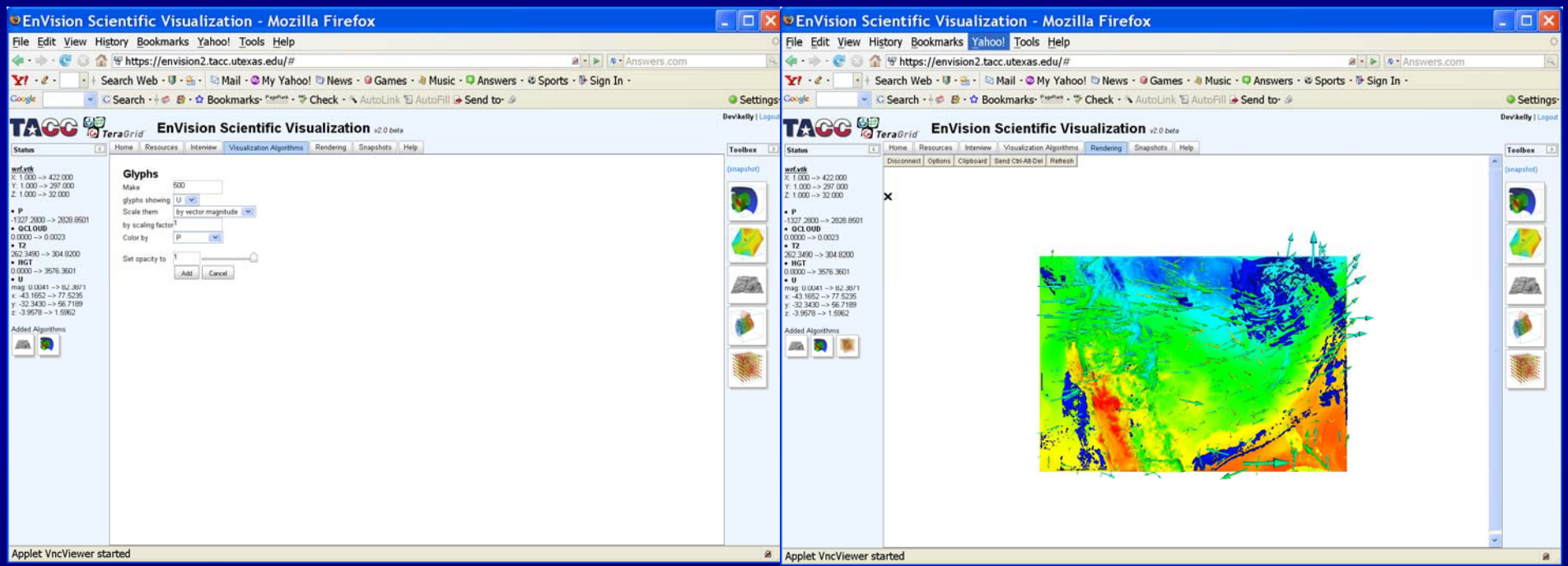
Step 7: Create an isosurface.





EnVision Demo

Step 8: Create glyphs.



Topics Covered

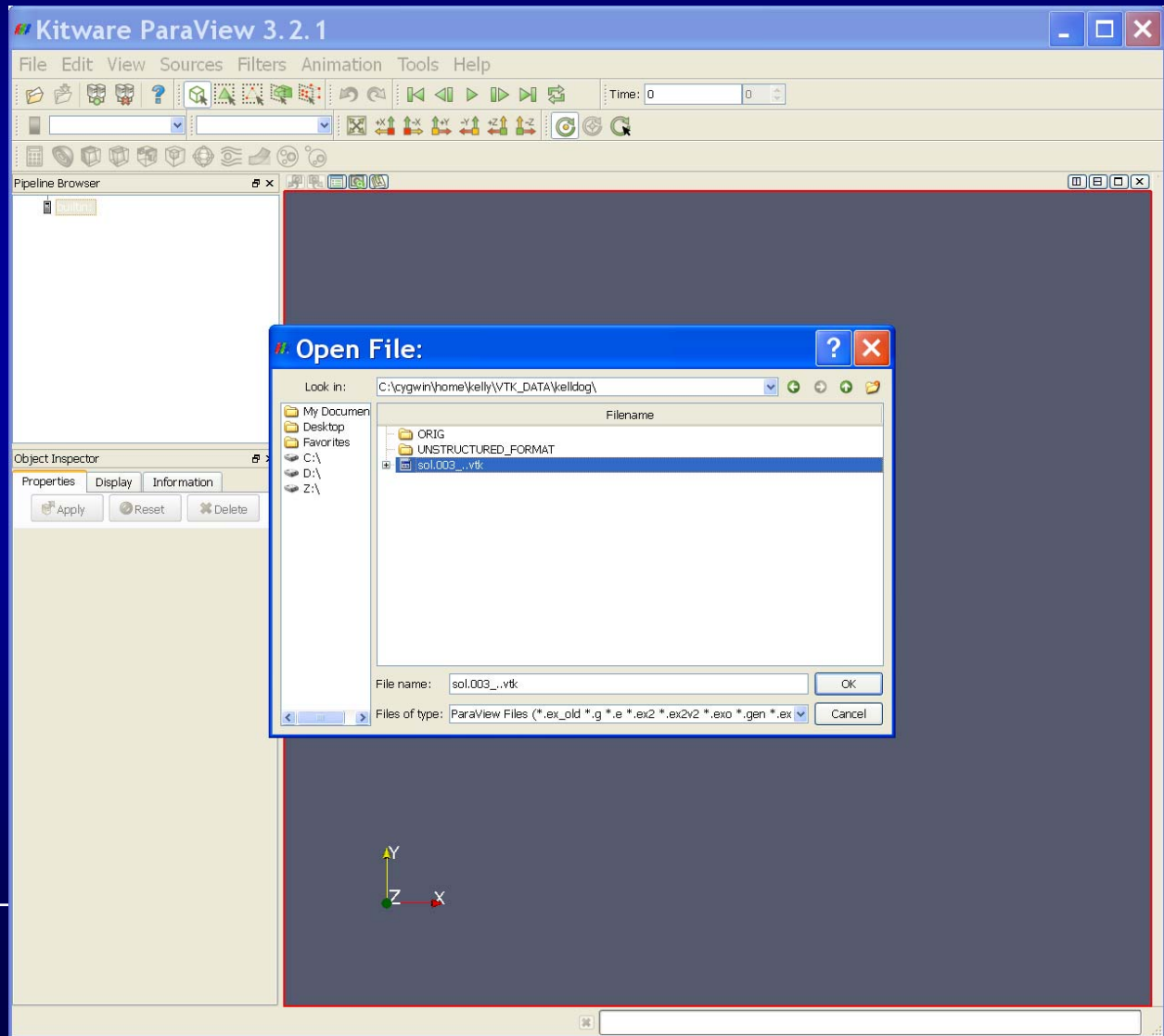
- Remote and Collaborative Visualization
- EnVision – simplified Interface for Visualization
- Visualizing and Animating a Time Series in Paraview
- VisIt
- Parallel Visualization

Visualizing and Animating a Time Varying Dataset

- Download the data from your maverick account at /work/utexas/staff/kelly/PORTUGAL/
- The data is called sol_data.tar.gz
- gunzip the file: gunzip sol_data.tar.gz
- Untar the file: tar –xvf sol_data.tar
- This will produce a series of time steps

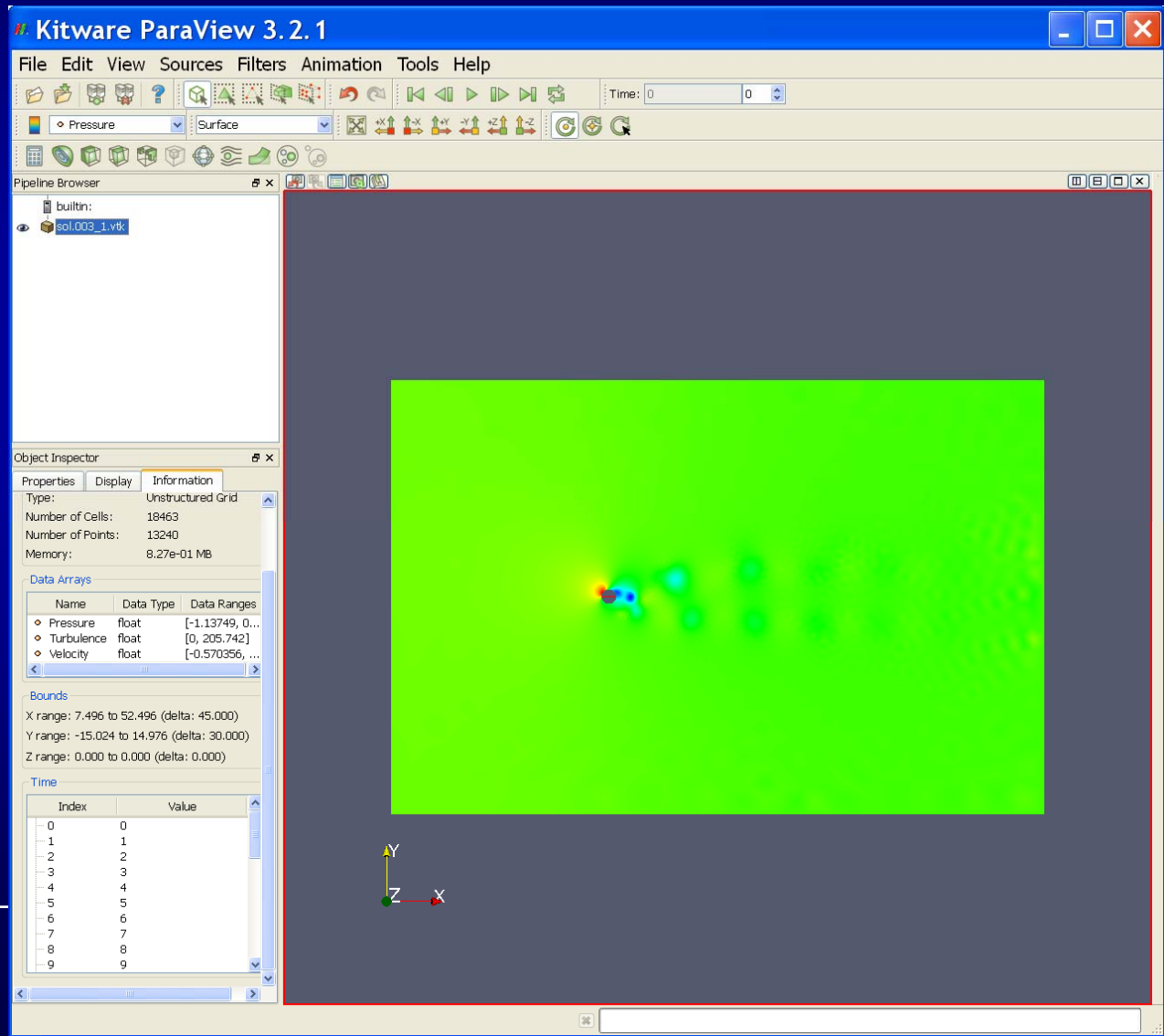
Visualizing and Animating a Time Varying Dataset

- Step 1: In paraview, locate the time series data, select it and load the data.



Visualizing and Animating a Time Varying Dataset

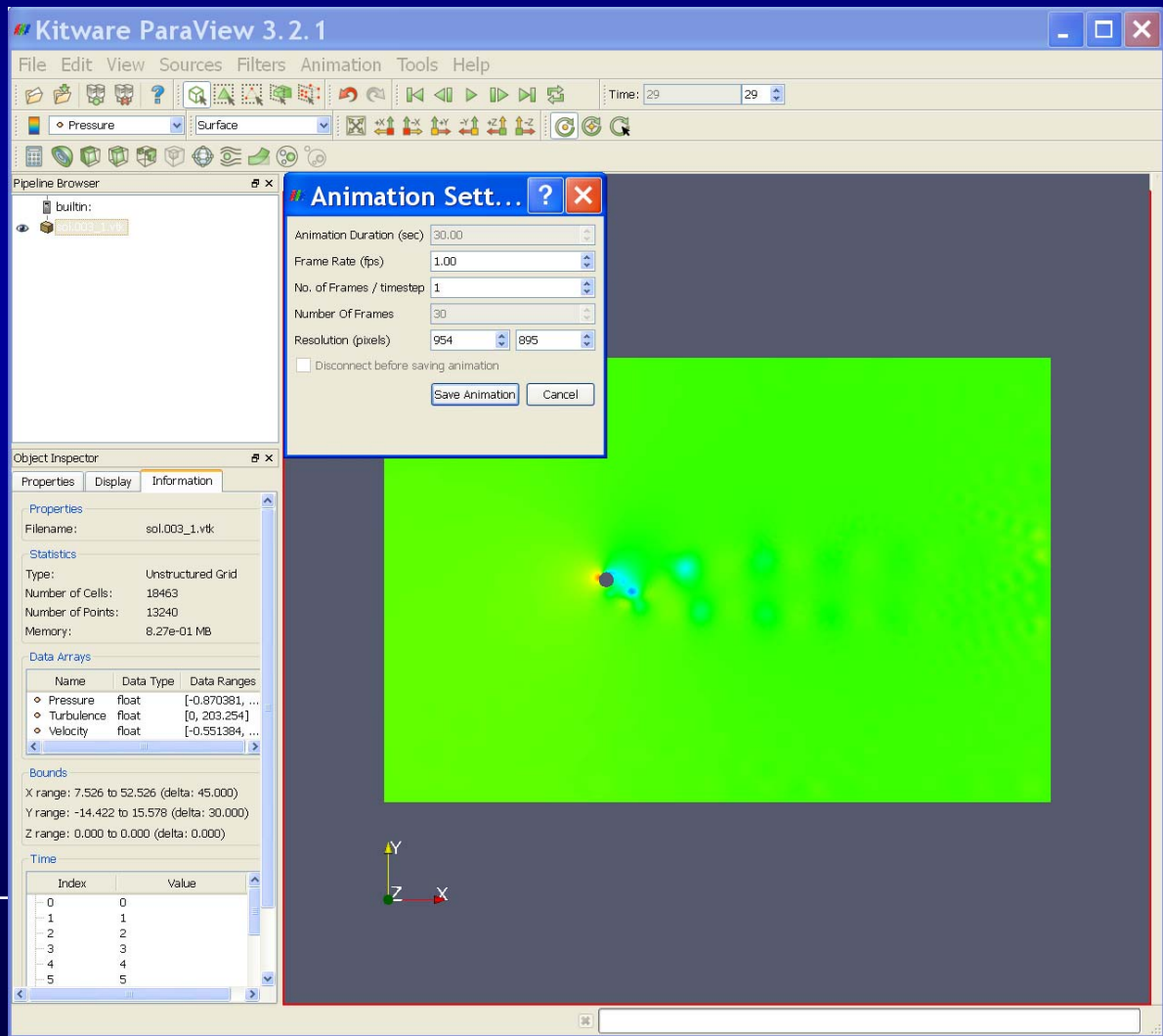
- Step2: Find the data's information and verify that all time steps were loaded.



TACC

Visualizing and Animating a Time Varying Dataset

- Step 3: In the File menu, select “Save Animation”. This will animate the time series.

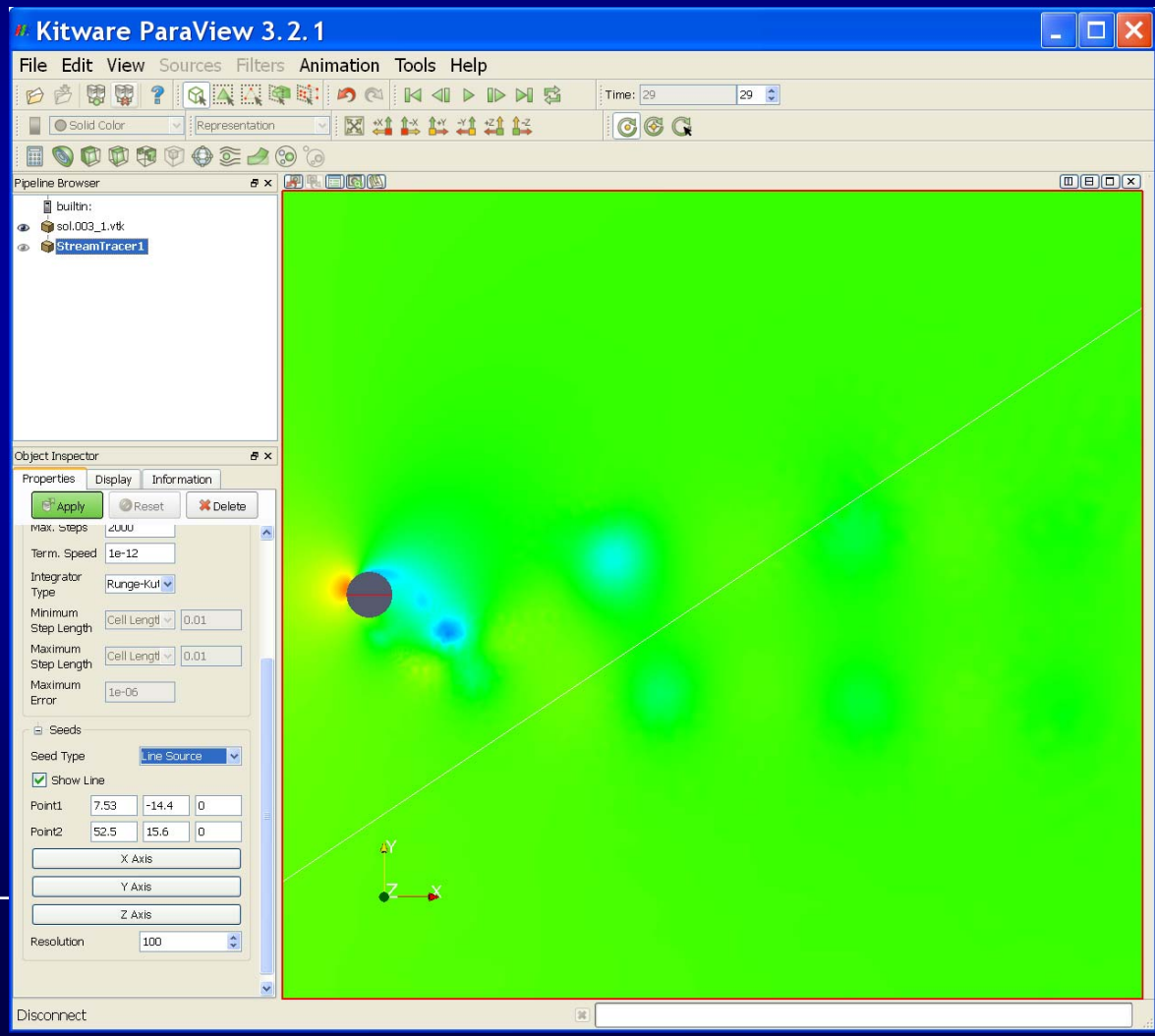


Visualizing and Animating a Time Varying Dataset

- The movie should look like:

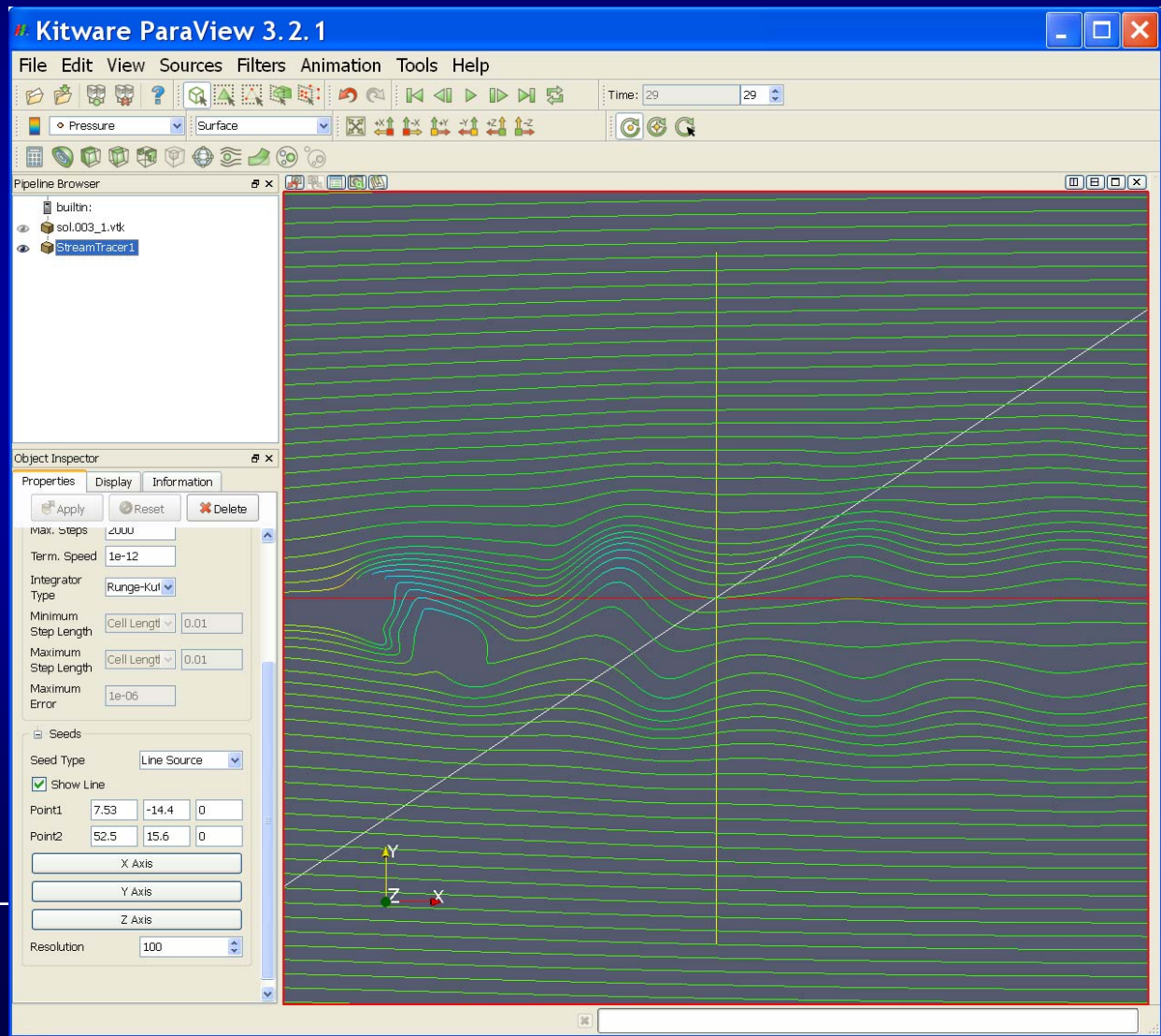
Visualizing and Animating a Time Varying Dataset

- Step 4: Add particle traces using a line to seed the points. Change the color to black and line width to 1.



Visualizing and Animating a Time Varying Dataset

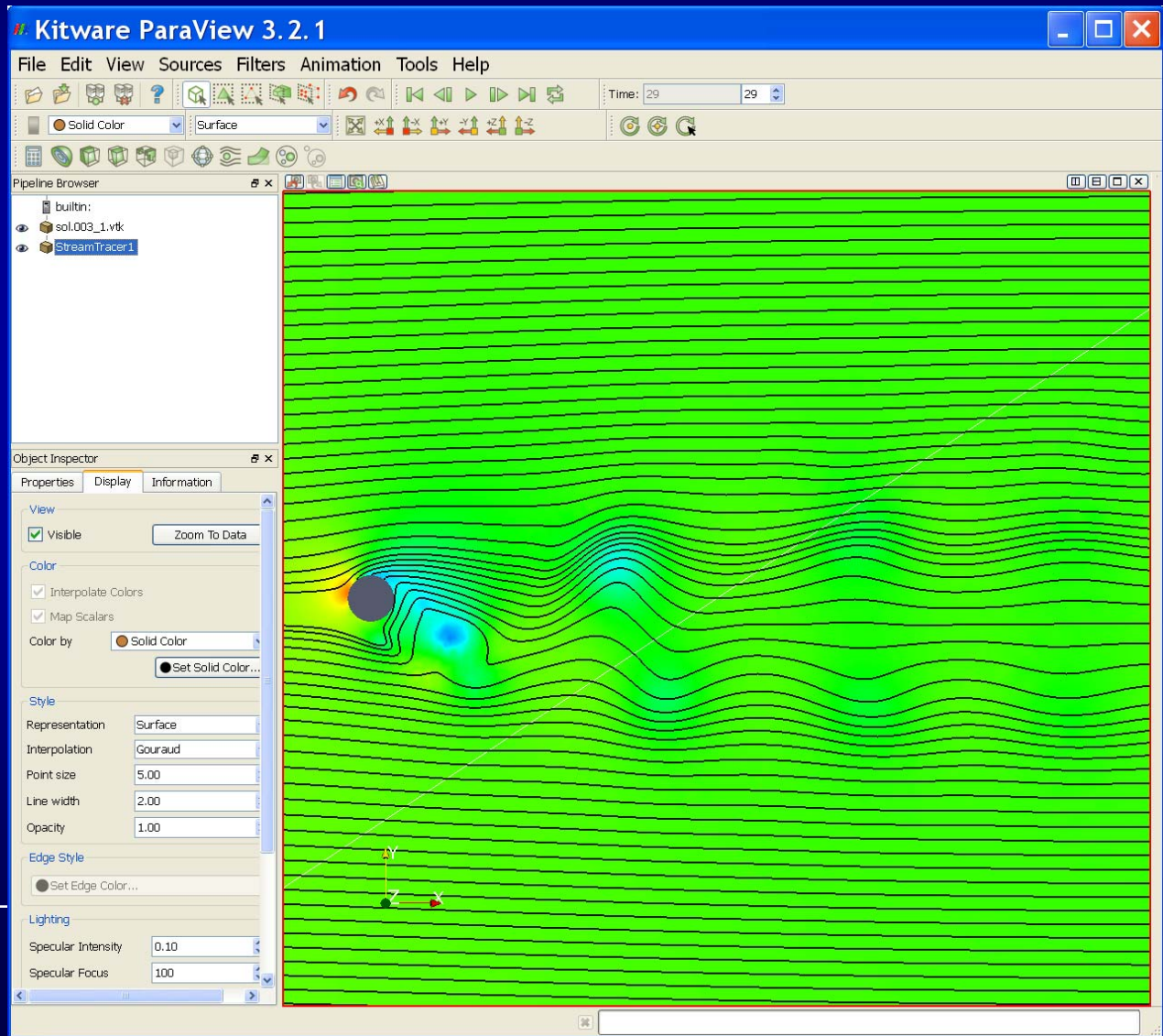
- Step 5: Go back to the contour and turn it back on.



TACC

Visualizing and Animating a Time Varying Dataset

- Step 6: Save an animation of this view.



Visualizing and Animating a Time Varying Dataset

- The movie should look like:

Topics Covered

- Remote and Collaborative Visualization
- EnVision – simplified Interface for Visualization
- Visualizing and Animating a Time Series in Paraview
- VisIt
- Parallel Visualization

VisIt

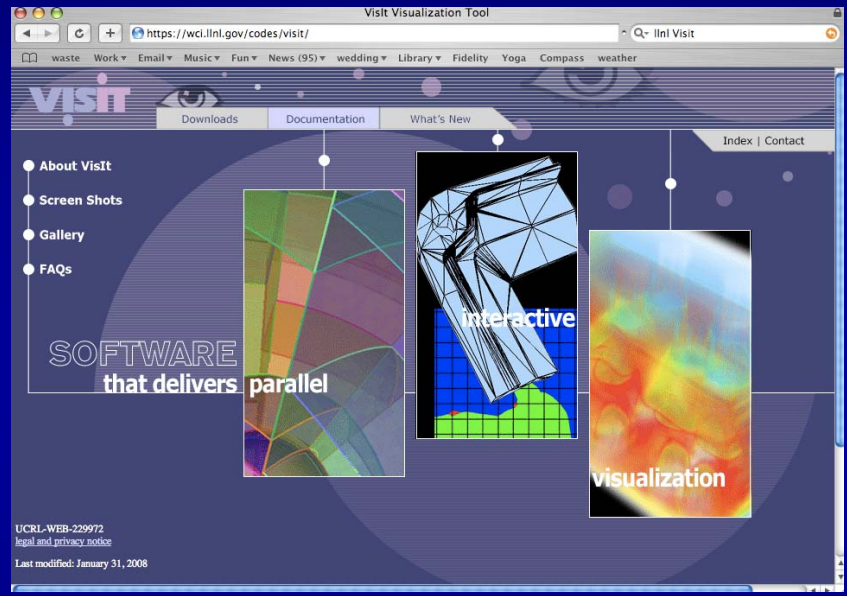
- Open source parallel visualization tool out of LLNL
- Key features:
  - Scalar, vector, and tensor field visualization
  - Structured and unstructured mesh types
  - Support for structured AMR meshes and CSG meshes\
  - Quantitative analysis (expressions, queries, picking, lineout)
  - GIS support
  - Image export (*BMP, JPEG, PNG, PPM, Raster Postscript, RGB, TIFF, stereo images*)
  - Annotation support for publication and presentation graphics
  - Built on VTK (open source graphics library)

VisIt Supported File Formats

ANALYZE
ANSYS
BOV (brick of values)
Boxlib
CGNS
Chombo
CTRL
Curve2D
Ensight Gold
Enzo
Exodus
FITS
FLASH
FLUENT
FVCOM
GGCM
H5Nimrod
H5Part
Image
ITAPS
MFIX
MM5
NASTRAN
Nek3D
NetCDF
OpenFOAM
PATRAN
PLOT3D
Protein Databank
SAMRAI
Silo
Spheral
STL
TecPlot
VASP
Vis5D
VTK
Wavefront OBJ
Xmdv
ZeusMP (HDF4)
GIS (DEM, ArcGrid, ESRI Shapefile, vector file formats…)

Running VisIt

- Download from web and install on local computer



VisIt Application

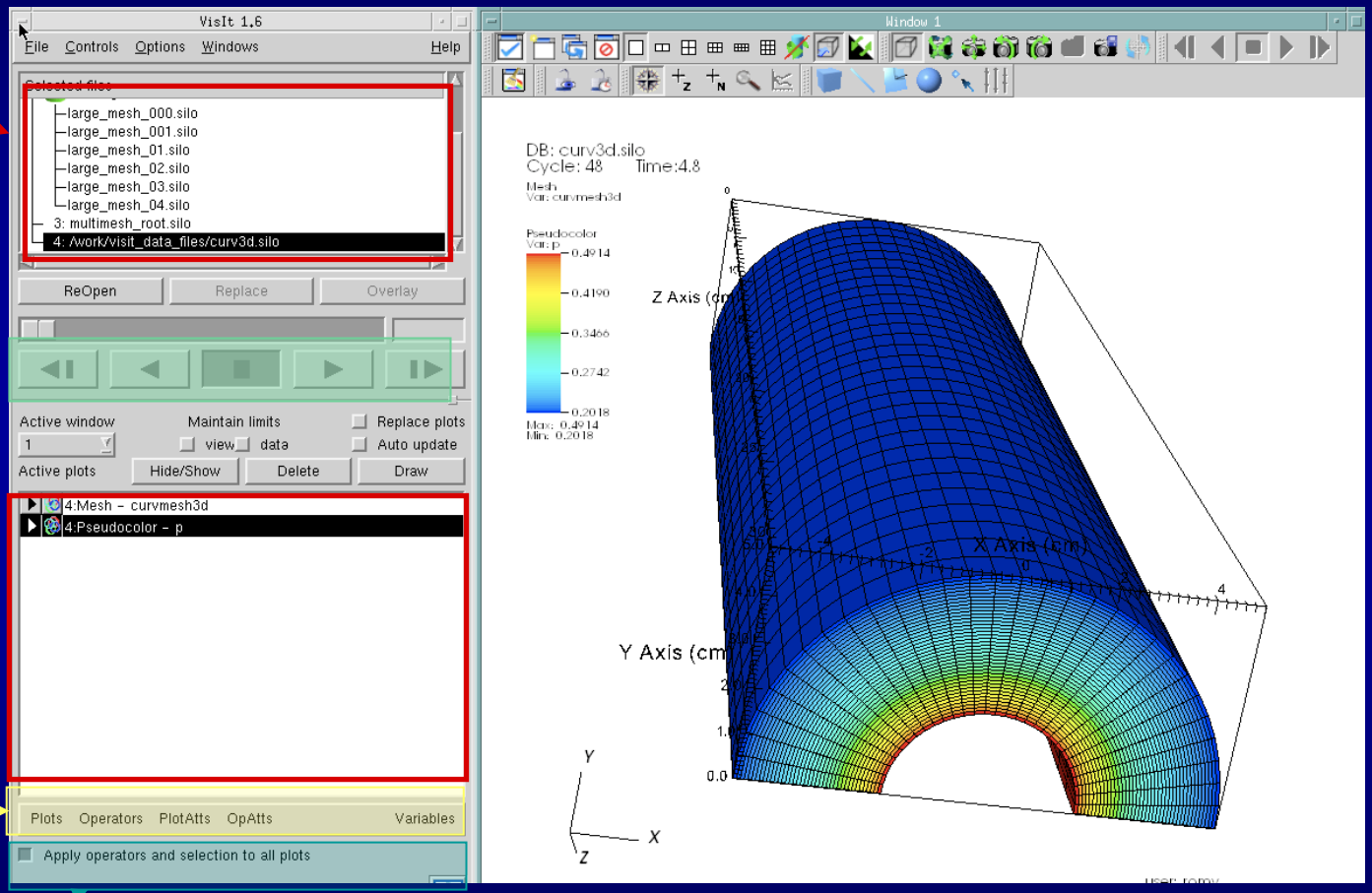
Database Viewer (File List)

Animation Controls

Plot List

Plot and Operator Menus

Status Bar



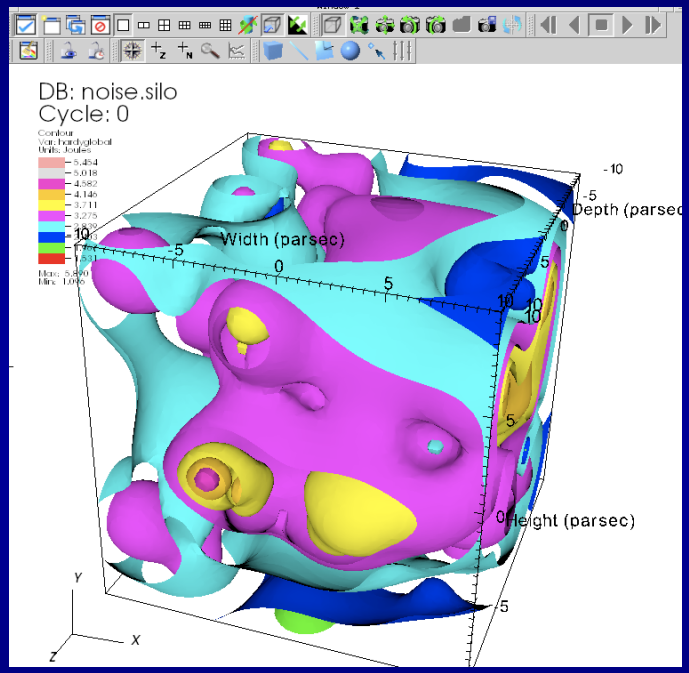
Viewer (Output Window)

Visit Terminology

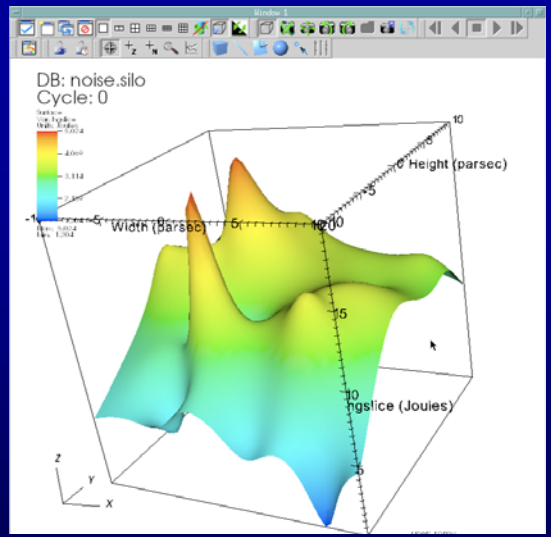
- Plot = Mapping Algorithm
  - Pseudocolor plot (scalar color map)
  - Surface plot (of 3D isosurface or 2D data)
  - Volume plot (3D volume rendered image)
- Operator = Data Manipulation Algorithm
  - Slice (extract data)
  - Resample (change data resolution)
  - Transform (move data in space or time)
- Database = file or set of files

Common Plots

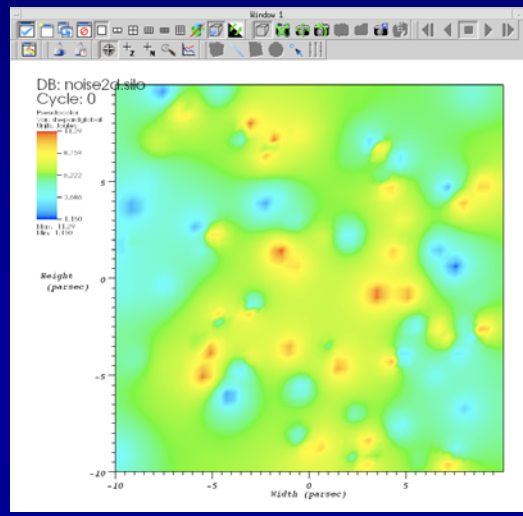
Contour



Surface

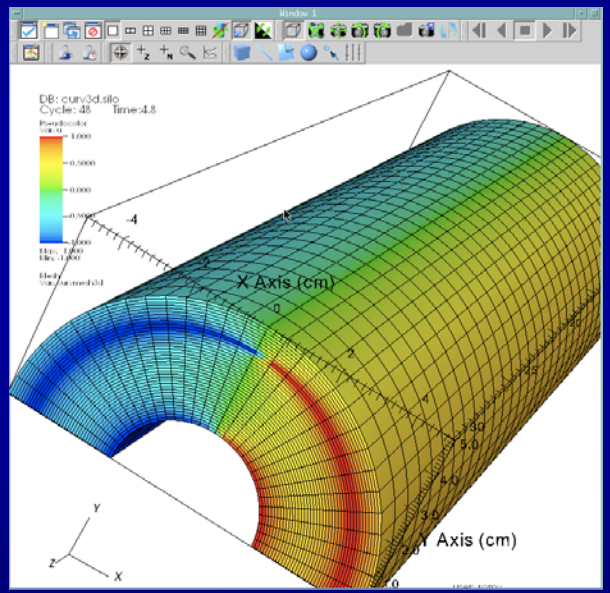


Pseudocolor

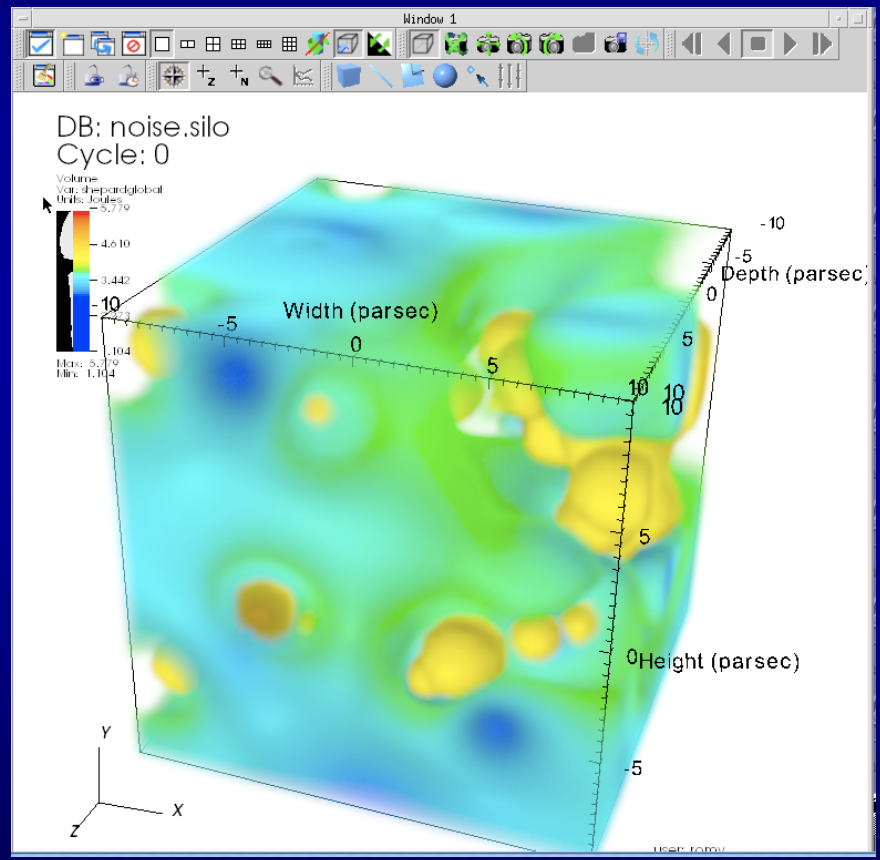


Common Plots

Mesh + Pseudocolor



Volume

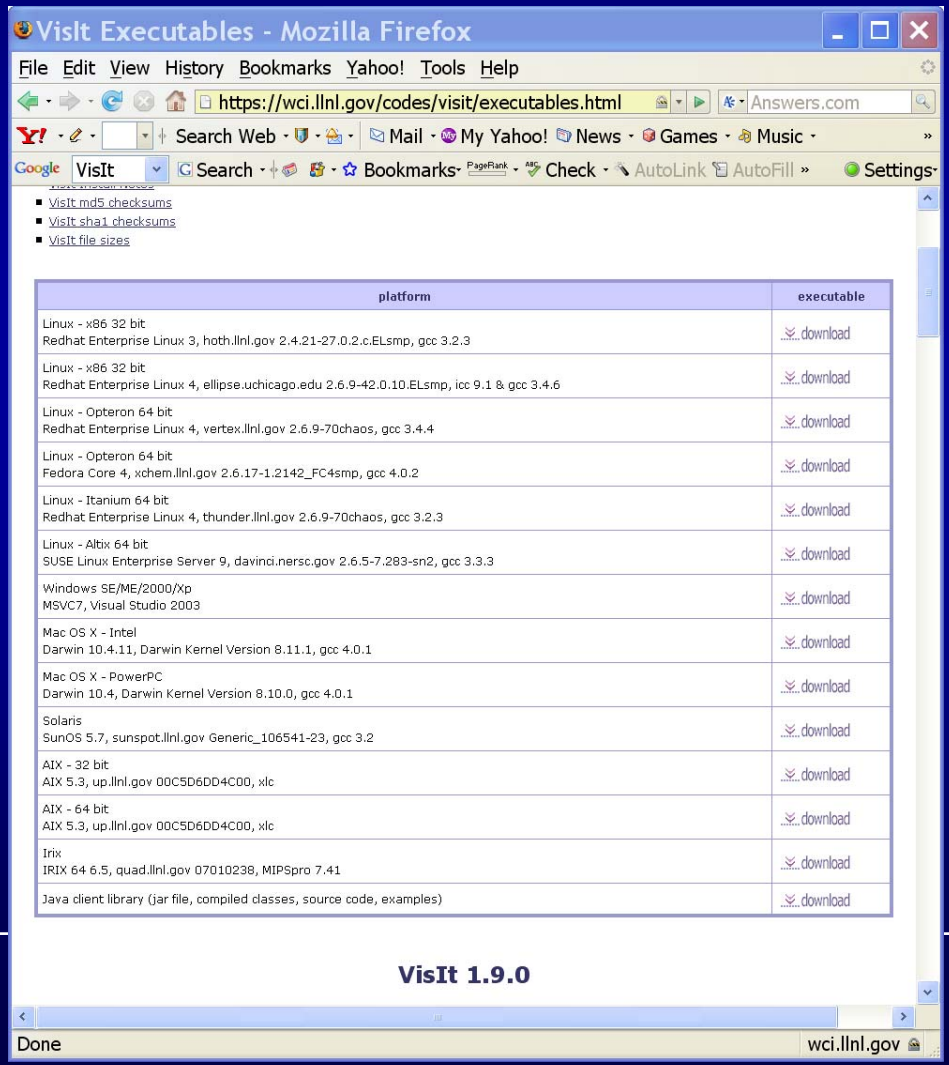


VisIt Advanced Features

- Various lighting, rendering, and view (camera) options
- Geometry export (*Curve, Alias WaveFront Obj, STL, ULTRA,* and *VTK)*
- Animation and movie generation
- Scripting interface with Python
- API interface with C++ and Java
- Dynamically extensible through plugins
- Parallel and distributed architecture for visualizing very large data sets
- Database export (BOV, SILO, Tecplot, SimV1Writer, VTK, columnar ascii file (xmdf))
- Multiple database correlation / visualization

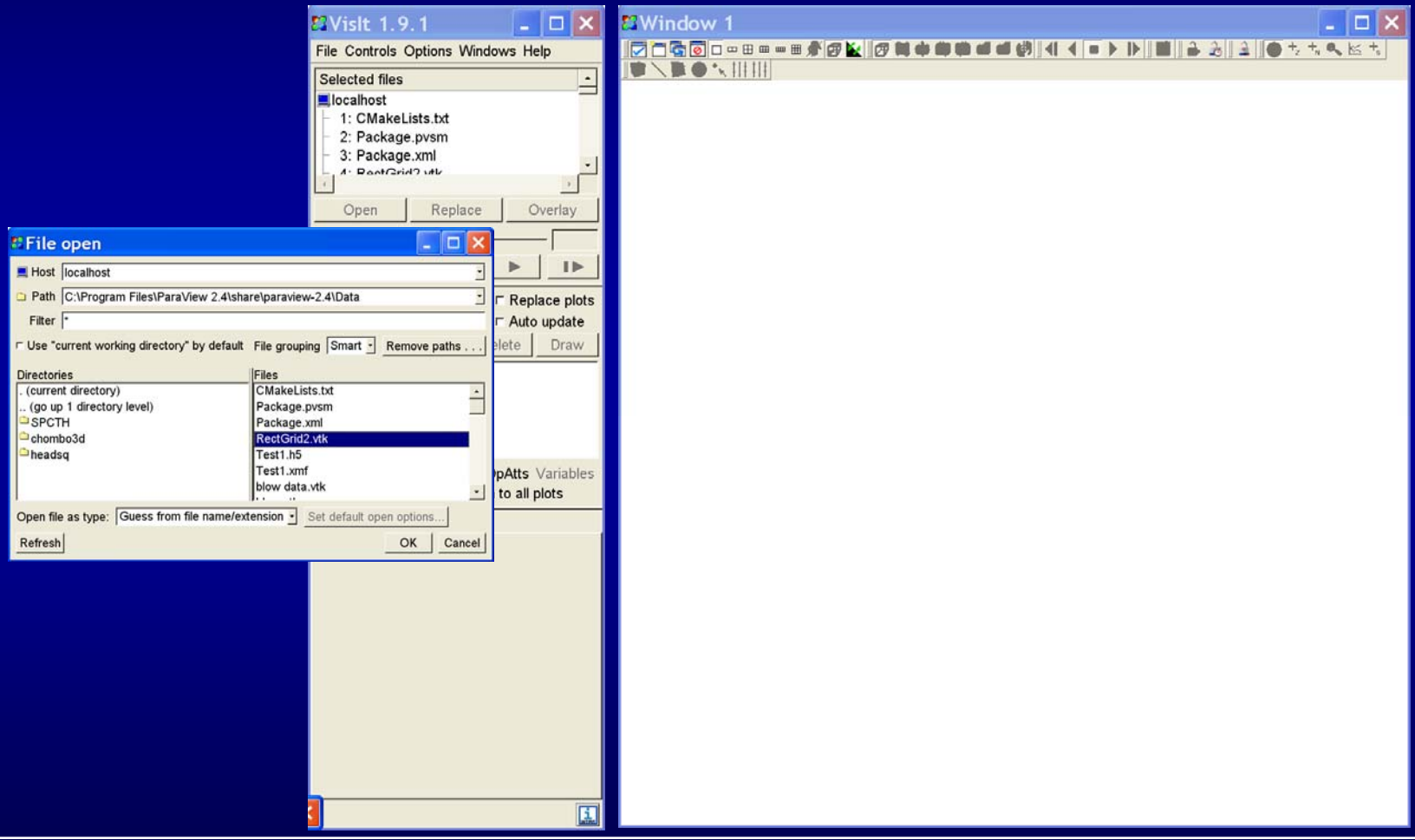
Download VisIt and Go Through Example

- Step 1: Download VisIt from
  - https://wci.llnl.gov/codes/visit/executables.html



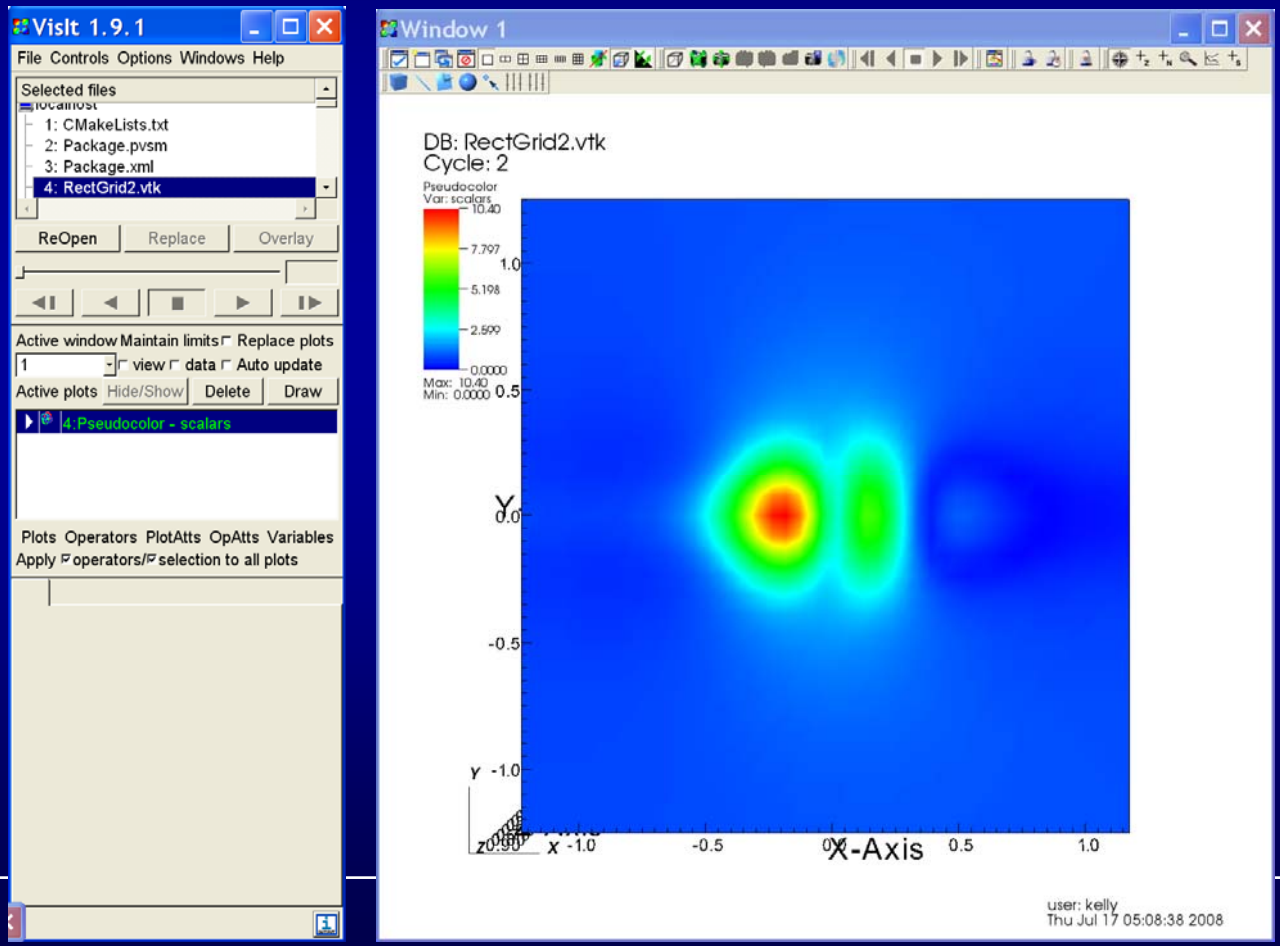
Download VisIt and Go Through Example

- Step 2: Bring up VisIt and Load the RectGrid2.vtk dataset



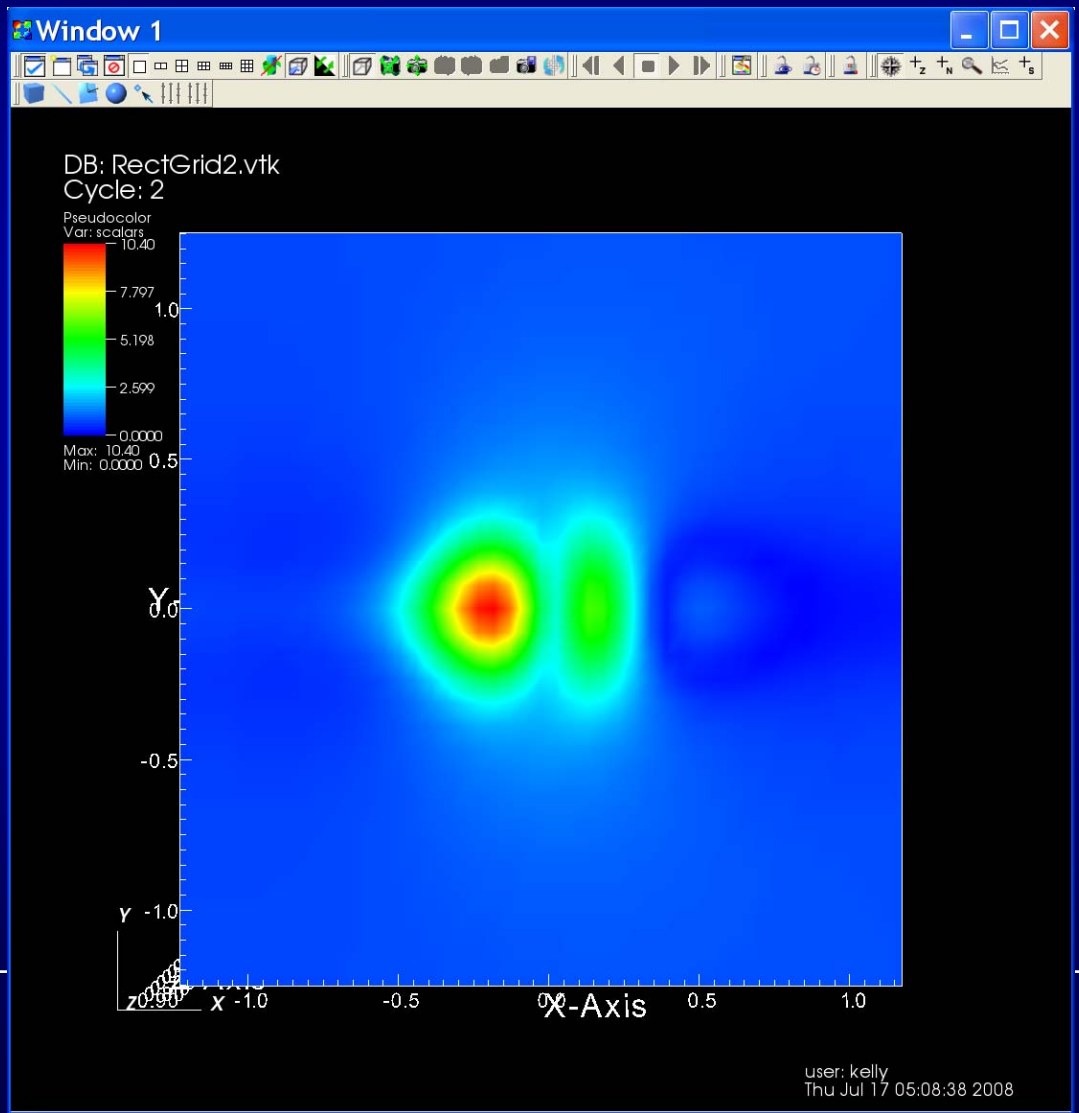
Download VisIt and Go Through Example

- Step 3: Create a surface plot (pseudocolor) of the volume boundary by clicking on plots and selecting pseudocolor.



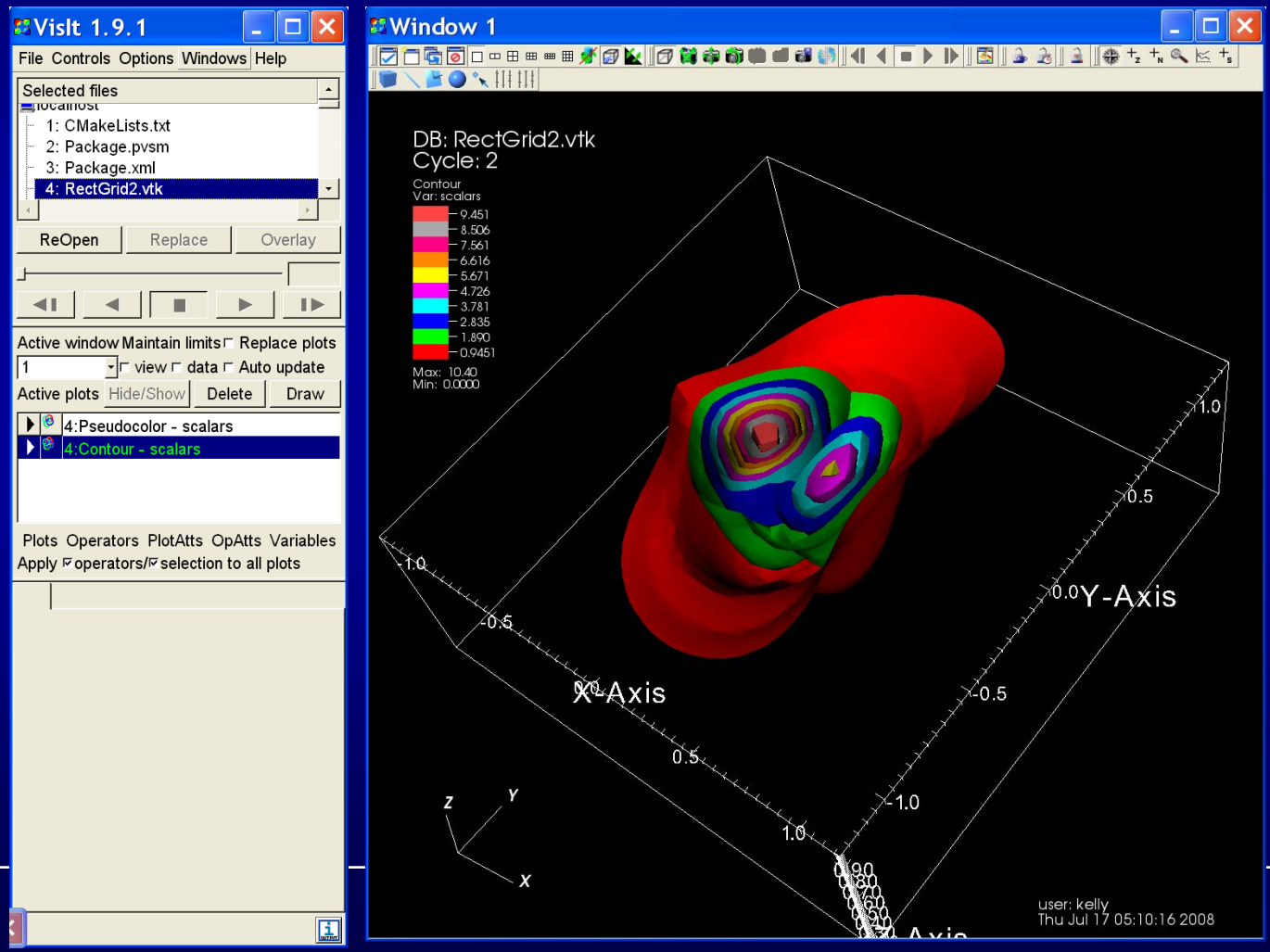
Download VisIt and Go Through Example

- Step 4: Swap the background color and foreground color



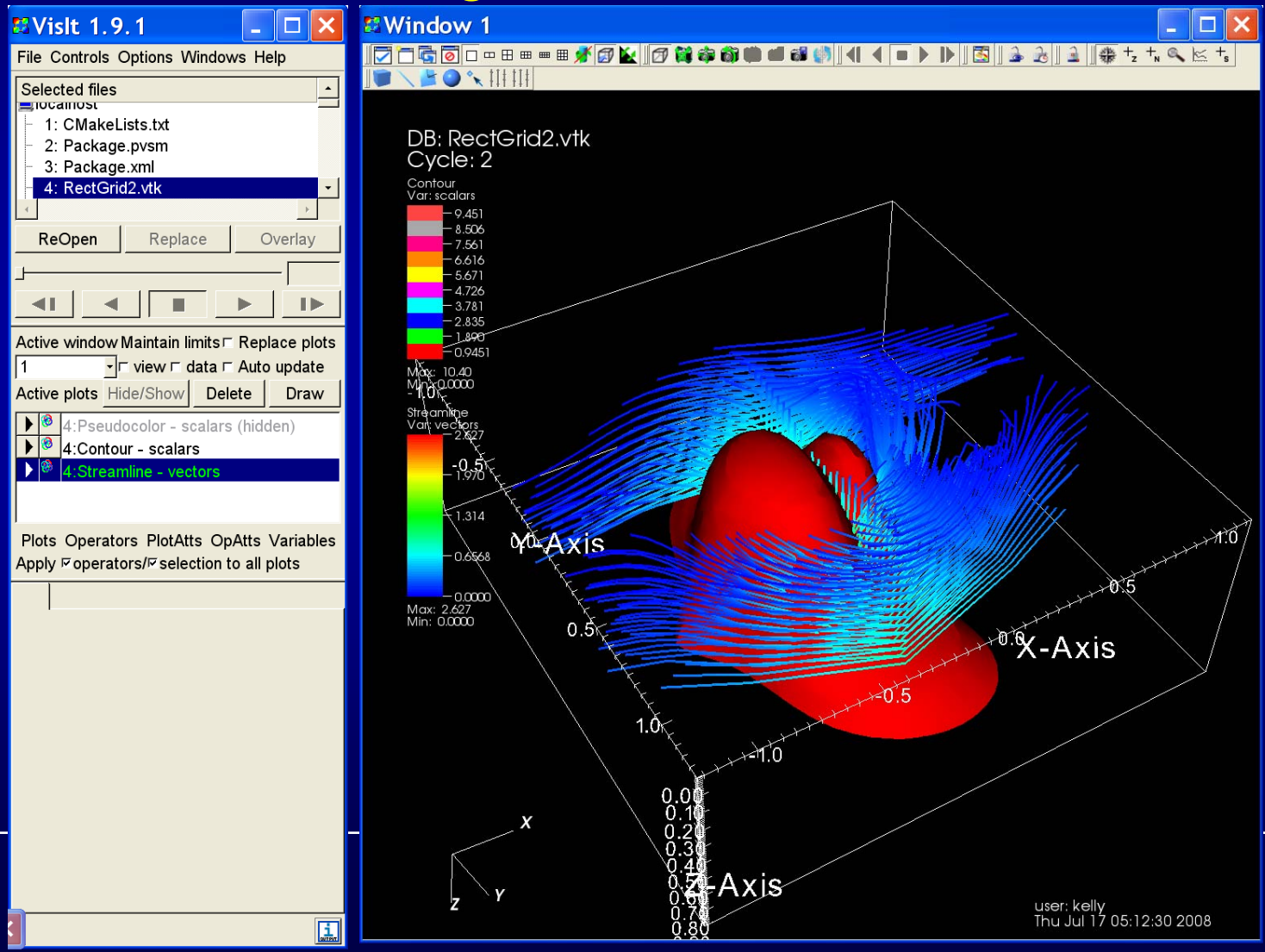
Download VisIt and Go Through Example

- Step 5: Create a set of isosurfaces by clicking on plots and selecting contours.



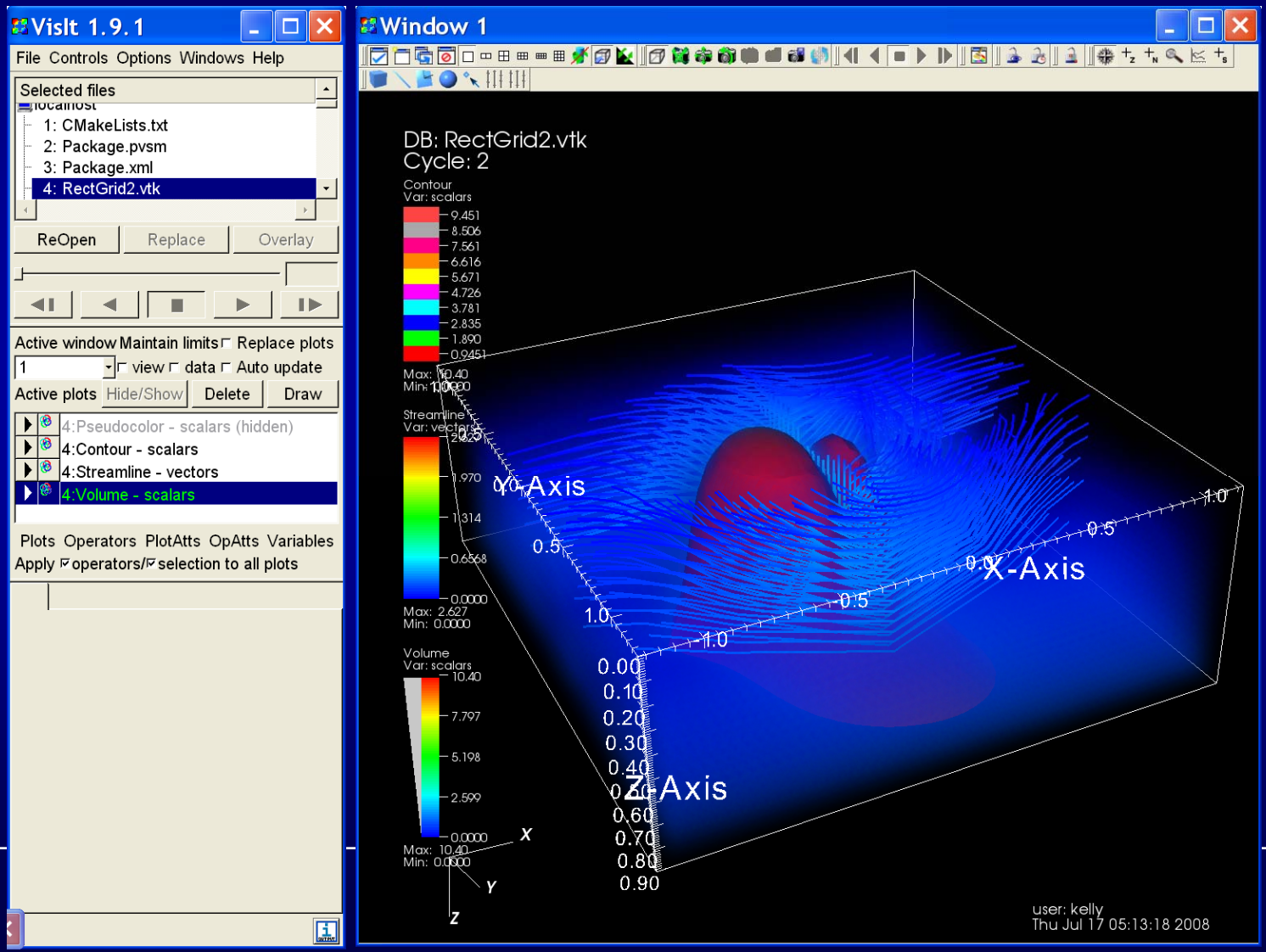
Download VisIt and Go Through Example

- Step 6: Create a set of streamlines by clicking on plots and selecting streamline



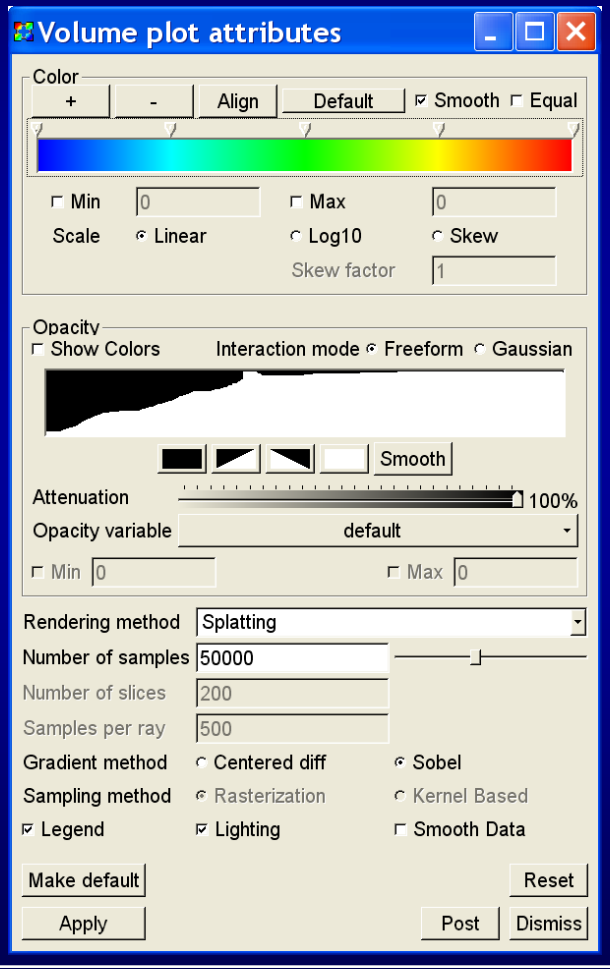
Download VisIt and Go Through Example

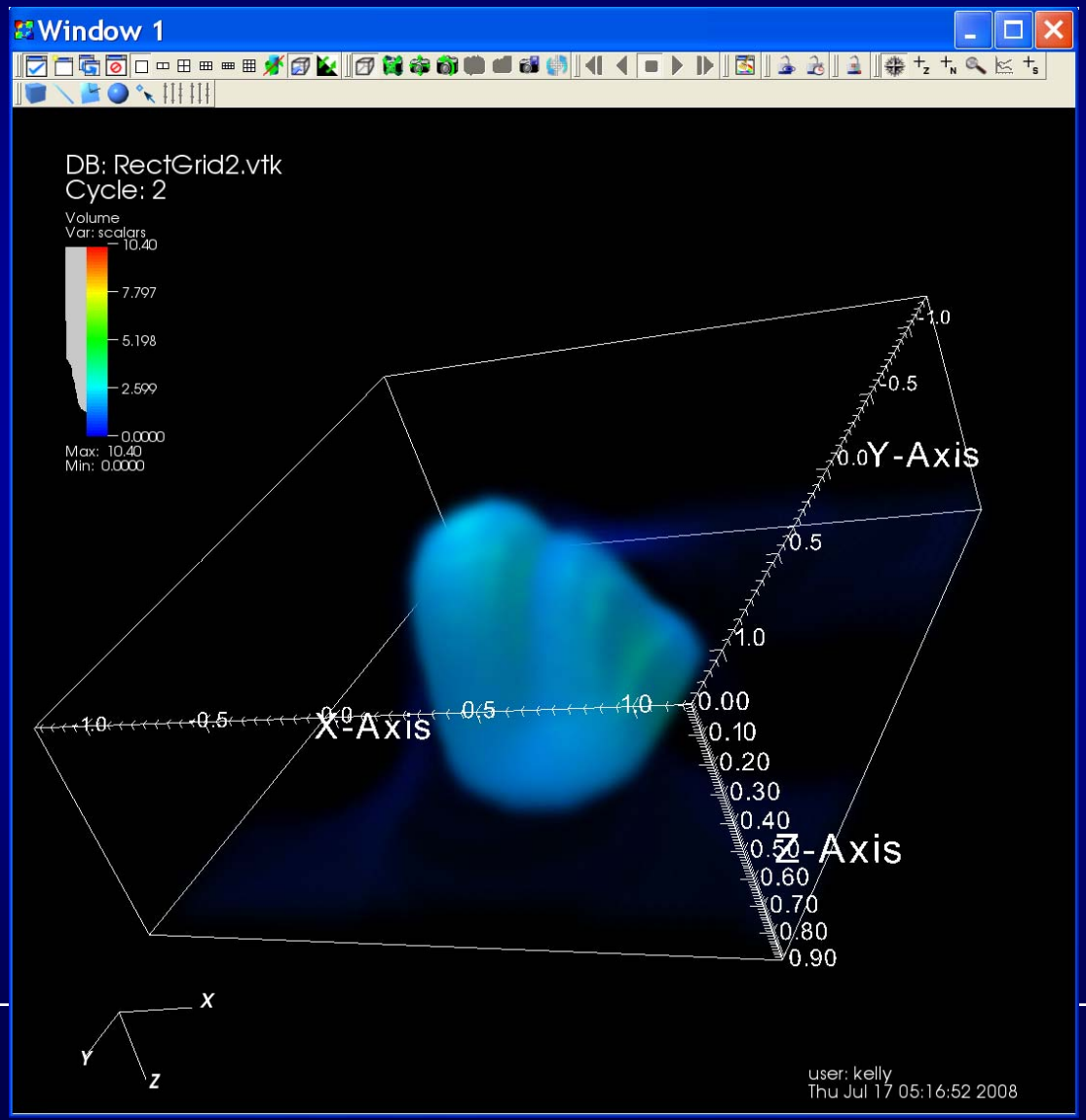
- Step 7: Create volume by selecting volume in plots



Download VisIt and Go Through Example

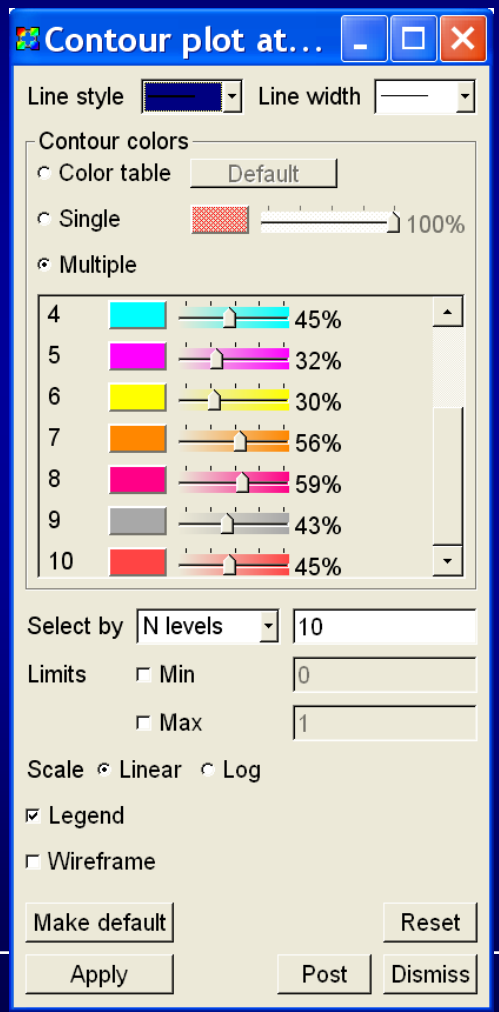
- Step 8: Adjust the volume rendering color profile

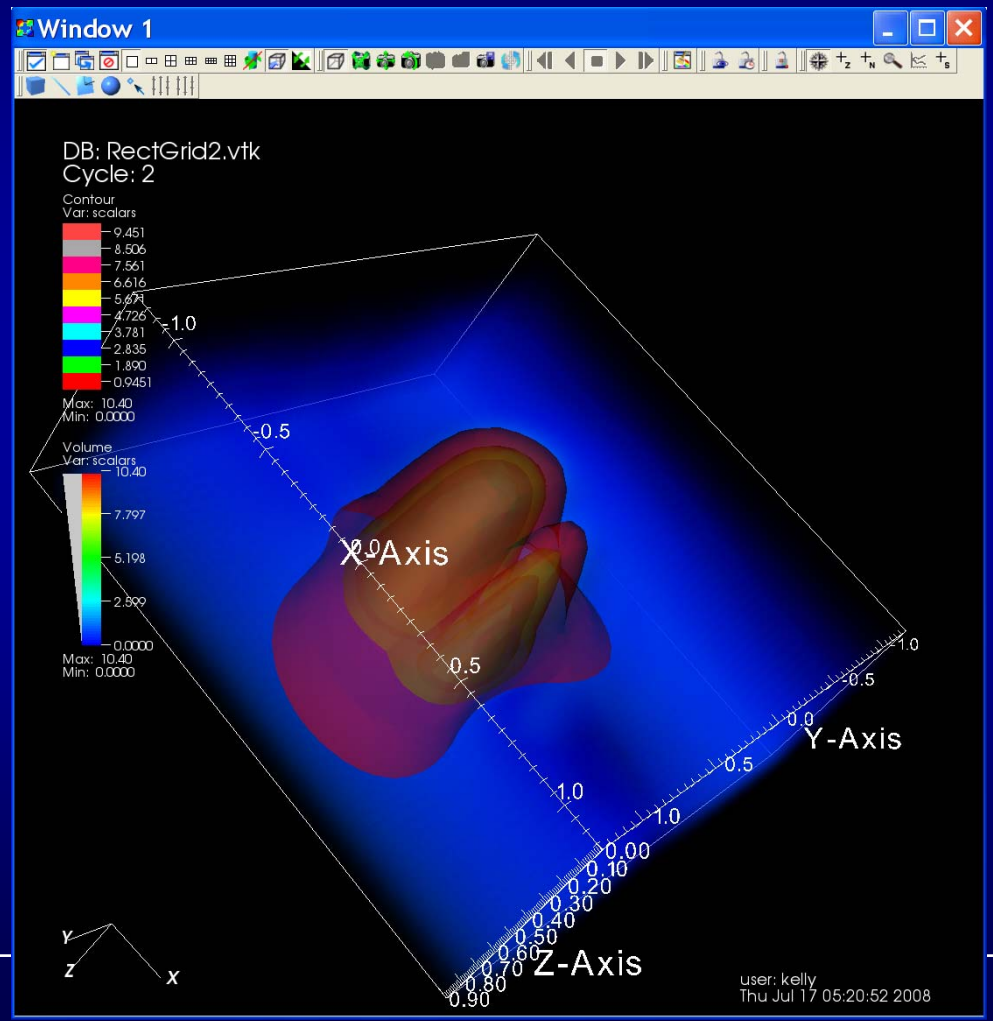




Download VisIt and Go Through Example

- Step 9: Adjust the transparency and color of isosurfaces





Topics Covered

- Remote and Collaborative Visualization
- EnVision – simplified Interface for Visualization
- Visualizing and Animating a Time Series in Paraview
- VisIt
- Parallel Visualization

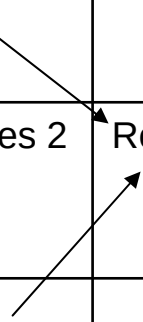
Parallel Visualization

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

Timesteps

Processes

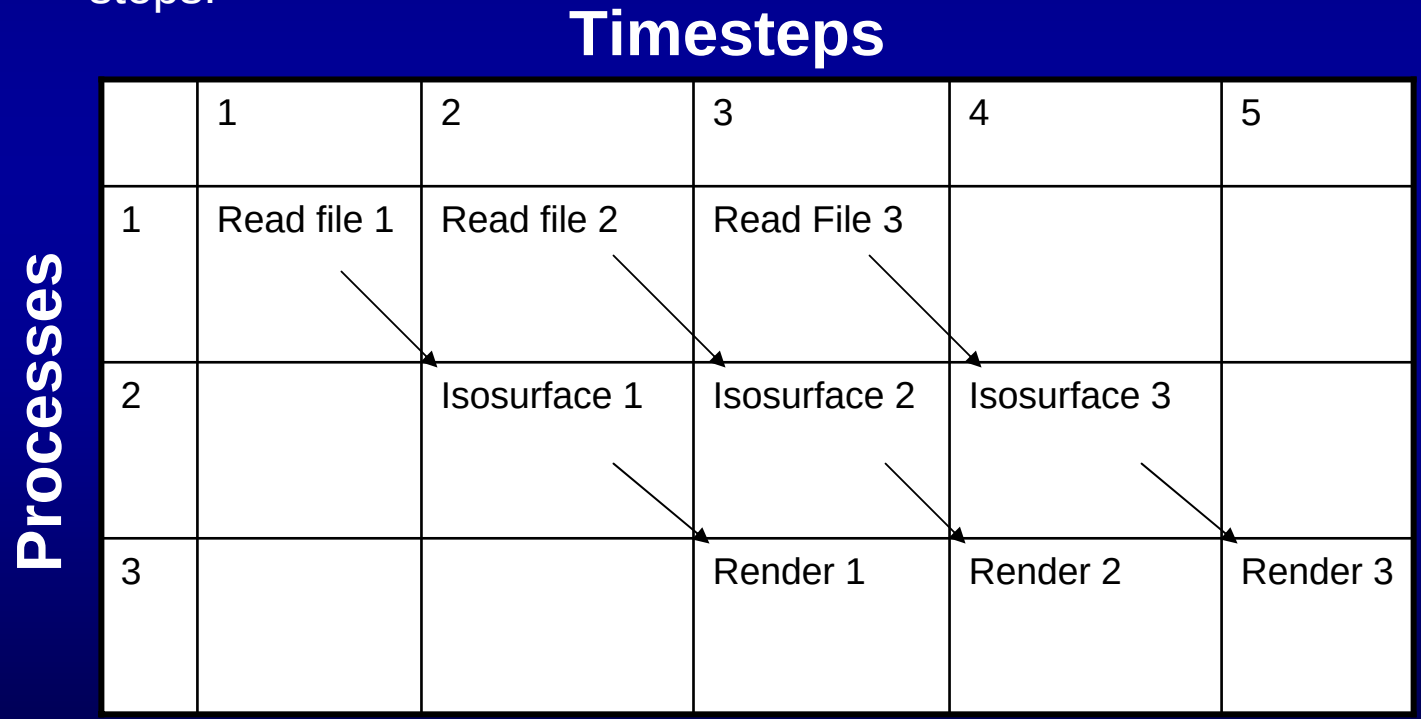
	1	2	3	4	5
1		Read file 1	Isosurface 1	Cut Plane 1	
2			Read file 2	Streamlines 2	Render
3	Read file 3	Triangulate 3	Decimate 3	Glyph 3	



TACC

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.



Parallel Visualization

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

Timesteps

Processes

	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 2	Render partition 3

Questions?

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 is configured to do so.
- It is possible to run the render server with less processes than the data server, but never more.

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 is configured to do so.
- It is possible to run the render server with less processes than the data server, but never more.

Parallel Paraview

- Paraview can be run in a variety of configurations
  - Client, Data Server and Render Server all running on the same process (the way we have been running this week)
  - Client as a single process program, Data Server and Render Server as MPI multi-process programs.
    - MPI is used to send messages between processes
    - Sockets are used to send messages and data between servers

Parallel Paraview

- Paraview can be run in a variety of configurations
  - Client, Data Server and Render Server all running on the same process (the way we have been running this week)
  - Client as a single process program, Data Server and Render Server as MPI multi-process programs.
    - MPI is used to send messages between processes
    - Sockets are used to send messages and data between servers

Parallel Paraview

- Distributed Stand-Alone Mode
  - mpirun –np 4 ./paraview
  - (Node 0: data server node 0, render server node 0, client)
  - (Node 1: data server node 1, render server node 1)
  - (Node 2: data server node 2, render server node 2)
  - (Node 3: data server node 3, render server node 3)

Parallel Paraview

- Client/Server Mode
  - ./paraview --client --host=server_host (on your client)
  - mpirun –np 4 ./pvserver (on your host server)

Running Parallel Paraview through VNC Session

- Login to maverick.tacc.utexas.edu
- Start your vnc session:
  - qsub /usr/local/qsub/RUN.vnc
- Connect to your vnc session
  - Remember to look in vnc_server for the display number
- Inside your vnc session:
  - ssh maverick /usr/local/qsub/paraview_mpi_64 $DISPLAY [num graphics devices]
  - Default number of graphics cards requested is 3
- Wait for paraview to show up in your vnc session

Running Parallel Paraview in Client-Server Mode

- Login to maverick.tacc.utexas.edu
- For PVSERVER_PORT below: choose something > 1024
- For 32 bit execution:
  - /usr/local/qsub/pvserver_mpi <PVSERVER_PORT> [num graphics devices]
- For 64 bit execution:
  - /usr/local/qsub/pvserver_mpi_64 <PVSERVER_PORT> [num graphics devices]
- Wait a minute for the pvservers to get started
- Connect to mav1.tacc.utexas.edu by:
  - pvclient –sh=mav1.tacc.utexas.edu –sp=<PVSERVER_PORT.

Questions?