Introduction to High-Performance Scientific Visualization

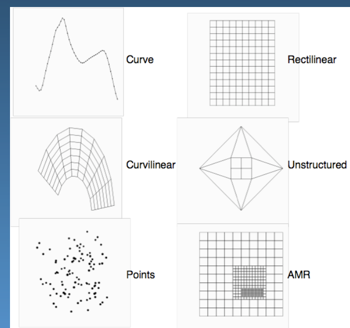
Software and Platform

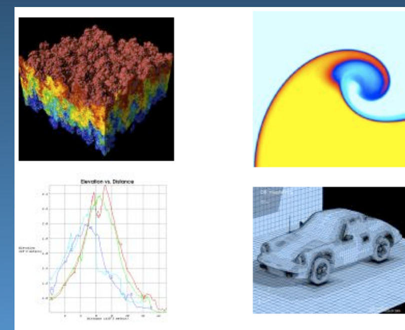
Jinghua Ge jinghuage@cct.lsu.edu
Advanced Visualization Service Lab, CCT, LSU

HPC training / Nov 3, 2010

Scientific Visualization

- Sciviz is part of scientific computing, is data-driven research
- What to expect from viz:
  - Construct observation of a phenomenon
  - Formulation of hypothesis to explain the phenomenon
  - Quantitatively predict the existence/results of new observations
  - Evaluate proposed methods and quantify effectiveness of the techniques

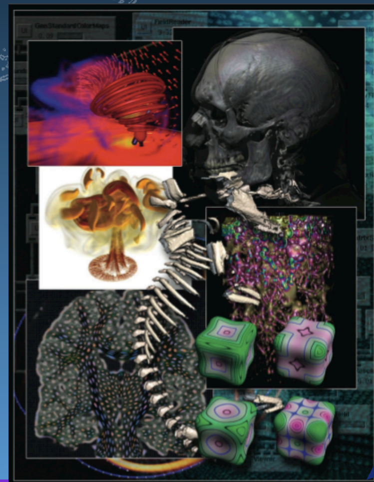




part of scientific computing, data-driven research. serve/construct description and observation of a phenomenon, formulation of a hypothesis to explain the phenomena, predict the quantitatively the existence/results of new observations, evaluation of the proposed methods and quantification of the effectiveness of their techniques.

Visualization Viewpoints, Chris Johnson

- CG&A, Visualization Viewpoints Column, 2004. Editor: Theresa-Marie Rhyne. Researchers talk about most important visualization problems
- Chris Johnson, University of Utah.



1. Think about science
2. Quantify effectiveness
3. Represent error and uncertainty
4. Perceptual Issues
5. Efficiently utilizing novel hardware architectures
6. Human-computer interaction
7. Global/local visualization (detail within context)
8. Integrated problem-solving environments (PSEs)
9. Multi-field visualization
10. Integrating scientific and information visualization
11. Feature Detection
12. Time-dependent visualization
13. Scalable, distributed and grid-based visualization
14. Visual abstraction
15. Theory of visualization

Quantitative analysis, Feature tracking, Handle large, dynamic, and complex data model. Define abstractions for the visualization and user interaction process. data models, workflow, human visual perception model, interaction model, distributed computing model.---guide efficient and usable software implementation

Organization

- Part One: High-Performance Visualization Overview -- Support in HPC infrastructure
- Part Two: Make them parallel -- Software Tutorial
- Part Three: Remote Visualization Enabled

High-Performance Visualization

- Prerequisites:
  - PBS Queueing System
  - Submit and monitor jobs - qsub, qstat
  - Level of Parallelism -- Task or MPI
- HPC architecture
  - Share memory architecture
  - Distributed memory architecture
- Level of Interaction
  - Batch processing
  - Interactive Visualization
  - Web portal, mashup
- Viewing
  - Local desktop
  - Remote View -- X11 forwarding, Remote Desktop(VNC) or Image Streaming
  - Tiled Display

Large scale scientific computing
HPC centers and Grid Infrastructure

- Grid organization
  - OSG--(bring scientists from different disciplines together for data-intensive research, provide high performance computing power to scientific problem solving, provide grid services for data integration, provide visualization techniques to users for data analysis and simulation steering)
  - The TeraGrid project is funded by the National Science Foundation and includes 11 partners: Indiana, LONI, NCAR, NCSA, NICS, ORNL, PSC, Purdue, SDSC, TACC and UC/ANL.
- Grid computing Intiatives for Science
  - BioGrid and its applications, NCBI BLAST
  - Earth science grid, http://www.earthsystemgrid.org/ and viz software: http://www.earthsystemgrid.org/browse/browse.htm?uri=http://datagrid.ucar.edu/metadata/scd/software.thredds
- Visualization as Service!
  - http://www.sdsc.edu/us/visservices/index.html
  - http://www.nersc.gov/nusers/visualization/
  - http://www.tacc.utexas.edu/research/users/
  - http://www.tacc.utexas.edu/research/users/features.php

Viz services and resources: TeraGrid

- TG dedicated viz resources
  - Purdue TeraDRE
  - NCSA Lincoln
  - TACC Longhorn
  - ANL IA-32 Viz
- TG Viz packages
  - Basic VTSS: Image processing: ImageMagick, Netpbm; 3D Viz: Paraview, VTK, Mesa; Distributed rendering: Chrominum
  - High quality rendering: Maya, POV-Ray, Blender, Gelato
  - Commercial viz software: Amira, Ensight, AVS, DX, IDL, EnVision, Techplot
  - Opensource viz software: VisIt, Paraview, OpenDX, EasyViz, Gnuplot, PYNGL/PYNOL
  - Opensource specialty software: NCAR Graphics, NCL, Vapor, VMD, Vis5D, RasMol, StarSplatter, NexRAD, Meshviewer, vl3, FELEET, Vista
  - Web portal: TG viz portal

Viz packages used in Teragrid sites (1)

- PSC -
  - Software: VisIt (www.llnl.gov/visit) and StarSplatter (http://www.psc.edu/Packages/StarSplatter_Home). The latter is the renderer for smooth particle hydrodynamics. Software developed at PSC (http://www.psc.edu/research/graphics/software.php)
- Purdue -
  - Software: TeraDRE (http://teradre.rcac.purdue.edu) and NexRAD Radar (https://gridsphere.rcac.purdue.edu:8443/gridsphere/gridsphere?cid=96&JavaScript=enabled). TeraDRE (Distributed Rendering Environment on the TeraGrid) is a resource that allows users to render their 3d animations using a cluster of over 4,000 machines. TeraDRE currently supports Maya, Gelato, Blender, POVRay. NexRAD Radar - Visualization of NexRAD radar data stream.
- LONI -
  - Software: (http://www.loni.org/teragrid/software.php) VisIt (www.llnl.gov/visit), VMD, Gnuplot.

Viz packages used in Teragrid sites (2)

- NCSA -
  - Software: IDL, NCAR Graphics, Netcdf, Tecplot, VTK, Paraview, Custom renderer, Maya, Blender, AfterEffects. NCSA visualization suite: Databridge + Easyviz(http://education.ncsa.uiuc.edu/products/dvs.html)
- NCAR -
  - Software: (http://www.vets.ucar.edu/software/index.shtml) : VAPOR (www.vapor.ucar.edu) , NCAR Graphics (http://www.ncarg.ucar.edu/), NCL (www.ncl.ucar.edu) , PyNGL/PyNIO, Vis5D+ . VAPOR is an open-source, targeted visual data analysis environment for earth sciences CFD data. VAPOR supports a multi-resolution data model that permits interactive data browsing of the largest simulation outputs using only a commodity PC and a consumer graphics card. VAPOR is integrated with ITT's Interactive Data Language (IDL), providing quantitative capabilities and mathematical data operators. NCAR Graphics and NCL are products of Computational & Information System Laboratory, National Center of Atmospheric Research, used for analysis of geo-referenced data (e.g. climate and weather).

Viz packages used in Teragrid sites (3)

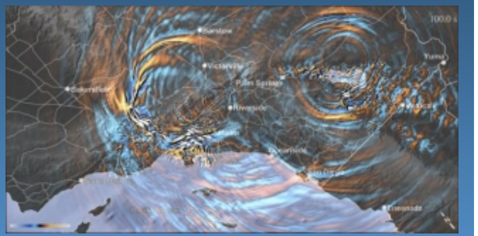
- TACC –
  - Software: (http://www.tacc.utexas.edu/resources/software/software_modules.php) VTSS, VisIt, Amira, EnSight, AVS, Ferret, DX, Vis5D, VMD, IDL, NCL, and EnVision
- SDSC -
  - Software: packages installed and used at SDSC are listed here: http://visservices.sdsc.edu/software/installed.php. NCL, Paraview, Vapor, Molden, RasMol, Techplot, Gnuplot, VisIt, Vista, VTK, ImageMagick, Gimp, MPEG2 Codec. Specialized software developed at SDSC and the UCSD campus is: http://visservices.sdsc.edu/software/software Mesh Viewer, VISTA Volume Renderer, DeskVOX, MayaTools
- ANL -
  - Software: (http://www.teragrid.org/userinfo/data/vis/uc_anl_sw.php) VTSS, POV-Ray, VisIt, VMD, Vl3, Vis Gateway (Paraview web portal). Host profile for UC/ANL TeraGrid is now included in the VisIt 1.10.0 distribution. vl3: a volume rendering library and application developed at the University of Chicago and Argonne. Vis Gateway: provides simplified access to launching the ParaView server on the UC/ANL cluster.

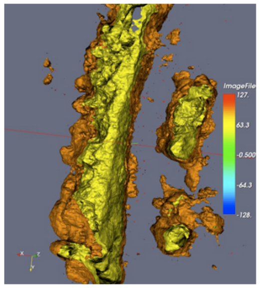
More viz software and service

- UMN SuperComputing Center -
  - http://www.msi.umn.edu/cgi-bin/soft/listing.html?subject_id=19&lab_id=&parent=1
- NorthWestern VisLab -
  - Lab installed software http://vislab.northwestern.edu/software.htm
  - General Visualization, Astronomy, Chemistry, GIS
- National Energy Research Scientific Computing Center -
  - Software installed: (http://www-vis.lbl.gov/NERSC/Software/) AVS5, AVS/Express, Ensight, gnuplot, IDL, VisIT, Ghostview, Gimp, ImageMagick, ferret, garlic, gsharp, grace, OpenDX, Paraview, rasmol, vmd, XV
  - Self developed software: (http://www-vis.lbl.gov/Software/) H5Part, svPerfGL, mpiReadWriteTest, semViewer, Visapult
- Clouds ----

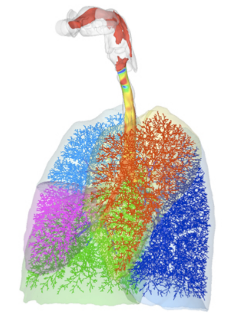
Visualization highlights in HPC

- http://visservices.sdsc.edu/projects/scec/shakeout/SO2.g3d7/
- Connecting the Virtual Human. http://www.tacc.utexas.edu/research/users/features/dynamic.php?m_b_c=karniadakis
- More Than A Magnifying Glass. http://www.tacc.utexas.edu/research/users/features/index.php?m_b_c=gilpin





This image shows a surface rendered view of a 3D reconstructed synaptic membrane. [Image courtesy of Chris Gilpin]



This image shows a computerized representation of a subject-specific breathing human lung. The airflow in the CT-based airway tree is simulated on TACC supercomputers, Lonestar and Ranger, by the multiscale 3D-1D coupled CFD technique.

Data and Software: Summary

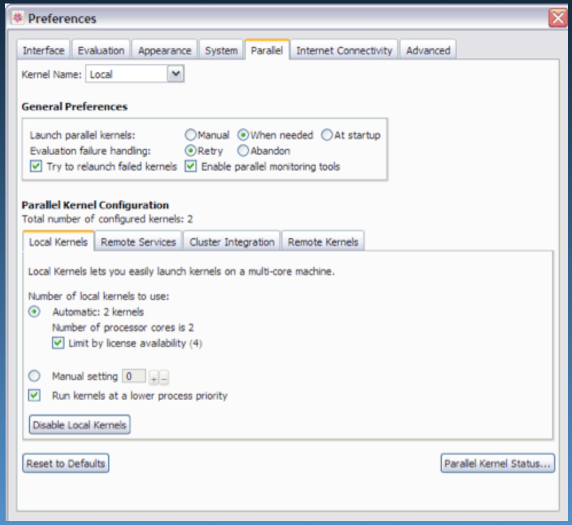
- PostProcessing: Matlab, Mathematica,
- Hi-quality rendering: Maya, POV-Ray*
- Commercial Viz package: Avizo, Ensight (CFD, parallel), Tecplot, ScienceGL.
- Open source: VisIt* (parallel) , Paraview* (parallel), OpenDX, SciRun
- Molecular: VMD*, PyMol*, Rasmol, Molden,AtomEye, bioconductor, MCell
- Medical (CT, MRI, DICOM): Osirix*, ImageJ, Imaris, ITK, Slicer
- WRF, wave surge: VAPOR*, SMS, Ferret
- GIS: ArcGis, GoogleEarth, GRASS, 3DEM
- Astronomy: Splash, Partview, Starsplatter
- Volume rendering: meshviewer, vista, vl3
- Plot: Gnuplot, Igor, Grace, pgplot
- Viz pipeline: VTK*, VisTrail*

Organization

- Part One: High-Performance Visualization Overview -- Support in HPC infrastructure
- Part Two: Make them parallel -- Software Tutorial
- Part Three: Remote Visualization Enabled

Mathematica

- http://reference.wolfram.com/mathematica/ParallelTools/tutorial/Overview.html



Ensight

- http://www.ensight.com/Manuals/
- EnSight Gold supports shared- memory parallel rendering
  - Config file specify screens. -- screen-based parallelism

    CVFd 1.0 # # conference room display wall # display wallresolution
    2560 2048 screen # lower-left
    displayid :0.1 resolution 1280 1024 wallorigin 0 0
    screen # lower-right displayid :0.2 resolution 1280 1024 wallorigin 1280 0
    screen # upper-left displayid :0.3 resolution 1280 1024 wallorigin 0 1024
    screen # upper-right displayid :0.4 resolution 1280 1024 wallorigin 1280 1024

- EnSight DR supports distributed-memory parallel rendering
  - Config file to setup collaboration hub, composting, and clients. -- data-split based parallelism

    Router [hostname] [options]
    pc [options]
    client [hostname] [options]
    client [hostname] [options]

IDL

- http://www.ittvis.com/ProductServices/IDL.aspx
- IDL enables multi-threading -- shared memory parallel
  – submit IDL job as a serial job to cluster nodes
  – batching process, when plotting, need some extra code to save image files to disk
  – similar with matlab: https://spaces.umbc.edu/display/hpc/Running+Matlab+on+HPC

```
#!/bin/bash
#PBS -N 'hello_parallel'
#PBS -o 'qsub.out'
#PBS -e 'qsub.err'
#PBS -W umask=007
#PBS -q low_priority
#PBS -l nodes=1:ppn=4
#PBS -m bea
cd $PBS_O_WORKDIR
idl -e main
```

```
qsub sayhello.qsub
```

- To enable distributed memory parallel:
  – http://www.txcorp.com/products/FastDL/
  – TaskDL -- Task server, and multiple IDL task
  – MPIDL -- MPI implementation as native IDL function calls

yt

- http://yt.enzotools.org/
- http://yt.enzotools.org/doc/howto/parallel_computation.html
- Empowered by mpi4py: http://code.google.com/p/mpi4py/
- Spatical decomposition, Grid decomposition

```
from yt.mods import *
pf = load("RD0035/RedshiftOutput0035")
v, c = pf.h.find_max("Density")
print v, c
pc = PlotCollection(pf, center = [0.5, 0.5, 0.5])
pc.add_projection("Density", 0)
pc.save()
```

```
mpirun -np 16 python2.6 my_script.py --parallel
```

VisIt, ParaView

- https://wci.llnl.gov/codes/visit/about.html
- http://www.paraview.org/paraview/project/features.html
- Visualization Capabilities: Handles structured (uniform rectilinear, non-uniform rectilinear, and curvilinear grids), unstructured, polygonal, image, multi-block and AMR data types. Contours, Clipping, Streamlines, data inspection
- Input/Output and File formats: http://www.visitusers.org/index.php?title=Detailed_list_of_file_formats_VisIt_supports
- User Interaction: 3D widgets, LOD
- Large Data and Distributed Compuing: Data Parallel model, Distributed rendering
- Scripting and Extensibility: Python interface

VisIt : Distribution and Documentation

- Binary distributions
  - https://wci.llnl.gov/codes/visit/executables.html
  - Windows
  - Linux
  - MacOS X
  - AIX (IBM)

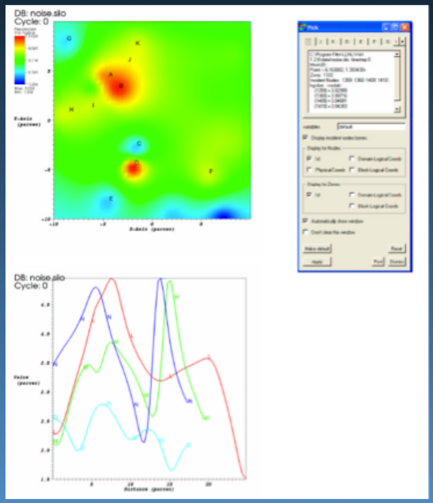
Same user interface on each platform

- Documentation
  - https://wci.llnl.gov/codes/visit/manuals.html
  - User's manual
  - Python Interface manual
  - Get Data into VisIt
  - Tutorials

- Wiki : http://www.visitusers.org/index.php?title=Main_Page

Visualization and Analysis

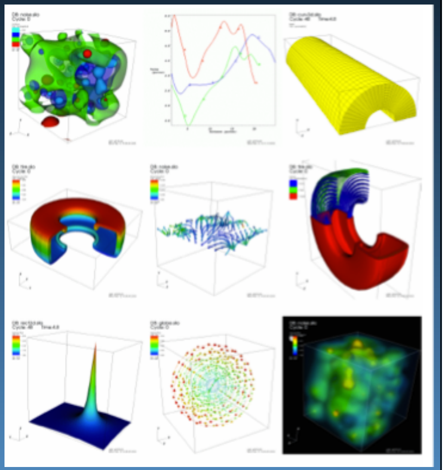
- Manipulate data or create new data using operators and expressions
- 2D and 3D Plots
- Develop new plots and operators as plug-ins
- Query and Quantitative Analysis



Plots

- A plot is a viewable object, created from a database, that can be displayed in a visualization window
- Plots come from plug-ins so you can extend VisIt's plotting capabilities by writing a new plug-in

- Type of plots
  - Pseudocolor
  - Mesh
  - FilledBoundary
  - Boundary
  - Contour
  - Volume
  - Vector
  - Surface
  - Subset
  - Streamline
  - Curve
  - Histogram
  - Tensor



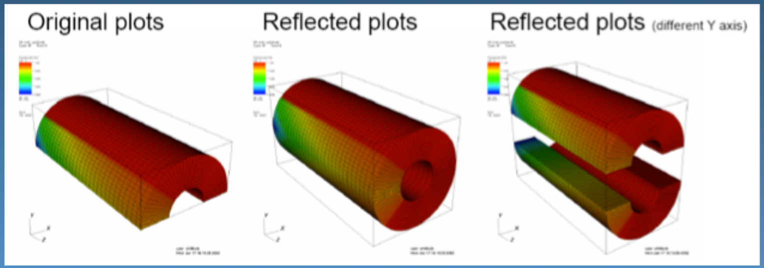
Operators

• An operator is a filter that is applied to a database variable before the compute engine uses that variable to generate a plot

• Operators come from plug-ins so you can extend VisIt's data manipulation capabilities by writing a new plug-in

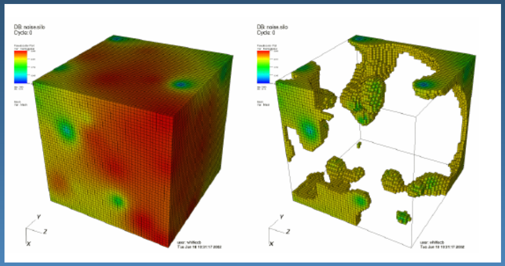
Reflect Operator

- The Reflect operator reflects database geometry across one or more axes
- Use this operator when your simulation data contains only part of the geometry and relies on symmetry to recover the rest of the geometry



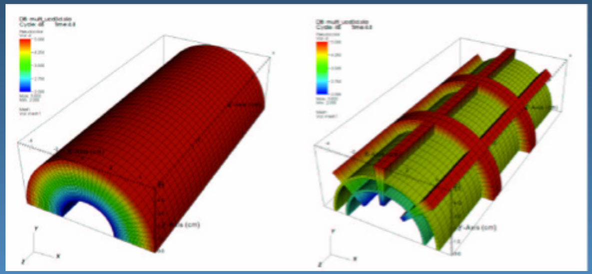
Threshold Operator

- Removes cells whose value is not in the specified range
- Use this operator when you only want to look at cells that have values within an interesting range



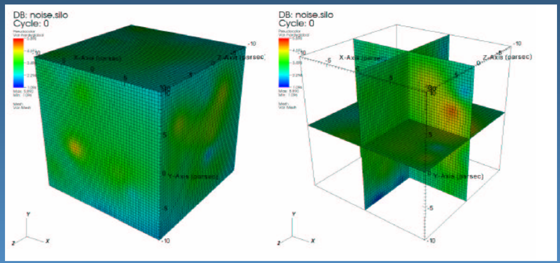
Inverse Ghost Zone Operator

- This operator makes ghost zones visible and real zones invisible
- Use this operator when you want to look at your database's ghost zones



Three Slice Operator

- This operator slices plots using three axisaligned slice planes
- Use this operator when you want to see into the interior of 3D plots



Quantitative Analysis: Expressions

– Use expressions to derive quantities that were not stored in your database

– Expressions can operate on scalars, vectors, tensors, or on meshes

– VisIt provides built-in functions

- Trigonometry functions: sin, cos, deg2rad, rad2deg...
- Math functions: ln, log10, sqrt, abs, min, max, mod...
- Vector functions: normalize, magnitude, cross, dot...
- Image Processing: mean_filter, median_filter

Quantitative Analysis: Query, Pick, Linout

- Query: Database queries, point queries, and line queries
  - Area, Centroid, Compactness
  - Minmax, num nodes/zones
  - Connected components
- Pick: Node pick, Zone pick
- Lineout: 2D, 3D, profile values and draw as curve plot

Interaction and Programming Interfaces

• Use VisIt as an application or a library

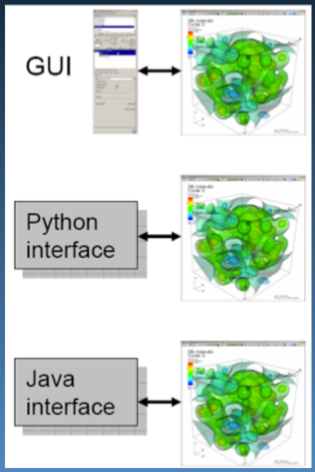
• C++, Python, Java interfaces allow other applications to control VisIt

• Interfaces
  – Graphical user interface
  – Python programming interface
  – Java programming interface
  – C++ programming interface

• All interfaces send commands to the viewer and in turn get the latest state from the viewer

• Use GUI when interaction is required

• Use Python interface to script actions or use VisIt as a batch mode movie generation tool



Example:
Batch Processing Using Python Script

```
def get_lineout_data():

    A=(1125190, 170726)
    B=(1125340, 170223)

    g = GetGlobalLineoutAttributes()
    g.windowId = 2
    g.createWindow = 0
    g.curveOption = g.CreateCurve
    g.colorOption = g.CreateColor
    SetGlobalLineoutAttributes(g)

    Lineout((A[0], A[1]), (B[0], B[1]))
    for i in range(0,8):
        toggle_sets(i)
        Lineout((A[0], A[1]), (B[0], B[1]))
```

```
SetWindowLayout(2)
for n in range(1, 37):
    demo(n)
```

```
def demo(n):

    SetActiveWindow(1)
    OpenDatabase("/Users/jinghua/Develop/visit_data/
lak_%04d.nc" % n)

    AddPlot("Pseudocolor", "salinity")
    AddOperator("Project")
    DrawPlots()

    get_lineout_data()

    for state in range(TimeSliderGetNStates()):
        SetTimeSliderState(state)
        SetActiveWindow(2);
        lineout_data = save_window("curve")

    for i in range(0, 2):
        SetActiveWindow(i+1)
        ClearWindow()
        DeleteAllPlots()

    CloseDatabase("/Users/jinghua/Develop/visit_data/
lak_%04d.nc" % n)
```

Remote and Parallel Compute Engine

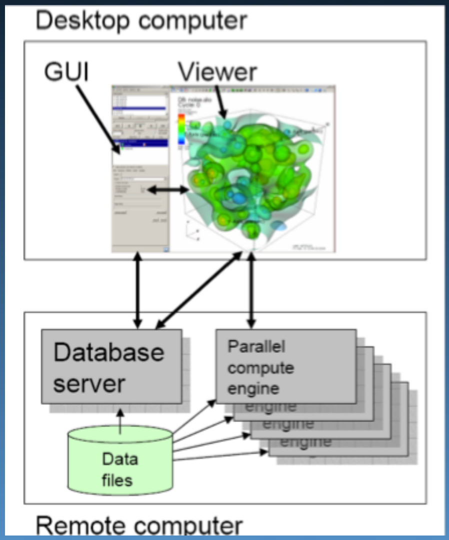
- GUI on your desktop computer and parallel compute engine on a remote supercomputer
- Scalable rendering in parallel for largest datasets, image streaming back to user
- Down-sampled data send back to local desktop to make use of graphics hardware

Software Architecture

- 4 main components
  - Graphical User Interface (GUI)
  - Viewer
  - Database server
  - Compute engine

- GUI and Viewer usually meant to run locally on your desktop computer

- Database server and parallel compute engine can run on remote computers where the data files are located and talk to the GUI and viewer running on your desktop computer



Make It Parallel :
Data Decomposition #1: Bov

BOV header for raw data:

DATA_FILE: all.raw
DATA_SIZE: 2048 2048 2048
DATA_FORMAT: FLOAT
VARIABLE: volume
DATA_ENDIAN: LITTLE
CENTERING: nodal
BRICK_ORIGIN: 0. 0. 0.
BRICK_SIZE: 2048. 2048. 2048.
DIVIDE_BRICK: true
DATA_BRICKLETS: 512 512 512
DATA_COMPONENTS: 1

Make It Parallel : Data Decomposition #2: Silo

DBCreate	DBCreate
DBMkdir	DBMkdir
DBSetDir	DBSetDir
DBPutQuadmesh	DBPutZonelist
DBPutQuadvar1	DBPutFacelist
DBPutQuadvar2	DBPutUcdmesh
...	DBPutMaterial
DBSetDir	DBPutUcdvar1
	...
DBClose	DBSetDir
	DBClose

Make It Parallel : Data Decomposition #2: Multimesh silo

- Create silo file
- Write a quadmesh object for each decomposed grid of the whole data
- Write a quadvar object for each decomposed var of the whole data
- Write a virtual multimesh combining all quadmesh objects
- Write a virtual multivar combining all quadvar objects
- Write multivar to multimesh map
- Close silo file
- Two level hierarchy

Make It Parallel : Data Combination #1: XDMF

http://www.visitusers.org/index.php?title=Using_XDMF_to_read_HDF5

http://www.xdmf.org/index.php/XDMF_Model_and_Format

Make It Parallel : Data Combination #1: XDMF

```xml
<?xml version="1.0" ?>
<!DOCTYPE Xdmf SYSTEM "Xdmf.dtd" []>
<Xdmf Version="2.0">
 <Domain>
 <Grid Name="grid" GridType="Collection" CollectionType="Spatial">

  <Grid Name="grid0" GridType="Uniform">
 <Topology TopologyType="3DRectMesh" NumberOfElements="500 500 24"/>
    <Geometry GeometryType="X_Y_Z">
      <DataItem Name="X" Dimensions="24" NumberType="Float" Precision="4"
Format="HDF">
        hdfOP110000.h5:/HDF4_DIMGROUP/fakeDim2
      </DataItem>
      <DataItem Name="Y" Dimensions="500" NumberType="Float"
Precision="4" Format="HDF">
        hdfOP110000.h5:/HDF4_DIMGROUP/fakeDim0
      </DataItem>
      <DataItem Name="Z" Dimensions="500" NumberType="Float"
Precision="4" Format="HDF">
        hdfOP110000.h5:/HDF4_DIMGROUP/fakeDim1
      </DataItem>
    </Geometry>
    <Attribute Name="Data-Set-2" AttributeType="Scalar" Center="Node">
      <DataItem Dimensions="500 500 24" NumberType="Float" Precision="4"
Format="HDF">
        hdfOP000000.h5:/Data-Set-2
      </DataItem>
    </Attribute>
    <Attribute Name="Data-Set-3" AttributeType="Scalar" Center="Node">
      <DataItem Dimensions="500 500 24" NumberType="Float" Precision="4"
Format="HDF">
        hdfOP000000.h5:/Data-Set-3
      </DataItem>
    </Attribute>
  </Grid>

<Grid Name="grid1" GridType="Uniform">
 <Topology TopologyType="3DRectMesh" NumberOfElements="500 500 24"/>
    <Geometry GeometryType="X_Y_Z">
      <DataItem Name="X" Dimensions="24" NumberType="Float" Precision="4"
Format="HDF">
        hdfOP110000.h5:/HDF4_DIMGROUP/fakeDim2
      </DataItem>
      <DataItem Name="Y" Dimensions="500" NumberType="Float"
Precision="4" Format="HDF">
        hdfOP110000.h5:/HDF4_DIMGROUP/fakeDim0
      </DataItem>
      <DataItem Name="Z" Dimensions="500" NumberType="Float"
Precision="4" Format="HDF">
        hdfOP110000.h5:/HDF4_DIMGROUP/fakeDim1
      </DataItem>
    </Geometry>
    <Attribute Name="Data-Set-2" AttributeType="Scalar" Center="Node">
      <DataItem Dimensions="500 500 24" NumberType="Float" Precision="4"
Format="HDF">
        hdfOP010000.h5:/Data-Set-2
      </DataItem>
    </Attribute>
    <Attribute Name="Data-Set-3" AttributeType="Scalar" Center="Node">
      <DataItem Dimensions="500 500 24" NumberType="Float" Precision="4"
Format="HDF">
        hdfOP010000.h5:/Data-Set-3
      </DataItem>
    </Attribute>
    </Grid>

 </Grid>
</Domain>
</Xdmf>
```

Make It Parallel :
Data Combination #2: Multimesh silo

- Single silo file for each cpu output
- Write a virtual multimesh combining all quadmesh objects in all silo files
  - mesh_name[i]: [<silo-filename>:]<path-to-mesh>
- Write a virtual multivar combining all quadvar objects in all silo files
  - var_name[i]: [<silo-filename>:]<path-to-var>
- Write multivar to multimesh map
- Close silo file
- Two level hierarchy

Visualization Using VisIt (Demo)

- Host profiles
- Compute Engines
- Expressions
- Query
- Session file
- Control
- Advanced data analysis:
  - http://www.visitusers.org/index.php?title=Connected_components
  - http://www.visitusers.org/index.php?title=Cmfe
  - http://www.visitusers.org/index.php?title=TimeSliceDiff
  - http://www.visitusers.org/index.php?title=Using_CLI

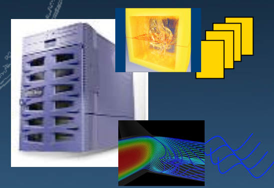
Pick viz software based on your tasks

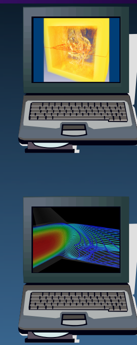
- Data format transform.
  - VisIt have TecPlot, Ensight Gold, FVCOM, NetCDF, STL reader. (https://wci.llnl.gov/codes/visit/FAQ.html#12). VTK file writing tool (C and Python), SILO API, XDMF schema for HDF data. Paraview doesn't have a lot of these database readers.
  - Data format support multiple domain decomposition. Silo, BOV, XDMF
- Viz algorithms: streamline. AMR. multiblock. etc.
- Performance! Can the software make use of multiple CPU/GPUs?
- Flexibility: Analyze and probe your data.
  - Does the software provide flexible tools to visualize and analyze your data creatively? Can you design your own workflow? write python source code in vistrails, create complicated expressions in VisIt. Can you extend the software?: Create reusable module in vistrails and write plugins for VisIt.
- Paper and Presentation:
  - high-res offscreen rendering. Visit: save window, Vistrail: offscreen buffer, html.
  - VisIt: save movie, visit -movie -s your_own_movie_script

Organization

- Part One: High-Performance Visualization Overview -- Support in HPC infrastructure
- Part Two: Make them parallel -- Software Tutorial
- Part Three: Remote Visualization Enabled

Remote visualization





– Why remote visualization?
– Architectures and how to use.
  - Software in client-server mode: visit, paraview
  - VirtualGL and TurboVNC: spider, spur. opengl app X forwarding.
  - webportal at teragrid visualization gateway.

Remote visualization on Longhorn(1): Connect to Longhorn

http://services.tacc.utexas.edu/index.php/longhorn-user-guide

When you add spur as a resource for your TG allocaton, Activate account at TACC: https://tas.tacc.utexas.edu/TASMigration/AccountActivation.aspx then you will have passwd to direct connect to longhorn.tacc.utexas.edu

- ssh jge@tg-login.spur.tacc.teragrid.org, (or jge@spur.tacc.utexas.edu)

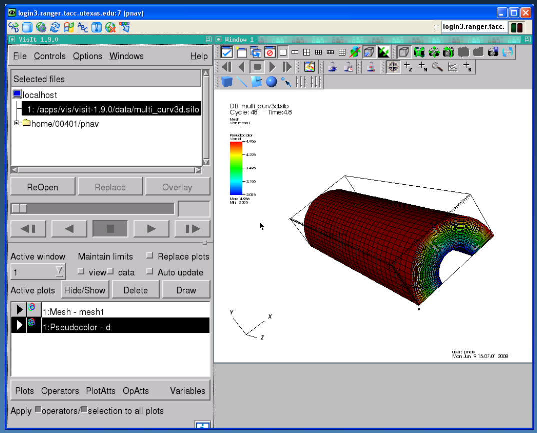
- vncpasswd
- qsub /share/sge/default/pe_scripts/job.vnc -geometry 1440x900
- (This script can be copied to your home directory and modified, particularly if you would like to increase the runtime of your job (currently limited to 24 hours). At present, the command-line option "-l h_rt" is not read properly by qsub.)
- tail -f ~/vncserver.out, to find server port once vnc server is started
- ssh -g -L 59xx:longhorn.tacc.utexas.edu:59yy jge@longhorn.tacc.utexas.edu
- connect the VNC client to localhost:59xx. (make sure use X11 in Mac to run /opt/TurboVNC/bin/vncviewer)
- (download vnc client here: TurboVNC: http://www.virtualgl.org/About/TurboVNC)

- load viz modules: **module load vis**
- Load the VisIt module: **module load visit**
- Launch VisIt: **vglrun visit**

Remote visualization on Longhorn (2): Configure a parallel host profile on VisIT

- Open the host profile: <Ctrl-H> or Options -> Host Profiles
- Click the button "New Profile"
- Under the "Selected profile" tab:
- Name the profile, e.g. "spur parallel"
- Remote host name will be the current vis node: localhost
- Host name aliases: vis*.ranger.tacc.utexas.edu
- This will permit this profile to be used from any node on spur
- Check the "Parallel computation engine" box
- (this activates the "Parallel options" tab)
- Under the "Parallel options" tab:
- Check the "Parallel launch method" box, and select "poe"
- Set the "Default number of processors" field to a value greater than one.
- Under the "Advanced options" tab:
- Check the box "Use VisIt script to set up parallel environment"
- Check the box "Tunnel data connections through SSH"
- Click the button "Apply"
- Click the button "Dismiss"
- Save your configuration! Select Options -> Save Settings

Remote visualization on Longhorn (Demo)



Longhorn Web Portal (Demo)