Argonne Training Program on Extreme-Scale Computing (ATPESC)

Data Analysis and Visualization
<table>
<thead>
<tr>
<th>Time</th>
<th>Title of presentation</th>
<th>Lecturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30 am</td>
<td>Data Analysis and Visualization Introduction</td>
<td>Mike Papka <em>ANL</em>, Joe Insley <em>ANL/NIU</em>, Silvio Rizzi <em>ANL</em>, Janet Knowles, <em>ANL</em></td>
</tr>
<tr>
<td>9:15 am</td>
<td>Scalable Molecular Visualization and Analysis Tools in VMD</td>
<td>Mariano Spivak <em>UIUC</em></td>
</tr>
<tr>
<td>10:00 am</td>
<td>Break</td>
<td></td>
</tr>
<tr>
<td>10:30 am</td>
<td>Large Scale Visualization with ParaView</td>
<td>Dan Lipsa <em>Kitware</em></td>
</tr>
<tr>
<td>12:00 pm</td>
<td>Visualization and Analysis of HPC Simulation Data with VisIt</td>
<td>Cyrus Harrison <em>LLNL</em></td>
</tr>
<tr>
<td>12:30 pm</td>
<td>Lunch</td>
<td></td>
</tr>
<tr>
<td>1:30 pm</td>
<td>Visualization and Analysis of HPC Simulation Data with VisIt</td>
<td>Cyrus Harrison <em>LLNL</em></td>
</tr>
<tr>
<td>2:30 pm</td>
<td>Vapor</td>
<td>Scott Pearse <em>UCAR</em></td>
</tr>
<tr>
<td>3:30 pm</td>
<td>Break</td>
<td></td>
</tr>
<tr>
<td>4:00 pm</td>
<td>Exploring Visualization with Jupyter Notebooks</td>
<td>• David Koop <em>NIU</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Cyrus Harrison <em>LLNL</em></td>
</tr>
<tr>
<td>5:30 pm</td>
<td>Hands-on</td>
<td>All</td>
</tr>
<tr>
<td>6:30 pm</td>
<td>Dinner</td>
<td></td>
</tr>
<tr>
<td>7:30 pm</td>
<td>After-dinner talk: How learning about GPUs actually made me good at computational science</td>
<td>Max Katz <em>NVIDIA</em></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
Physics: Stellar Radiation

Data courtesy of: Lars Bildsten and Yan-Fei Jiang, University of California at Santa Barbara
ARTERIAL BLOOD FLOW

Data courtesy of: Amanda Randles, Duke University
Physics: Magnetic Confinement Fusion

Data courtesy of Sean Dettrick, TAE Technologies, Inc.
Engineering Technologies: Combustion

Data courtesy of: Saumil Patel, Muhsin Ameen, Sicong Wu, Argonne National Laboratory; Tanmoy Chatterjee, GE Global Research
Materials Science / Molecular

Data courtesy of: Jeff Greeley, Nichols Romero, Argonne National Laboratory

Data courtesy of: Paul Kent, Oak Ridge National Laboratory, Anouar Benali, 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 Theta, Eagle, and Grand file systems
Visualization Tools and Data Formats
All Sorts of Tools

Visualization Applications
– VisIt *
– ParaView *
– EnSight

Domain Specific
– VMD, PyMol, Ovito, Vapor

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 Type</th>
<th>File Format</th>
<th>Example Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTK</td>
<td>PLOT3D</td>
<td>Facet</td>
</tr>
<tr>
<td>Parallel (partitioned) VTK</td>
<td>SpyPlot CTH</td>
<td>Tetrad</td>
</tr>
<tr>
<td>VTK MultiBlock</td>
<td>HDF5 raw image data</td>
<td>PNG</td>
</tr>
<tr>
<td>(MultiGroup, Hierarchical,</td>
<td>DEM</td>
<td>SAF</td>
</tr>
<tr>
<td>Hierarchical Box)</td>
<td>VRML</td>
<td>LS-Dyna</td>
</tr>
<tr>
<td>Legacy VTK</td>
<td>PLY</td>
<td>Nek5000</td>
</tr>
<tr>
<td>Parallel (partitioned) legacy VTK</td>
<td>Polygonal Protein Data Bank</td>
<td>OVERFLOW</td>
</tr>
<tr>
<td>EnSight files</td>
<td>XMed Molecule</td>
<td>paraDIS</td>
</tr>
<tr>
<td>EnSight Master Server</td>
<td>Stereo Lithography</td>
<td>PATRAN</td>
</tr>
<tr>
<td>Exodus</td>
<td>Gaussian Cube</td>
<td>PFLOTRAN</td>
</tr>
<tr>
<td>BYU</td>
<td>Raw (binary)</td>
<td>Pixie</td>
</tr>
<tr>
<td>XDMF</td>
<td>AVS</td>
<td>PuReMD</td>
</tr>
<tr>
<td>PLOT2D</td>
<td>Meta Image</td>
<td>S3D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SAS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Data Representations
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: Volume Rendering
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: 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: 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
Data Representations: Pathlines

From vector field on a mesh (needs connectivity)
– Trace the path an element will travel over 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
Advanced Rendering
Intel® oneAPI Rendering Toolkit ("Render Kit"/"Render Framework")
Open Source Software for Advanced Rendering and Visualization

Stellar Radiation
Lars Bildsten (UC Santa Barbara),
Vis: Joe Insley (ANL), Greg Johnson (Intel)

51 M particle “Last Journey” HACC,
Katrin Heitmann (ANL) and Tom Uram (ANL).
Vis: Aaron Knoll (Intel)

Harvey Proxy Simulation
Amanda Randles (Duke University)
Vis: Silvio Rizzi (ANL), David DeMarle (Intel)

Tumor cell simulation,
Jifu Tan and Michael Hood, (NIU),
Vis: Joe Insley, Silvio Rizzi,
Michael Papka and Janet Knowles,
(ANL)

Kohn-Kuriyama simulation
Jifu Tan and Michael Hood, (NIU),
Vis: Joe Insley, Silvio Rizzi,
Michael Papka and Janet Knowles,
(ANL)

Slide courtesy of Intel Visualization Team
Intel® oneAPI Rendering Toolkit ("Render Kit")
Open Source Software for Advanced Rendering and Visualization

Intel® OSPRay Studio
Pro-vis frontend and scene graph
https://github.com/ospray/ospray_studio

Application
ParaView, VisIt, VMD, etc.

Intel® OSPRay
Scalable rendering engine, API and SDK
Distributed MPI Rendering via OSPRay MPI
Intel implementation of the Khronos ANARI specification (https://www.khronos.org/anari)
http://www.ospray.org

Intel® Open VKL
API for volume sampling, traversal, interpolation and classification
http://www.openvkl.org

Intel® Embree
Optimized geometry ray tracing kernels
- BVH builders, traversal and intersection
http://www.embree.org

Intel® Open Image Denoise
AI / DL – based denoising of sampling artifacts from path tracing
http://www.openimagedenoise.org

Slide courtesy of Intel Visualization Team
In Situ Visualization and Analysis
Five orders of magnitude between compute and I/O capacity on Titan Cray system at ORNL

- Computation: 125 PB/s
- Node memory: 4.5 PB/s
- Interconnect: 24 TB/s
- Storage: 1.4 TB/s

Image courtesy Ken Moreland
What are the problems?

- Not enough I/O capacity on current HPC systems, and the trend is getting worse.
- If there’s not enough I/O, you can’t write data to storage, so you can’t analyze it: lost science.
- Energy consumption: it costs a lot of power to write data to disk.
- Opportunity for doing better science (analysis) when have access to full spatiotemporal resolution data.

Slide courtesy the SENSEI team [www.sensei-insitu.org](http://www.sensei-insitu.org)
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Contact person at ATPESC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPINE</td>
<td>In Situ algorithms and infrastructure for the Exascale Computing Project</td>
<td>Silvio Rizzi, Cyrus Harrison</td>
</tr>
<tr>
<td>ASCENT</td>
<td>A flyweight in situ visualization and analysis runtime for multi-physics HPC</td>
<td>Cyrus Harrison</td>
</tr>
<tr>
<td>SENSEI</td>
<td>Write once run anywhere. Multiple backends. MxN in transit communication</td>
<td>Silvio Rizzi, Joe Insley</td>
</tr>
<tr>
<td></td>
<td>patterns</td>
<td></td>
</tr>
<tr>
<td>ParaView/Catalyst</td>
<td><em>In situ</em> use case library, with an adaptable application programming</td>
<td>Dan Lipsa</td>
</tr>
<tr>
<td></td>
<td>interface (API), that orchestrates the delicate alliance between simulation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and analysis and/or visualization tasks</td>
<td></td>
</tr>
<tr>
<td>Libsim</td>
<td>Originally developed to facilitate interactive connections from VisIt to</td>
<td>Cyrus Harrison</td>
</tr>
<tr>
<td></td>
<td>running simulations</td>
<td></td>
</tr>
<tr>
<td>SmartSim</td>
<td>SmartSim is a software framework that facilitates the convergence of</td>
<td>Silvio Rizzi</td>
</tr>
<tr>
<td></td>
<td>numerical simulations and AI workloads on heterogeneous architectures</td>
<td></td>
</tr>
</tbody>
</table>
Exascale Computing Project
Software Technology Data and Visualization

Apps

In situ Infrastructure
ALPINE
ASCENT
Visit Libsim
ParaView Catalyst

In situ Algorithms
New Algorithms
Traditional Algorithms
Compression
ZFP

Output/Artifacts
Cinema

Post Processing
ParaView
Visit

Slide courtesy of the ECP ALPINE project
Ascent

- Flyweight design, minimizes dependencies
- Data model based on Conduit from LLNL
- Vis and analysis algorithms implemented in VTK-m

///
/// Run Ascent
///
Ascent ascent;
ascent.open();
ascent.publish(data);
ascent.execute(actions);
ascent.close();

Slide courtesy of the Ascent team
SENSEI: Write once run everywhere

- “Write once, run everywhere” design
- Data model based on VTK from Kitware
- Supports a variety of backends, including ParaView/Catalyst, VisIt/LibSim, ADIOS, Python
- MxN in transit capabilities
SENSEI in transit demo

- Containerized workflow

- Producer:
  - LAMMPS molecular dynamics simulation.
  - 16 ranks on ThetaGPU

- Consumer:
  - SENSEI endpoint with Catalyst backend.
  - 4 ranks on Cooley.

- ADIOS2 used as transport

- Container recipes and config files available on Zenodo
The SmartSim open-source library enables scientists, engineers, and researchers to embrace a “data-in-motion” philosophy to accelerate the convergence of AI/data science techniques and HPC simulations.

SmartSim enables simulations to be used as engines within a system, producing data, consumed by other services enable new applications:

- Embed machine learning training and inference with existing in Fortran/C/C++ simulations
- Communicate data between C, C++, Fortran, and Python applications
- Analyze and visualize data streamed from HPC applications while they are running
- Launch, configure, and coordinate complex simulation, analysis, and visualization workflows

All of these can be done without touching the filesystem, i.e. data-in-motion.
Infrastructures
VTK-m’s main thrust: a write-once-run-everywhere framework

Contour | Streams | Clip | Render
---|---|---|---
Surface | Normals | Ghost Cells | Warp | ...

VTK-m

x86 | CUDA | Xeon Phi | Radeon | Xe

Demonstrated | Upcoming

Slide courtesy of the ECP VTK-m project
What is Cinema?

- **Cinema** is part of an integrated workflow, providing a method of extracting, saving, analyzing or modifying and viewing complex data artifacts from large scale simulations.
  - If you’re having difficulty exploring the complex results from your simulation, Cinema can help.

- **The Cinema ‘Ecosystem’** is an integrated set of writers, viewers, and algorithms that allow scientists to export, analyze/modify and view Cinema databases.
  - This ecosystem is embodied in widely used tools (*ParaView, VisIt, Ascent*) and the database specification.
Additional Resources
Visualization Help
support@alcf.anl.gov

Publication Images & Covers

Animations

- SC Visualization Showcase [Best Vis Finalist 2014-2020]
- APS Division of Fluid Dynamics Gallery of Fluid Motion
- SC Gordon Bell Submissions
- Press Releases

InSitu Vis and Analysis
Additional information

ALPINE: https://alpine.dsscale.org/
Ascent: https://github.com/Alpine-DAV/ascent
SENSEI: https://sensei-insitu.org/
SmartSim: https://developer.hpe.com/platform/smartsim/home/
ParaView/Catalyst: https://www.paraview.org/in-situ/
Libsim: https://www.visitusers.org/index.php?title=VisIt-tutorial-in-situ
VTK-m: https://m.vtk.org/
Cinema: https://cinemascience.github.io/
OSPRay: https://github.com/ospray/ospray
QUESTIONS?

Joe Insley
insley@anl.gov

Silvio Rizzi
srizzi@anl.gov

Janet Knowles
jknowles@anl.gov

Victor Mateevitsi
vmateevitsi@anl.gov