Argonne Training Program on Extreme-Scale Computing (ATPESC)

Data Analysis and Visualization
## Visualization & Data Analysis

<table>
<thead>
<tr>
<th>Time</th>
<th>Title of presentation</th>
<th>Lecturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:30 am</td>
<td><strong>Visualization Introduction</strong></td>
<td>Mike Papka, Joe Insley, Silvio Rizzi, ANL</td>
</tr>
<tr>
<td>10:15 am</td>
<td>Break</td>
<td></td>
</tr>
<tr>
<td>10:45 am</td>
<td><strong>Large Scale Visualization with ParaView</strong></td>
<td>Dave DeMarle, Kitware</td>
</tr>
<tr>
<td>12:30 pm</td>
<td>Lunch and Hands-on Exercises</td>
<td></td>
</tr>
<tr>
<td>1:30 pm</td>
<td><strong>Visualization and Analysis of Massive Data with VisIt</strong></td>
<td>Cyrus Harrison, LLNL</td>
</tr>
<tr>
<td>3:15 pm</td>
<td>Break</td>
<td></td>
</tr>
<tr>
<td>3:45 pm</td>
<td><strong>Scalable Molecular Visualization and Analysis Tools in VMD</strong></td>
<td>John Stone, UIUC</td>
</tr>
<tr>
<td>4:30 pm</td>
<td><strong>Exploring Visualization with Jupyter Notebooks</strong></td>
<td>Mike Papka, Joe Insley, Silvio Rizzi, ANL</td>
</tr>
<tr>
<td>5:30 pm</td>
<td><strong>Dinner Talk: DreamWorks Animation</strong></td>
<td>Mark Jackels</td>
</tr>
<tr>
<td>6:30 pm</td>
<td><strong>Hands-on Exercises</strong></td>
<td></td>
</tr>
</tbody>
</table>
Here’s the plan...

- Examples of visualizations
- Visualization resources
- Visualization tools and formats
- Data representations
- Visualization for debugging
- In Situ Visualization and Analysis
Climate

Data courtesy of: Mark Taylor, Sandia National Laboratory; Rob Jacob, Argonne National Laboratory; Warren Washington, National Center for Atmospheric Research
Aerospace
(Jet Nozzle Noise)

Data courtesy of: Anurag Gupta and Umesh Paliath, General Electric Global Research
Cosmology

Data courtesy of: Salman Habib, Katrin Heitmann, and the HACC team, Argonne National Laboratory
Cooley: Analytics/Visualization cluster

Peak 223 TF
126 nodes; each node has
– Two Intel Xeon E5-2620 Haswell 2.4 GHz 6-core processors
– NVIDIA Telsa K80 graphics processing unit (24GB)
– 384 GB of RAM
Aggregate RAM of 47 TB
Aggregate GPU memory of ~3TB
Cray CS System
216 port FDR IB switch with uplinks to our QDR infrastructure
Mounts the same file systems as Mira, Cetus, Theta
Visualization Tools and Data Formats
All Sorts of Tools

Visualization Applications
– VisIt ✭
– ParaView ✭
– EnSight

Domain Specific
– VMD, PyMol, Ovito

APIs
– VTK: visualization
– ITK: segmentation & registration

GPU performance
– vl3: shader-based volume and particle rendering

Analysis Environments
– Matlab
– Parallel R

Utilities
– GnuPlot
– ImageMagick

Available on Cooley
Available on Theta
ParaView & VisIt vs. vtk

ParaView & VisIt
- General purpose visualization applications
- GUI-based
- Client / Server model to support remote visualization
- Scriptable / Extendable
- Built on top of vtk (largely)
- In situ capabilities

vtk
- Programming environment / API
- Additional capabilities, finer control
- Smaller memory footprint
- Requires more expertise (build custom applications)
## Data File Formats (ParaView & VisIt)

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTK</td>
<td>Parallel (partitioned) VTK SpyPlot CTH</td>
</tr>
<tr>
<td>VTK MultiBlock</td>
<td>HDF5 raw image data</td>
</tr>
<tr>
<td>Legacy VTK</td>
<td>DEM</td>
</tr>
<tr>
<td>Legacy VTK</td>
<td>VRML</td>
</tr>
<tr>
<td>EnSight files</td>
<td>PLY</td>
</tr>
<tr>
<td>EnSight Master Server</td>
<td>Polygonal Protein Data Bank</td>
</tr>
<tr>
<td>Exodus</td>
<td>Xmol Molecule</td>
</tr>
<tr>
<td>BYU</td>
<td>Stereo Lithography</td>
</tr>
<tr>
<td>XDMF</td>
<td>Gaussian Cube</td>
</tr>
<tr>
<td>PLOT2D</td>
<td>Raw (binary)</td>
</tr>
<tr>
<td>PLOT3D</td>
<td>AVS</td>
</tr>
<tr>
<td>Facet</td>
<td>Meta Image</td>
</tr>
<tr>
<td>PNG</td>
<td>Tetrad</td>
</tr>
<tr>
<td>SAF</td>
<td>UNIC</td>
</tr>
<tr>
<td>LS-Dyna</td>
<td>VASP</td>
</tr>
<tr>
<td>Nek5000</td>
<td>ZeusMP</td>
</tr>
<tr>
<td>OVERFLOW</td>
<td>ANALYZE</td>
</tr>
<tr>
<td>paraDIS</td>
<td>BOV</td>
</tr>
<tr>
<td>PATRAN</td>
<td>GMV</td>
</tr>
<tr>
<td>PFLOTRAN</td>
<td>Tecplot</td>
</tr>
<tr>
<td>Pixie</td>
<td>Vis5D</td>
</tr>
<tr>
<td>PuReMD</td>
<td>Xmdv</td>
</tr>
<tr>
<td>S3D</td>
<td>XSF</td>
</tr>
<tr>
<td>SAS</td>
<td></td>
</tr>
</tbody>
</table>
Data Representations
Data Representations: Volume Rendering
Data Representations: Glyphs

2D or 3D geometric object to represent point data
Location dictated by coordinate
- 3D location on mesh
- 2D position in table/graph
Attributes of graphical entity dictated by attributes of data
- color, size, orientation
Data Representations: Contours (Isosurfaces)

A Line (2D) or Surface (3D), representing a constant value

VisIt & ParaView:
– good at this

vtk:
– same, but again requires more effort
Data Representations: Cutting Planes

Slice a plane through the data
– Can apply additional visualization methods to resulting plane
VisIt & ParaView & vtk good at this
VMD has similar capabilities for some data formats
Data Representations: Streamlines

From vector field on a mesh (needs connectivity)
– Show the direction an element will travel in at any point in time. VisIt & ParaView & vtk good at this
Molecular Dynamics Visualization

VMD:
- Lots of domain-specific representations
- Many different file formats
- Animation
- Scriptable

VisIt & ParaView:
- Limited support for these types of representations, but improving

VTK:
- Anything’s possible if you try hard enough
Visualization for Debugging
Visualization for Debugging
Visualization for Debugging
Visualization as Diagnostics: Color by Thread ID
In Situ Visualization and Analysis
In situ visualization and analysis with SENSEI

Slides courtesy SENSEI in situ project:

www.sensei-insitu.org
The Need of *In Situ* Analysis and Visualization

Research challenges for enabling scientific knowledge discovery at extreme-scale concurrency
Widening gap between FLOPs and I/O capacity
– will make full-resolution, I/O-intensive post hoc analysis prohibitively expensive, if not impossible.

**OUR APPROACH**

Data model – to pass data between Simulation & Analysis
API – for instrumenting simulation and analysis codes
SENSEI In Situ Architecture

“write once, run everywhere”

A simulation can use any back-end through a single API

SENSEI’s analysis adaptors provide the API for simulations to drive analysis and visualization

SENSEI’s data adaptor API and data model expose simulation data structures to the back-end

Bridge/instrumentation code is added to call SENSEI Analysis. Typically: Initialize, Execute, Finalize

1. initialize sim
2. if do_insitu bridge::initialize
3. do
4. compute new state
5. if do_io write plot file
6. if do_insitu bridge::execute
7. while !done
8. if do_insitu bridge::finalize
9. finalize sim
Using the configurable analysis adaptor

- Enable analysis in .xml file
- Run instrumented simulation

```xml
<sensei>
  <!-- Custom Analyses -->
  <analysis type="histogram" mesh="atoms" array="type" association="point"
           bins="10" enabled="0" />

  <analysis type="histogram" mesh="atoms" array="id" association="point"
           bins="10" enabled="1" />

  <!-- Available with ENABLE_VTK_IO -->
  <analysis type="PosthocIO" mode="paraview" output_dir="/vtkio" enabled="0">
    <mesh name="atoms">
      <point_arrays> type, id </point_arrays>
    </mesh>
  </analysis>
</sensei>
```
Catalyst example

• Load a representative dataset in ParaView
• Define your visualization pipeline
• Export Catalyst Python script
Catalyst example

- Configure XML file
- Run instrumented simulation
- Result: one .png image per simulation timestep

```xml
<sensei>
  <!-- Available with ENABLE_CATALYST -->
  <analysis type="catalyst" pipeline="pythonscript"
            filename="gaussianptsbyid.py" enabled="1" />
</sensei>
```
In Situ Visualization and Analysis with Ascent

Slides courtesy:

Hank Childs, University of Oregon
Matthew Larsen, Lawrence Livermore National Laboratory
Cyrus Harrison, Lawrence Livermore National Laboratory
Kenneth Moreland, Sandia National Laboratories
David Rogers, Los Alamos National Laboratory

2019 Exascale Computing Project Annual Meeting

HELD AT
Royal Sonesta Houston Galleria
Houston, Texas

January 14–17, 2019 – ECP Annual Meeting
January 14–18, 2019 - Individual or Group Meetings
Ascent focuses on ease of use and efficient in-situ execution

Ascent Delivers
An easy to use API
– Designed to enable three use cases
  • Making Pictures
  • Transforming Data
  • Capturing Data
– Leverages Conduit (http://software.llnl.gov/conduit)
  • Underpins support for C, C++, Fortran, and Python
  • Simplifies handoff of mesh-based simulation data
  • Convention for specifying data called “Blueprint”

A flyweight design
– Efficient distributed-memory + many-core execution
  • Leverages MPI, VTK-m (http://m.vtk.org/)
– Lower memory requirements then current tools
– Less dependencies than current tools (ex: no OpenGL)
Ascent is ready for common visualization use cases

- Iso-Volume
- Threshold
- Slice
- Contour

- Clips
- Rendering
- Pseudocolor
- Volume
- Mesh
Design Goal: Support custom analysis as a first class citizen

Mainstream visualization only gets you so far
– Scientists often want something other than a contour

In-situ visualization frameworks need to be
– Flexible
– Easy to use
– Easy to connect with other “things”
Ascent supports multiple languages and output types

**Language Bindings**
- C/C++
- Python
- Fortran

**Output Types**
- HDF
- ADIOS
- CINEMA
Ascent provides example integrations that also serve as built-in data sources

Cloverleaf3D  Lulesh  Kripke  Smooth Noise
Additional Resources
SENSEI additional resources

- SENSEI project page
  https://sensei-insitu.org/

- Gitlab repo
  https://gitlab.kitware.com/sensei/sensei

- SENSEI SC18 Tutorial - Slides and Virtual Machine
  https://sensei-insitu.org/tutorials/sc18.html
Ascent additional resources

Website + Docs:
http://ascent-dav.org

GitHub Repo:
https://github.com/Alpine-DAV/ascent

Email Help:
help@ascent-dav.org

Example in-situ rendering created using Ascent
QUESTIONS?

Joe Insley
insley@anl.gov

Silvio Rizzi
srizzi@anl.gov