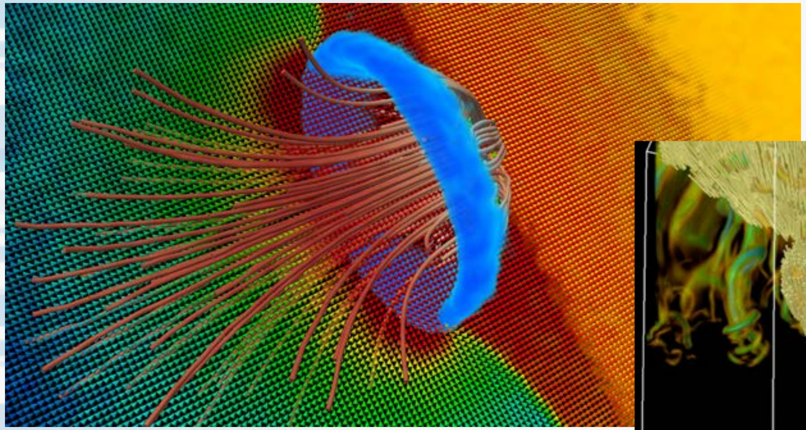
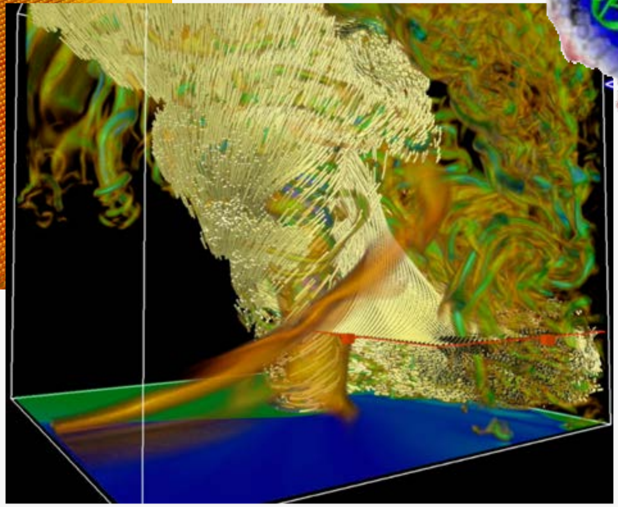
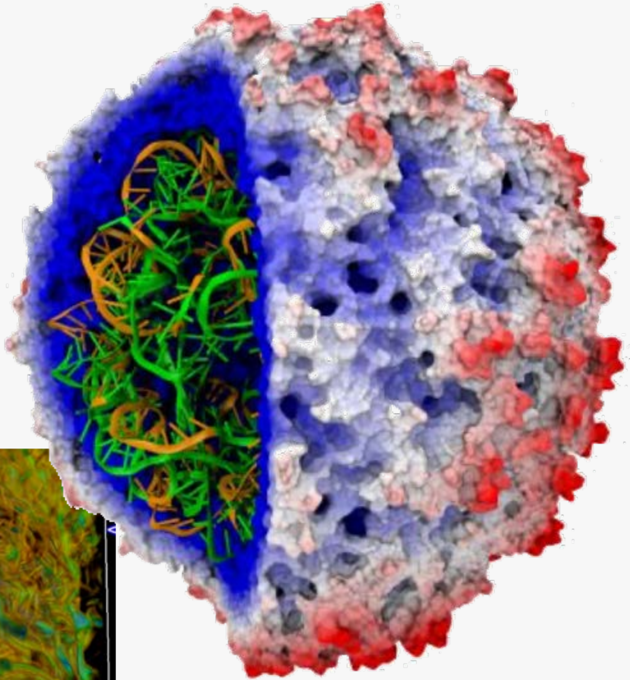
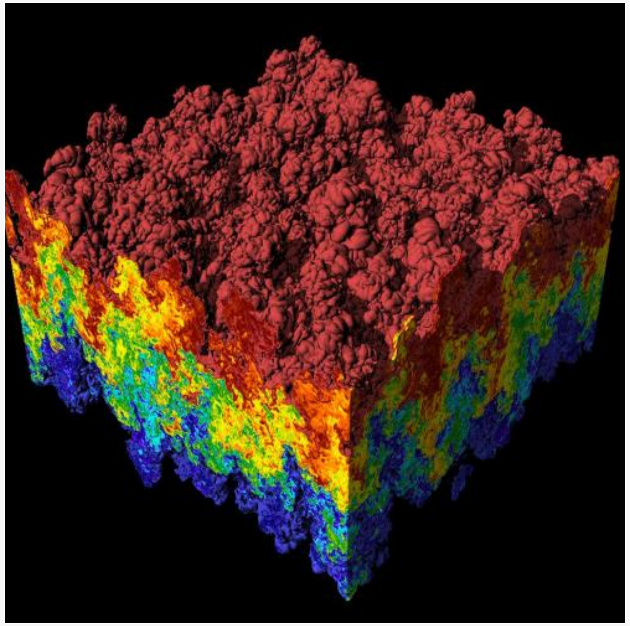
Argonne Training Program on Extreme-Scale Computing (ATPESC)

Data Analysis and Visualization









ECP
EXASCALE COMPUTING PROJECT

U.S. DEPARTMENT OF ENERGY | Office of Science

NNSA National Nuclear Security Administration

Visualization & Data Analysis

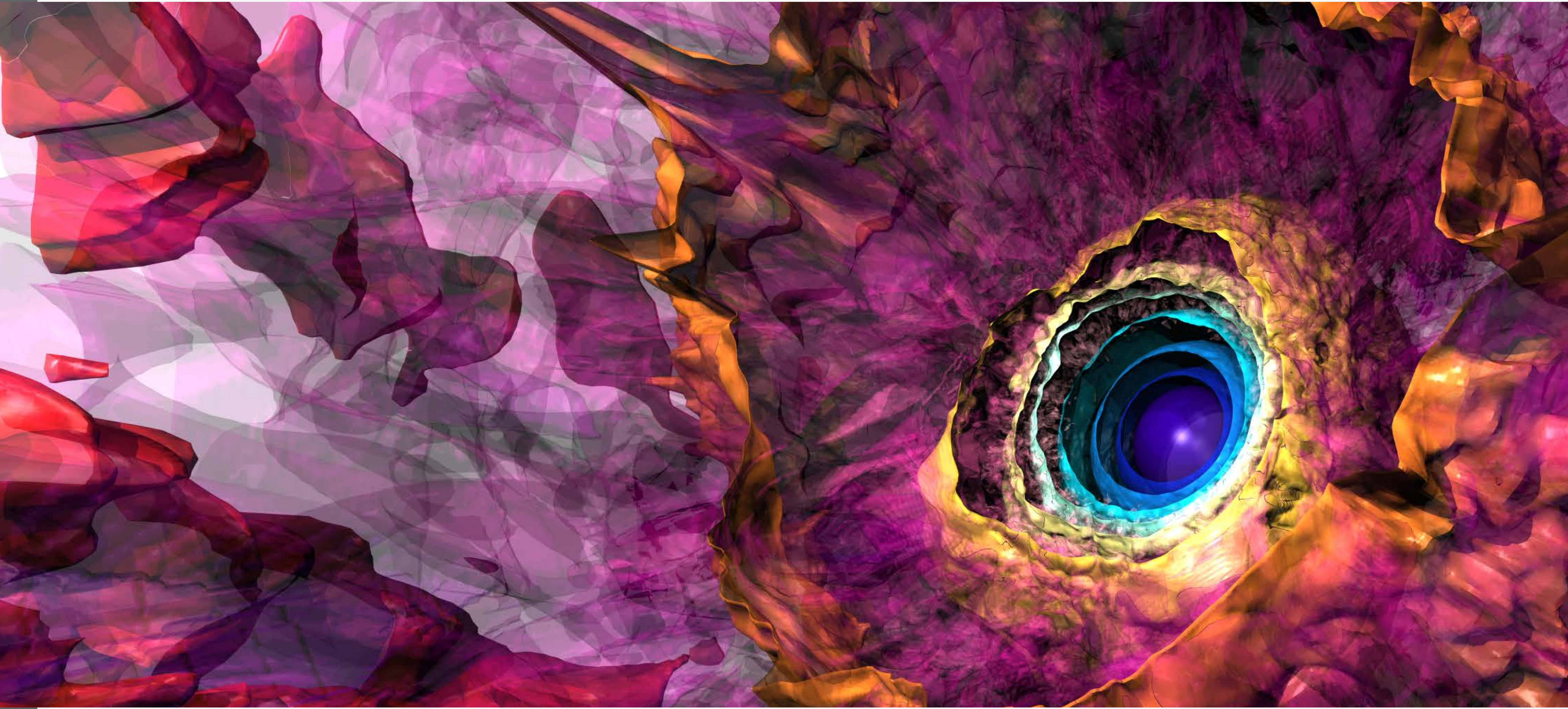
Time	Title of presentation	Lecturer
8:30 am	**Data Analysis and Visualization Introduction**	Mike Papka *ANL*, Joe Insley *ANL/NIU*, Silvio Rizzi *ANL*, Janet Knowles, *ANL*
9:15 am	**Scalable Molecular Visualization and Analysis Tools in VMD**	Mariano Spivak *UIUC*
10:00 am	*Break*	
10:30 am	**Large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

Here's the plan…

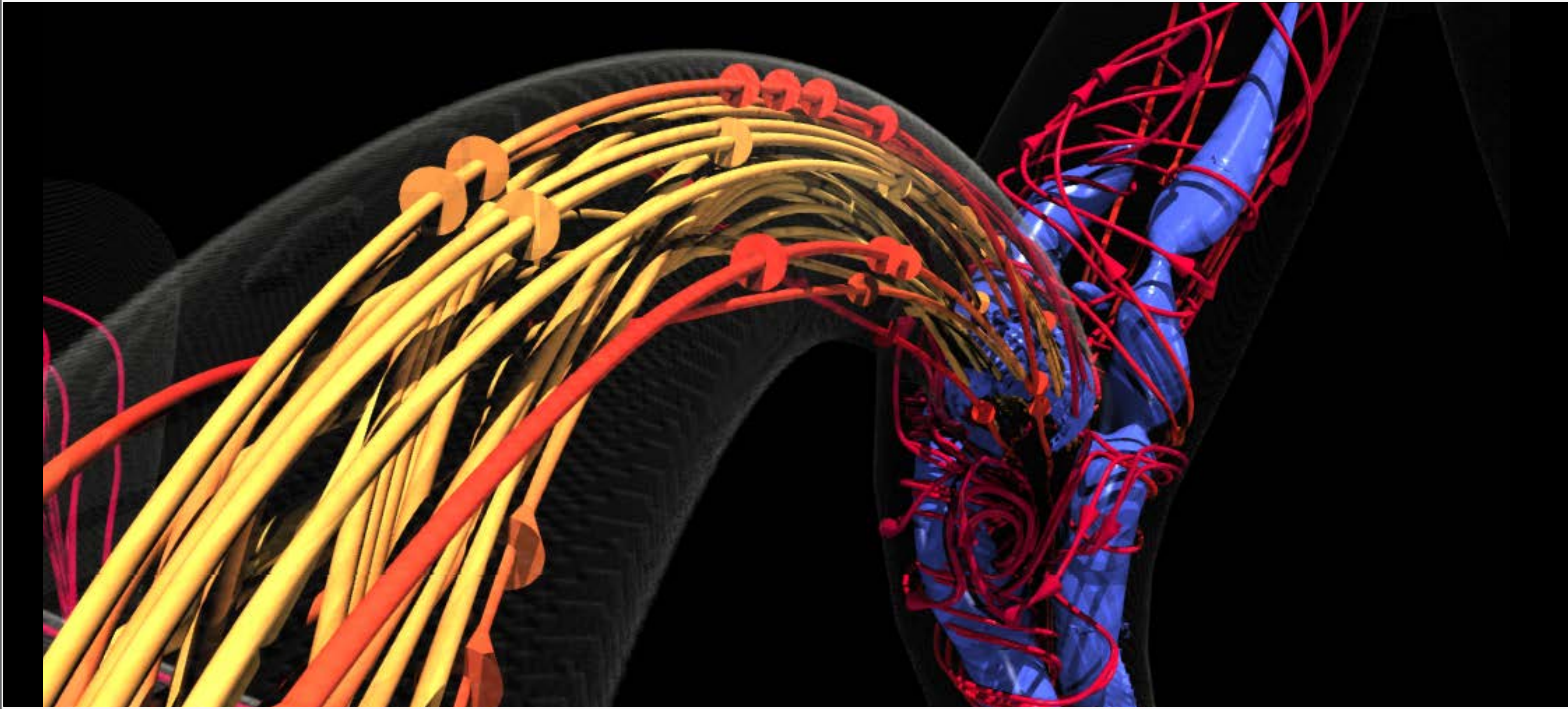
- Examples of visualizations
- Visualization resources
- Visualization tools and formats
- Data representations
- Visualization for debugging
- In Situ Visualization and Analysis

Physics: Stellar Radiation

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

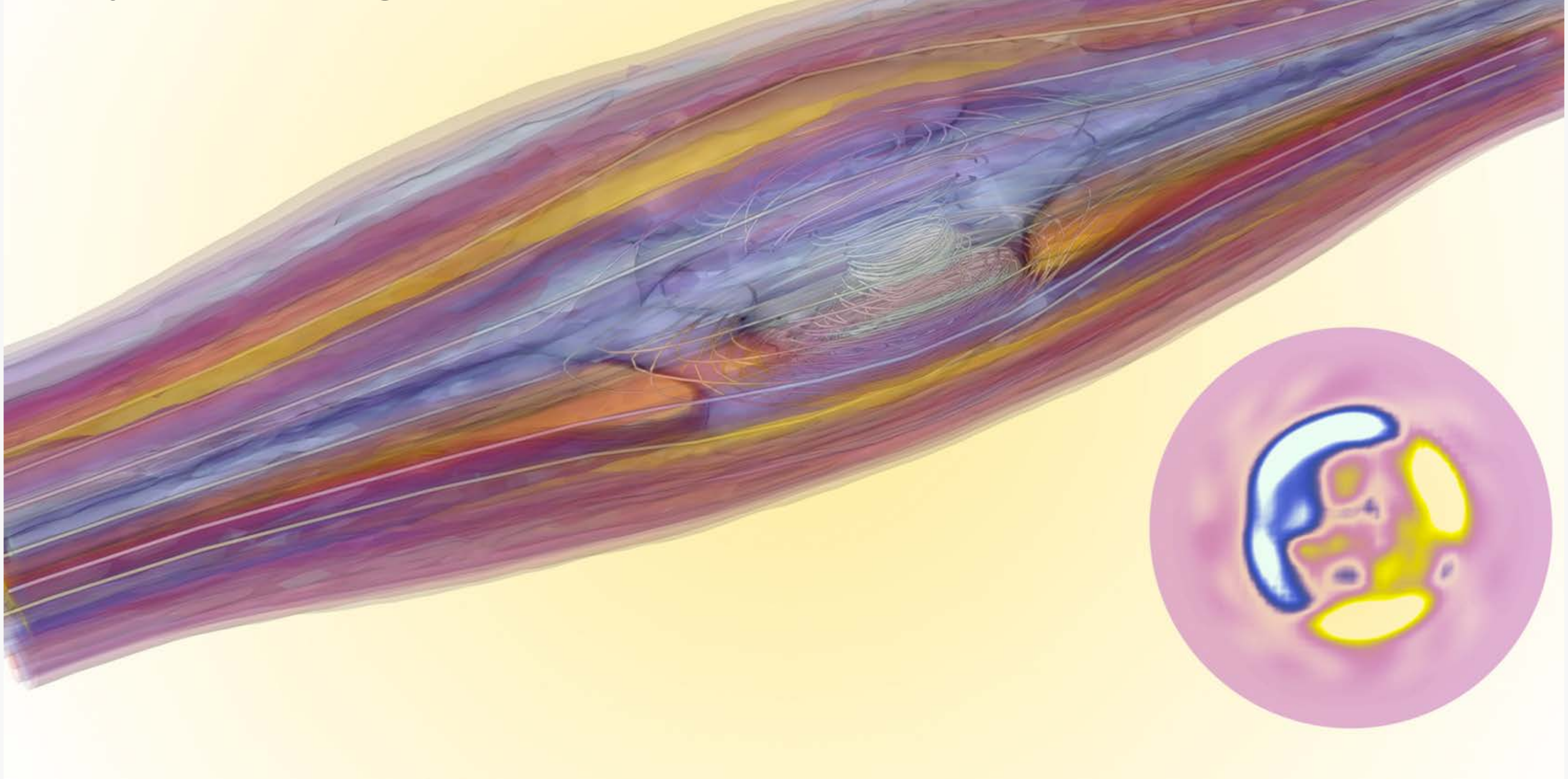


ARTERIAL BLOOD FLOW



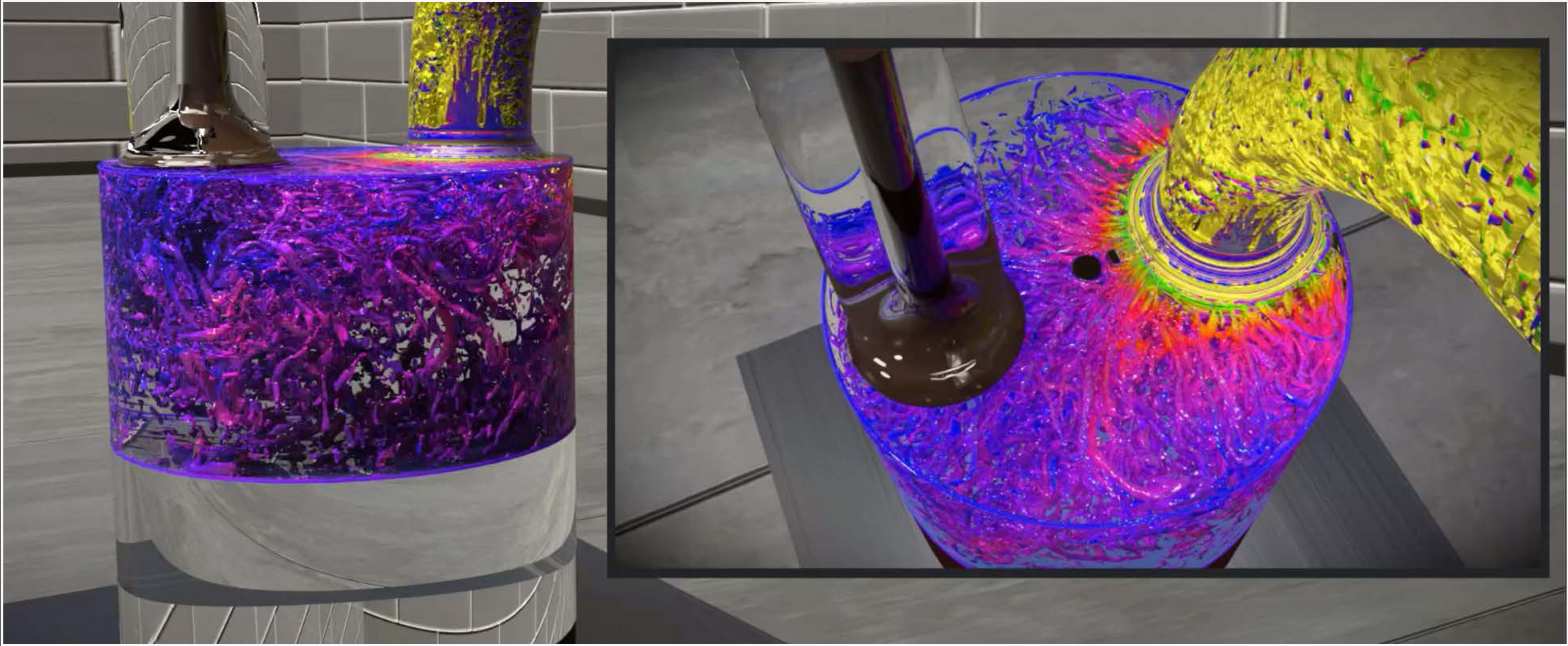
Data courtesy of: Amanda Randles, Duke University

Physics: Magnetic Confinement Fusion



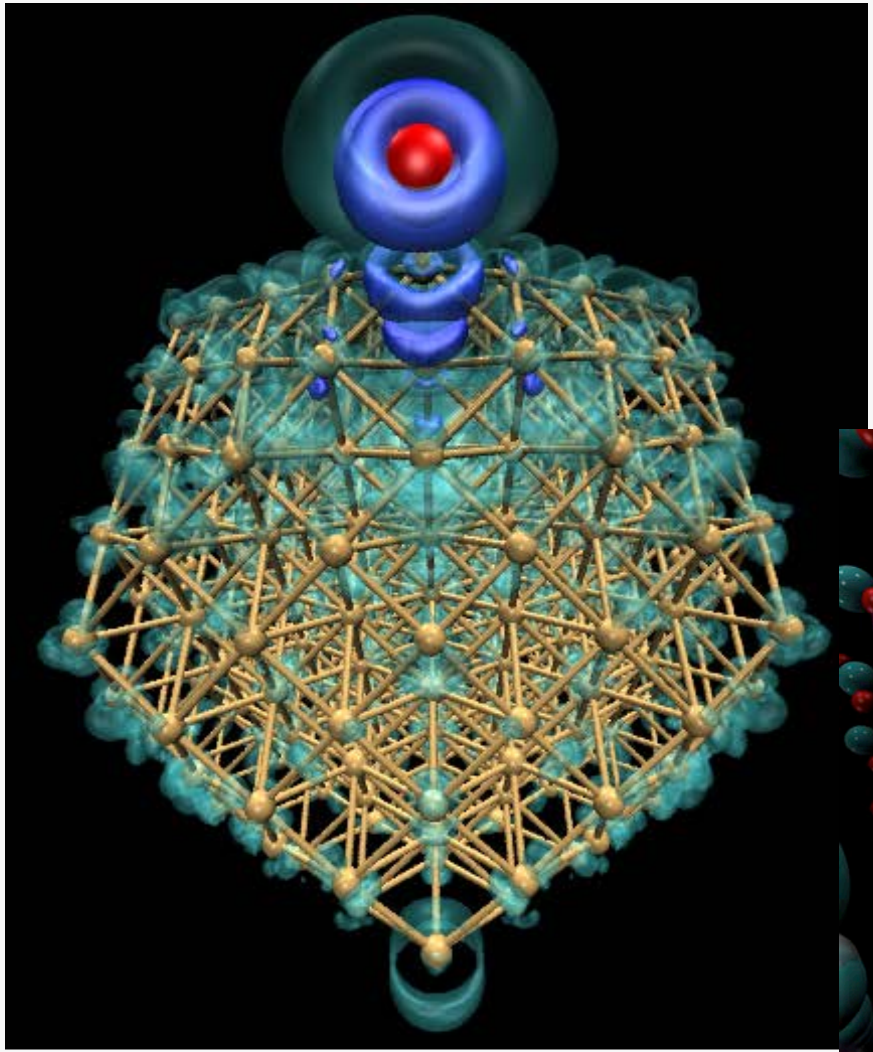
Data courtesy of Sean Dettrick, TAE Technologies, Inc.

Engineering Technologies: Combustion



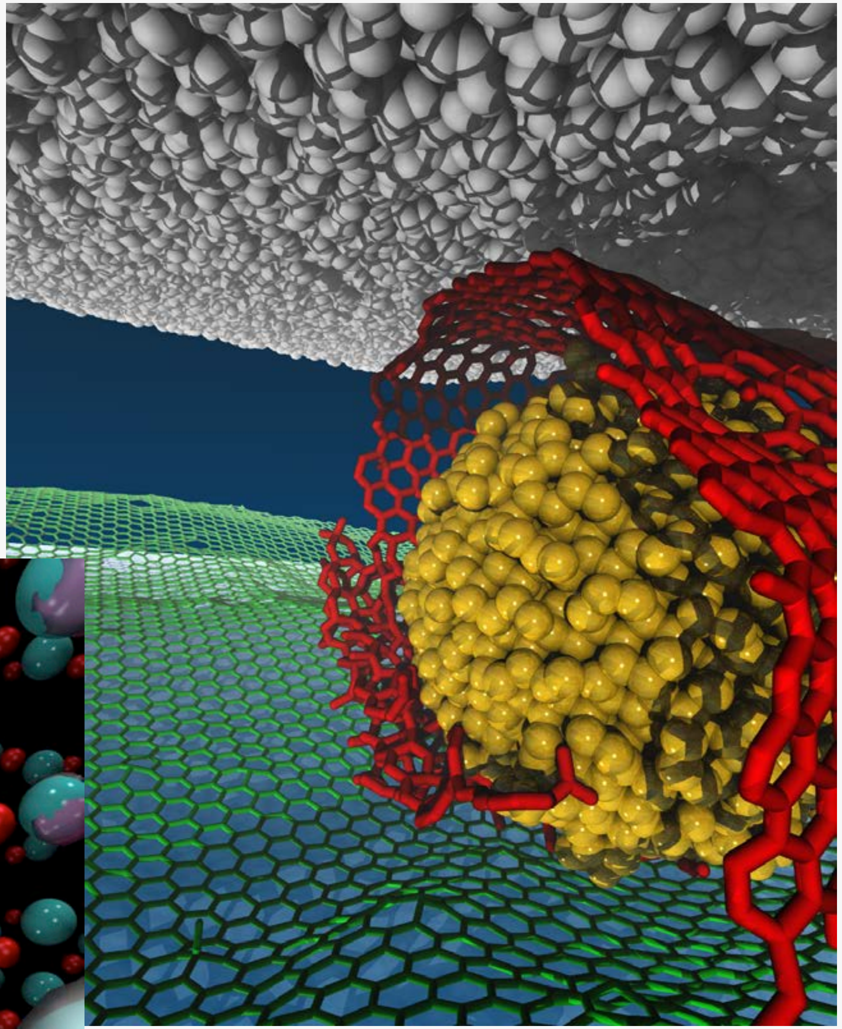
Data courtesy of: Saumil Patel, Muhsin Ameen, Sicong Wu, Argonne National Laboratory; Tanmoy Chatterjee, GE Global Research

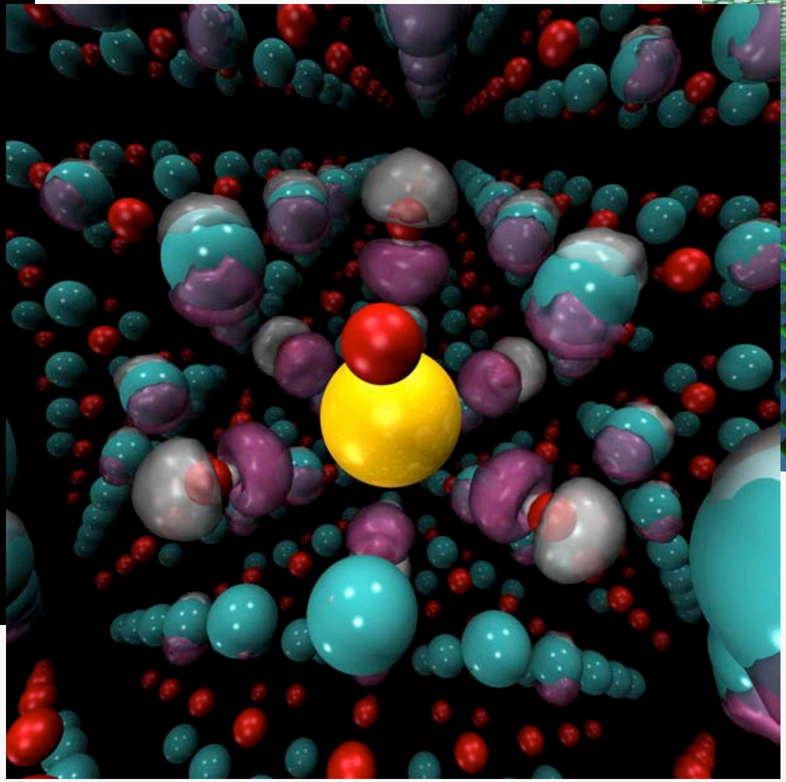
Materials Science / Molecular



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

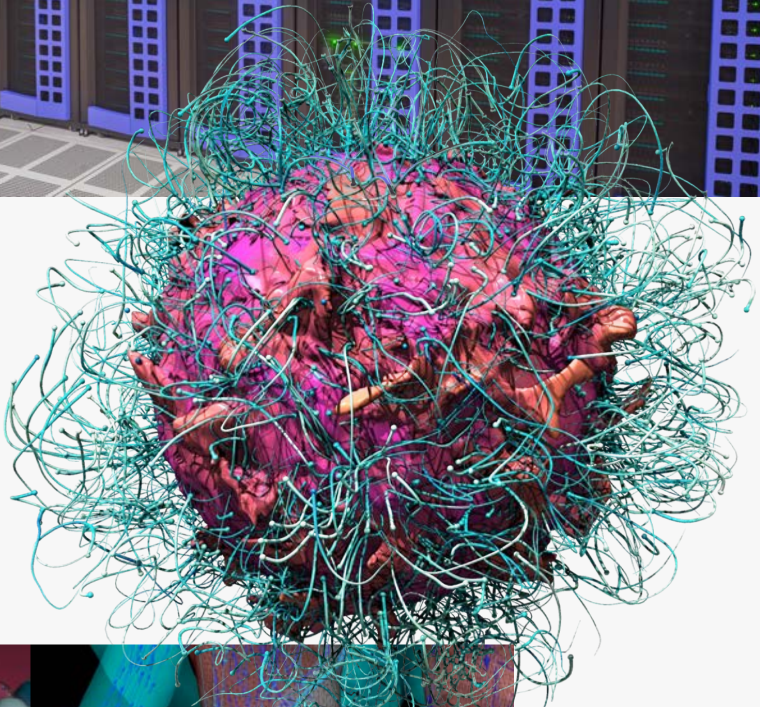
Data courtesy of: Subramanian Sankaranarayanan, Argonne National Laboratory





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

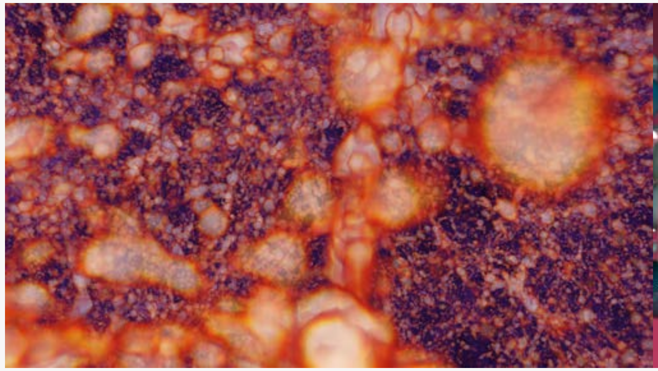
Aggregate RAM of 47 TB

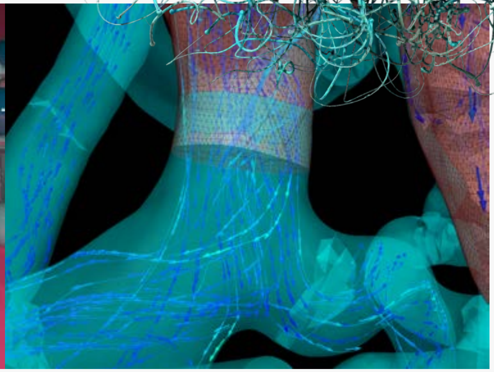
Aggregate GPU memory of ~3TB

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)

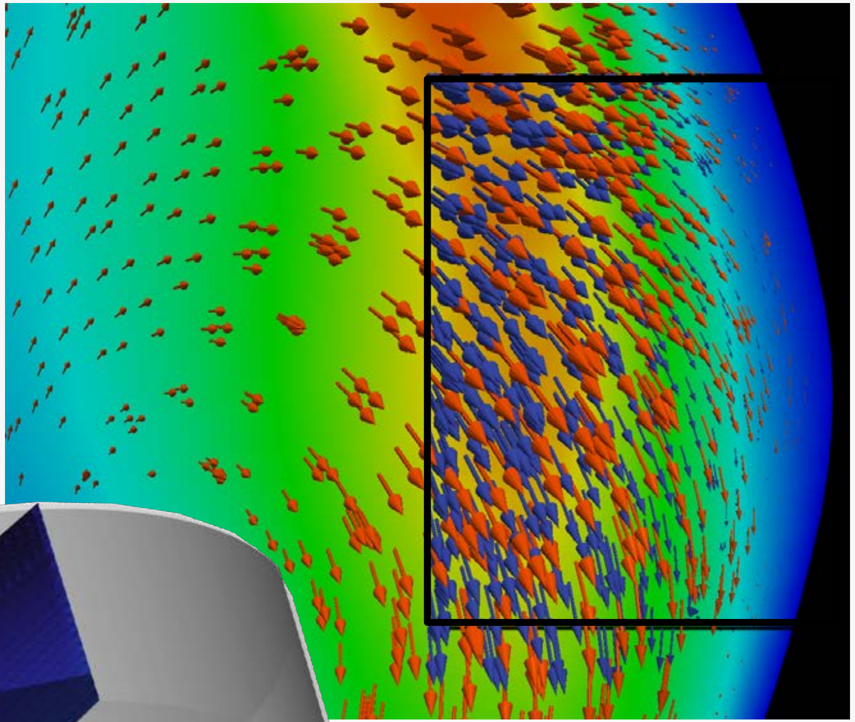
Data File Formats (ParaView & VisIt)

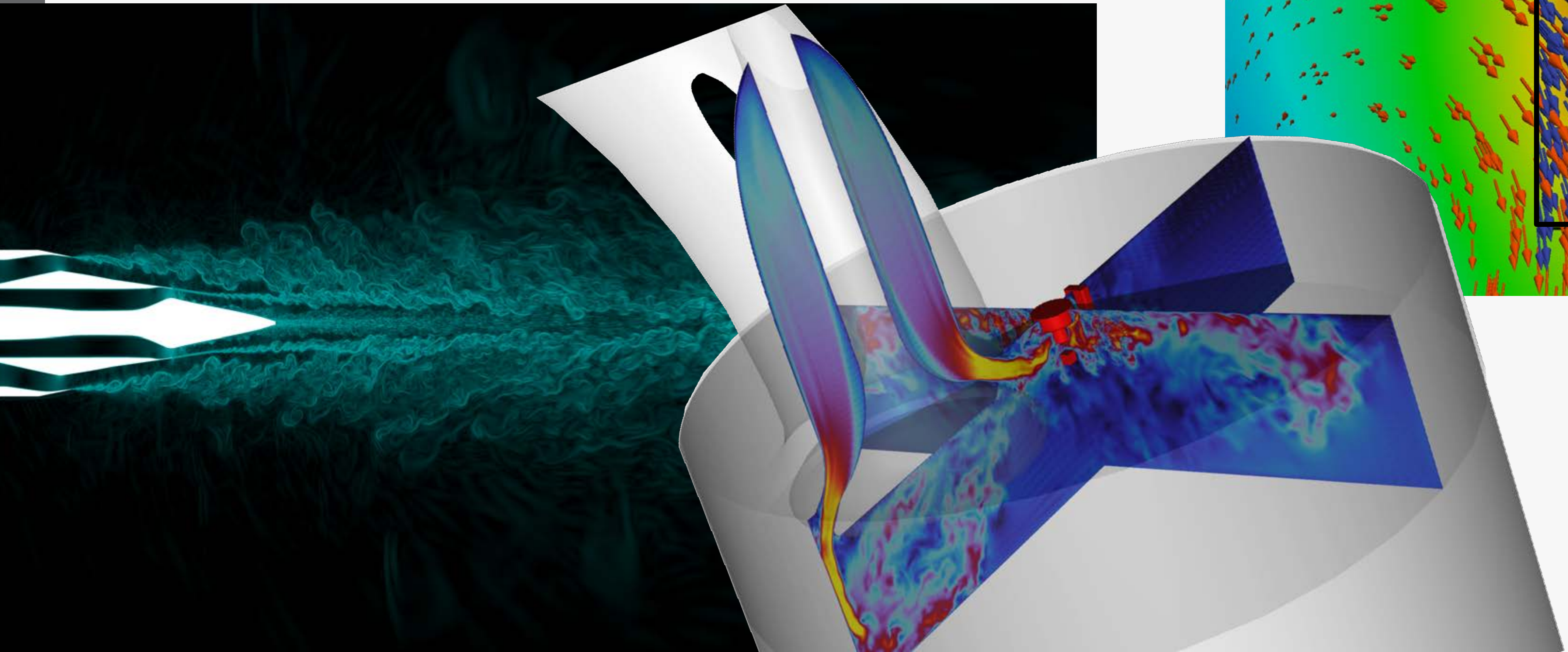
VTK
Parallel (partitioned) VTK
VTK MultiBlock (MultiGroup, Hierarchical, Hierarchical Box)
Legacy VTK
Parallel (partitioned) legacy VTK
EnSight files
EnSight Master Server
Exodus
BYU
XDMF
PLOT2D
PLOT3D
SpyPlot CTH
HDF5 raw image data
DEM
VRML
PLY
Polygonal Protein Data Bank
XMol Molecule
Stereo Lithography
Gaussian Cube
Raw (binary)
AVS
Meta Image
Facet
PNG
SAF
LS-Dyna
Nek5000
OVERFLOW
paraDIS
PATRAN
PFLOTRAN
Pixie
PuReMD
S3D
SAS
Tetrad
UNIC
VASP
ZeusMP
ANALYZE
BOV
GMV
Tecplot
Vis5D
Xmdv
XSF

Data Representations

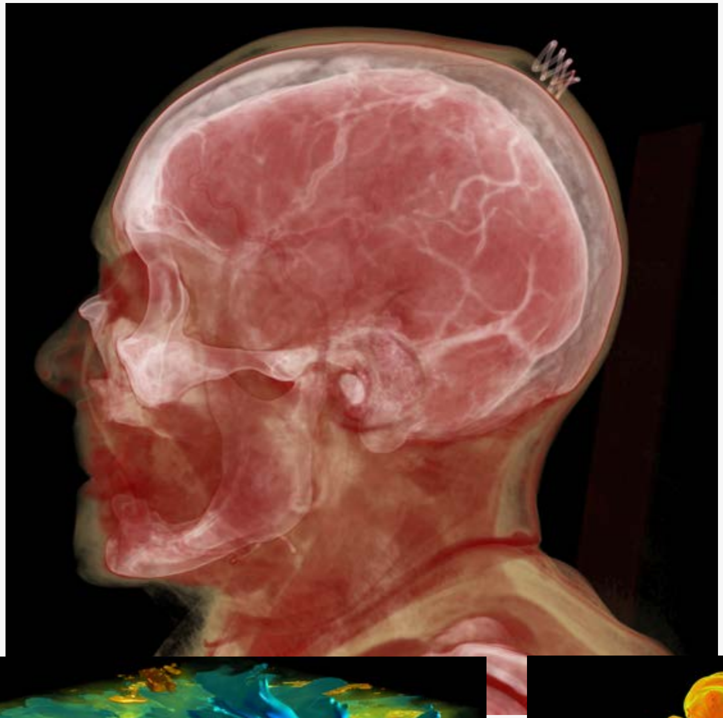
Data Representations: Cutting Planes

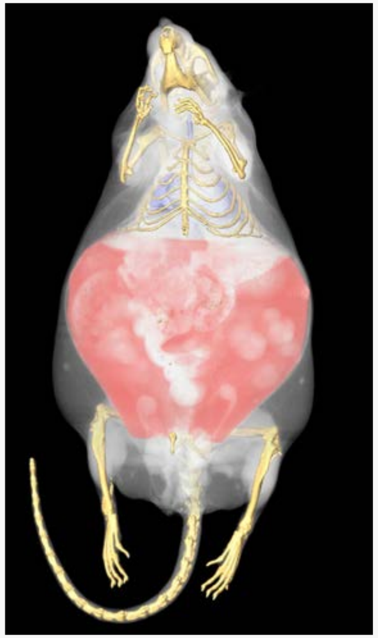
- Slice a plane through the data
  – Can apply additional visualization methods to resulting plane
- VisIt & ParaView & vtk good at this
- VMD has similar capabilities for some data formats

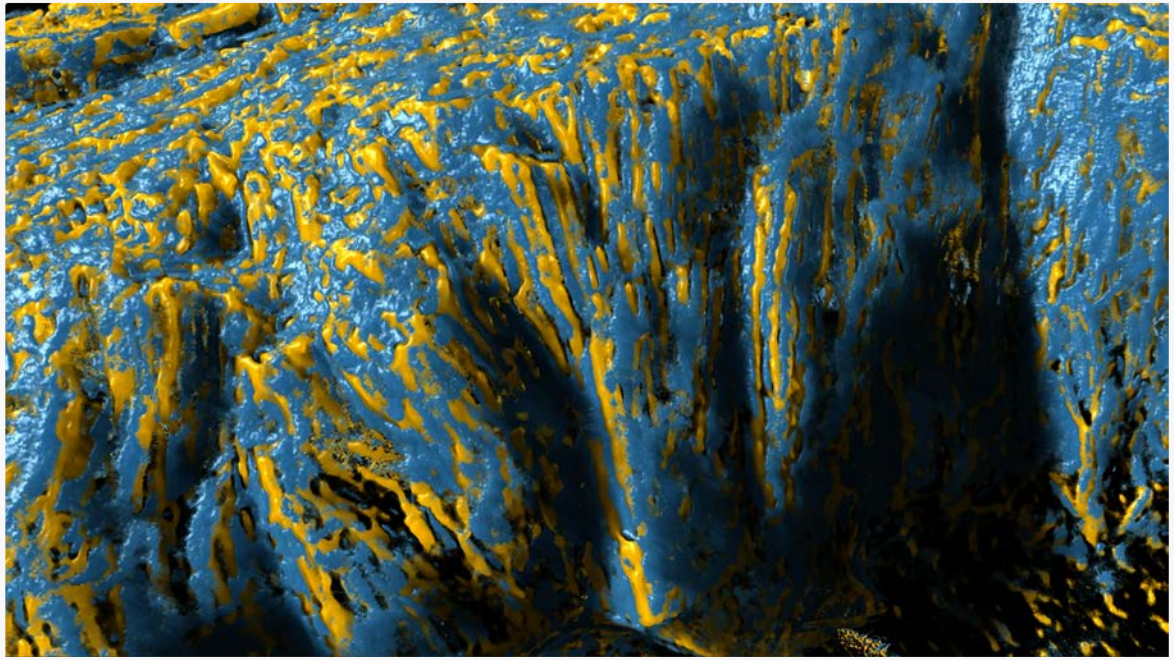


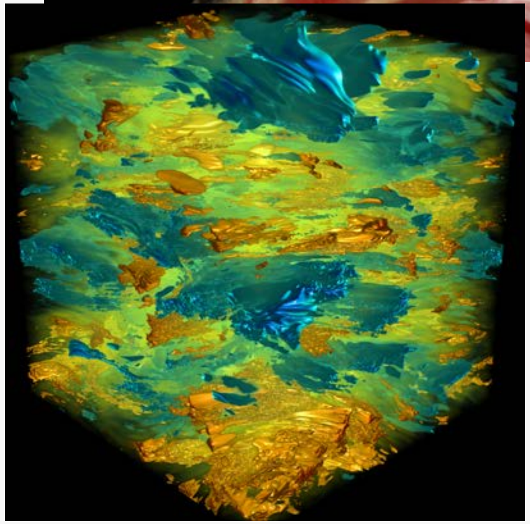


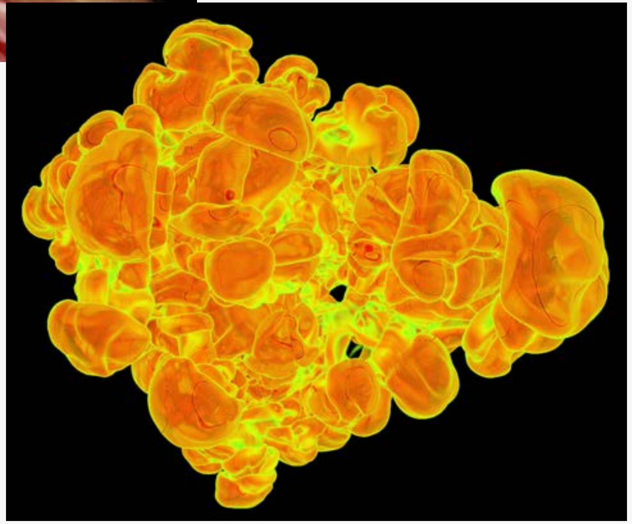
Data Representations: Volume Rendering

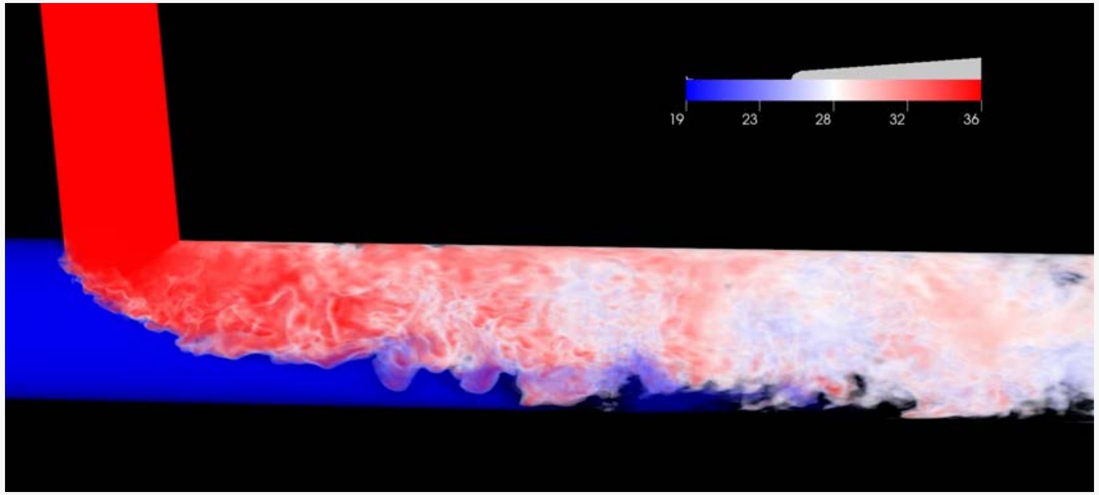












Data Representations: Contours (Isosurfaces)

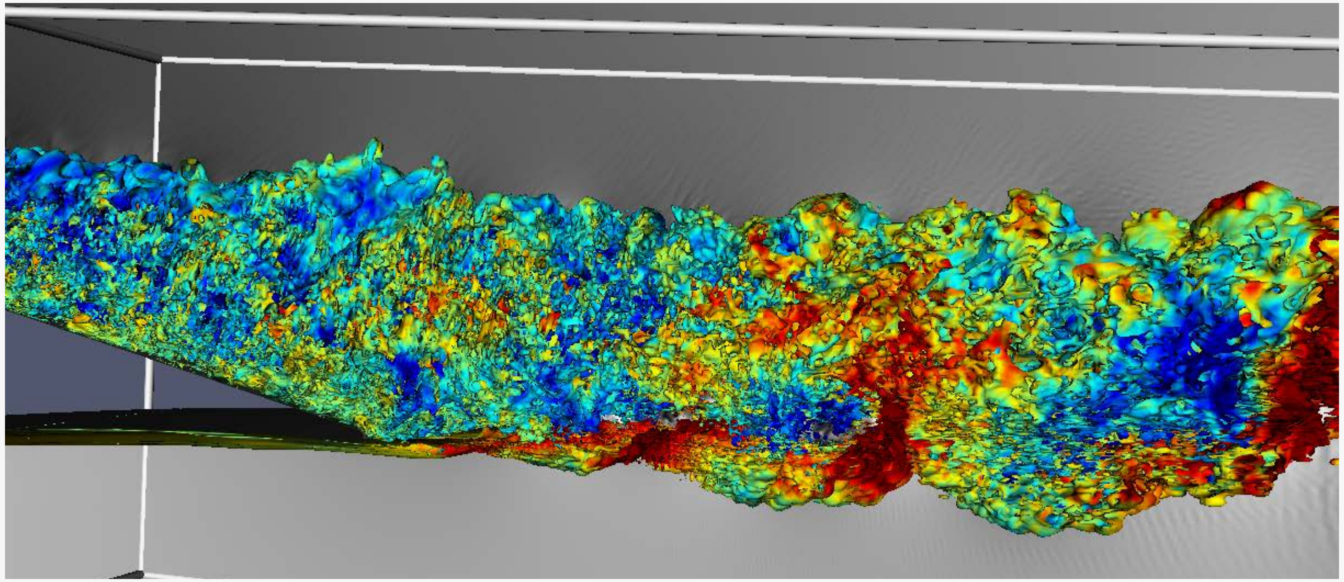
A Line (2D) or Surface (3D), representing a constant value

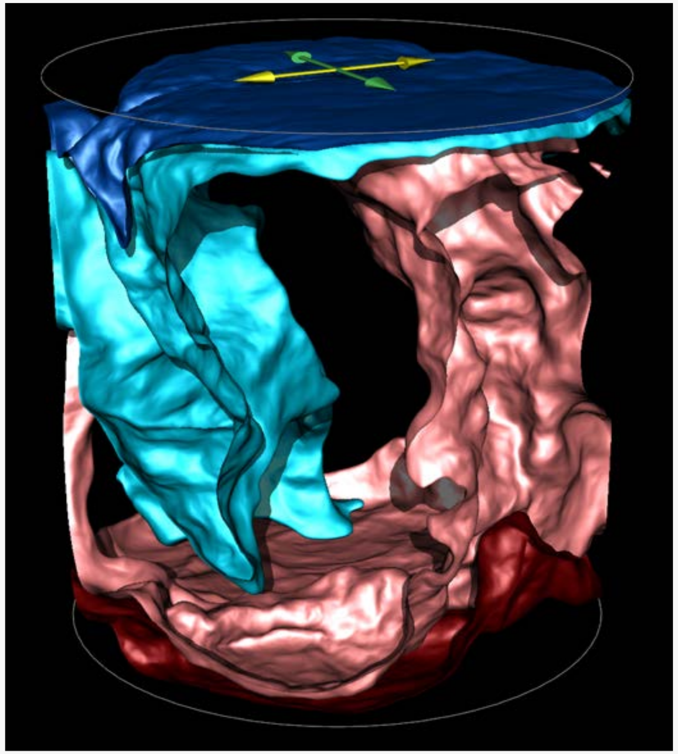
VisIt & ParaView:

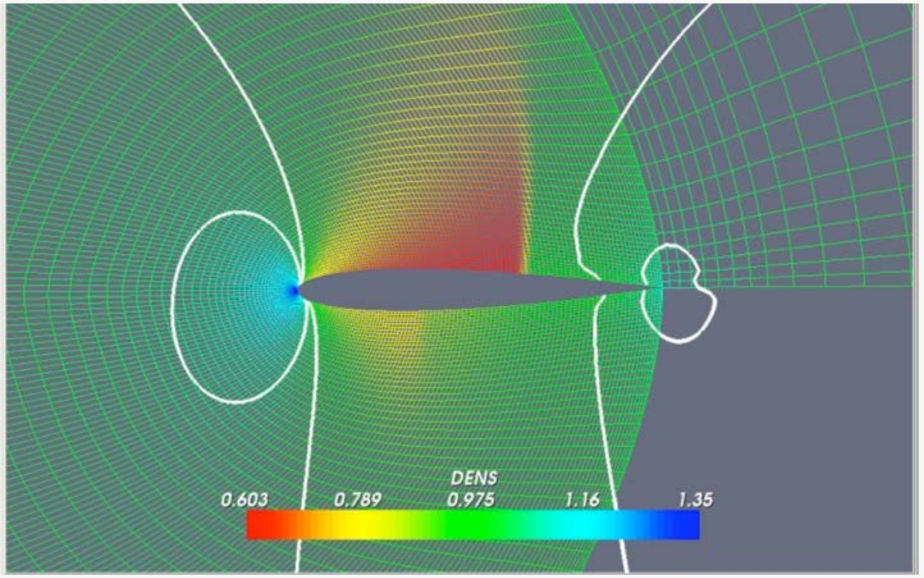
– good at this

vtk:

– same, but again requires more effort







Data Representations: Glyphs

2D or 3D geometric object to represent point data

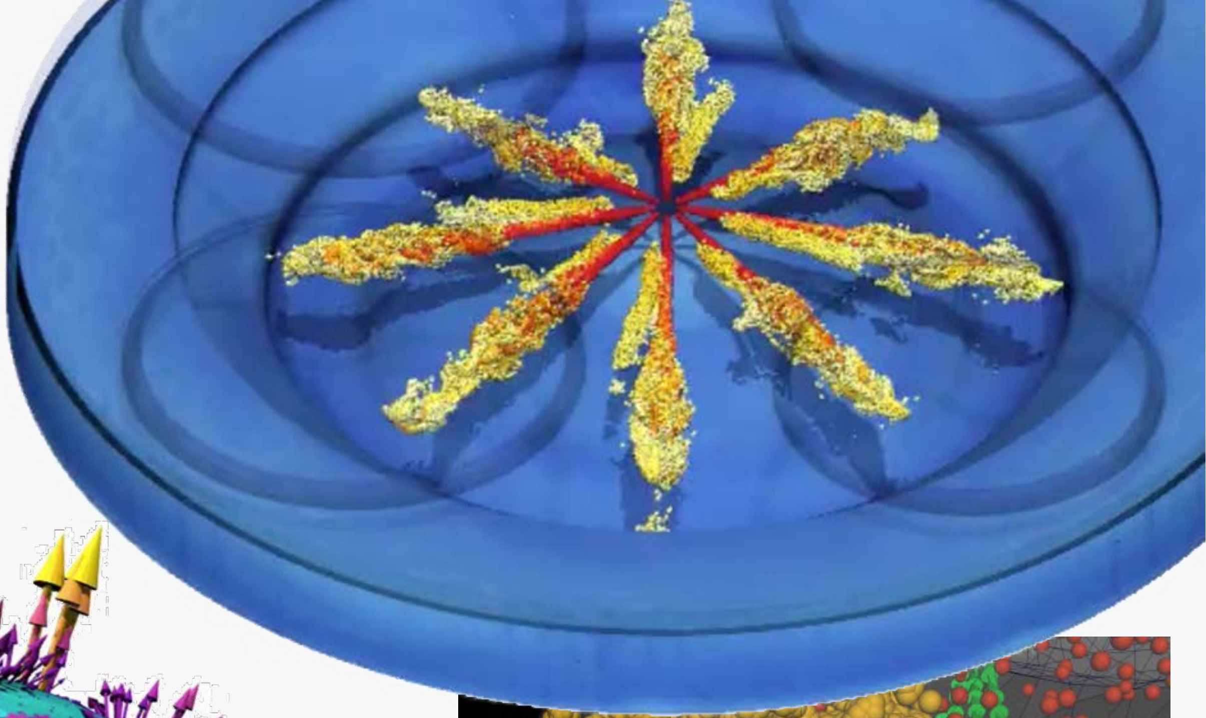
Location dictated by coordinate

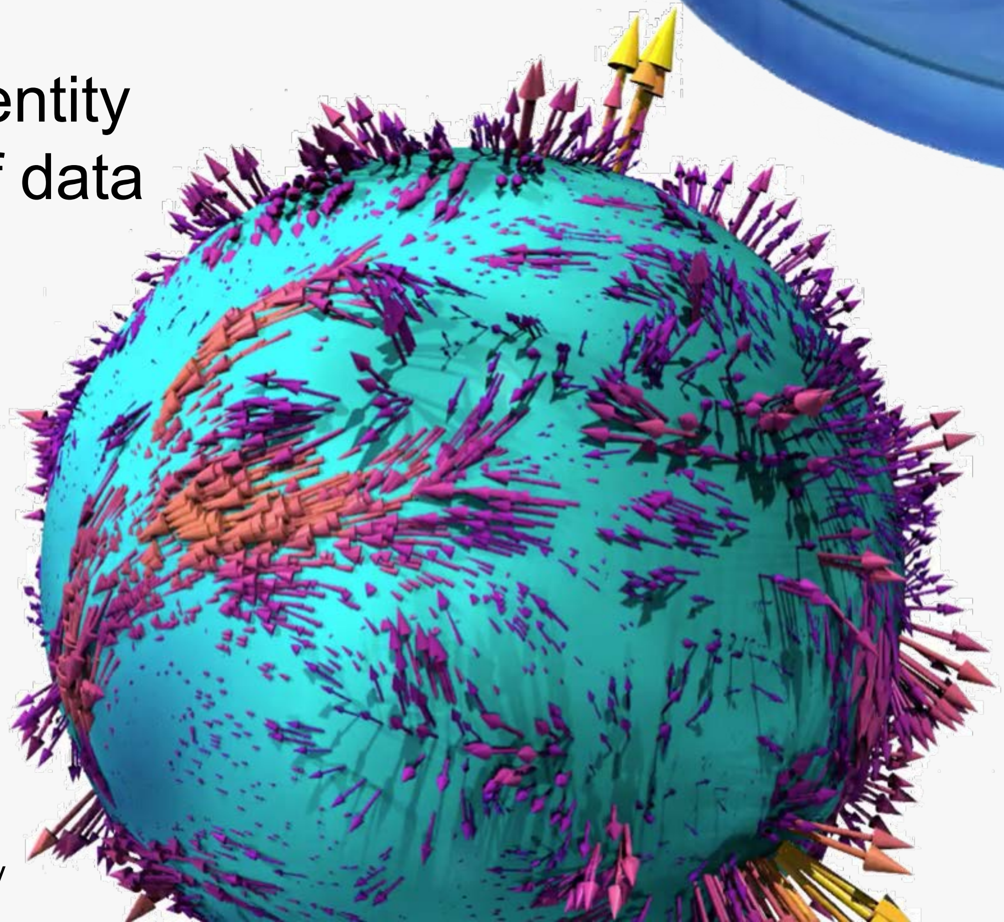
– 3D location on mesh

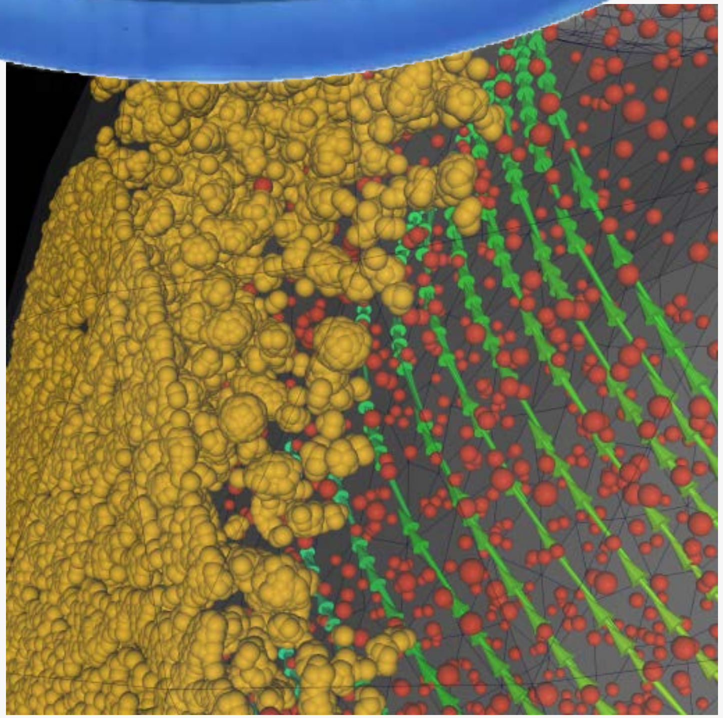
– 2D position in table/graph

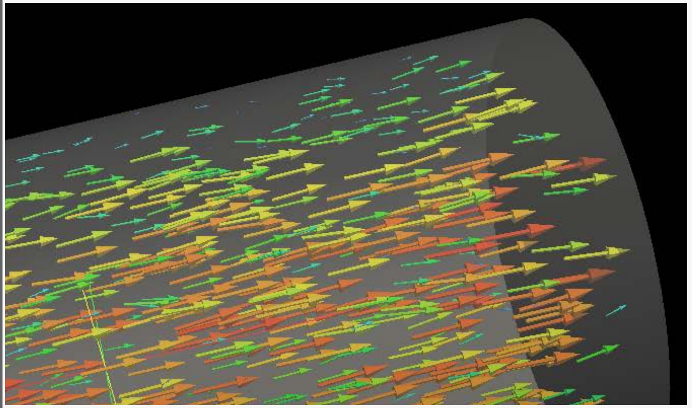
Attributes of graphical entity dictated by attributes of data

– color, size, orientation







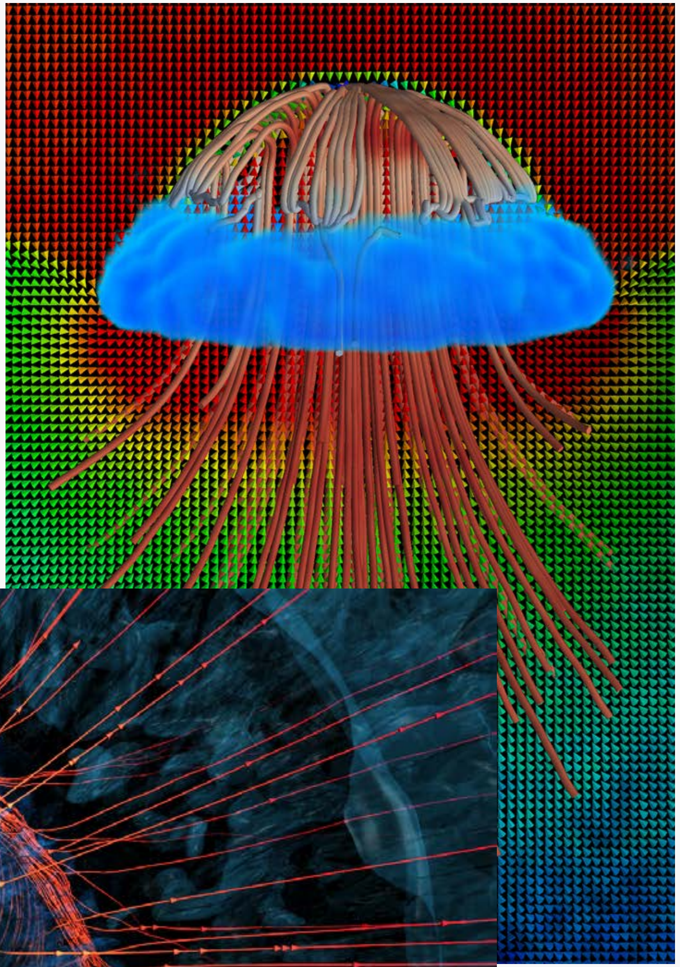


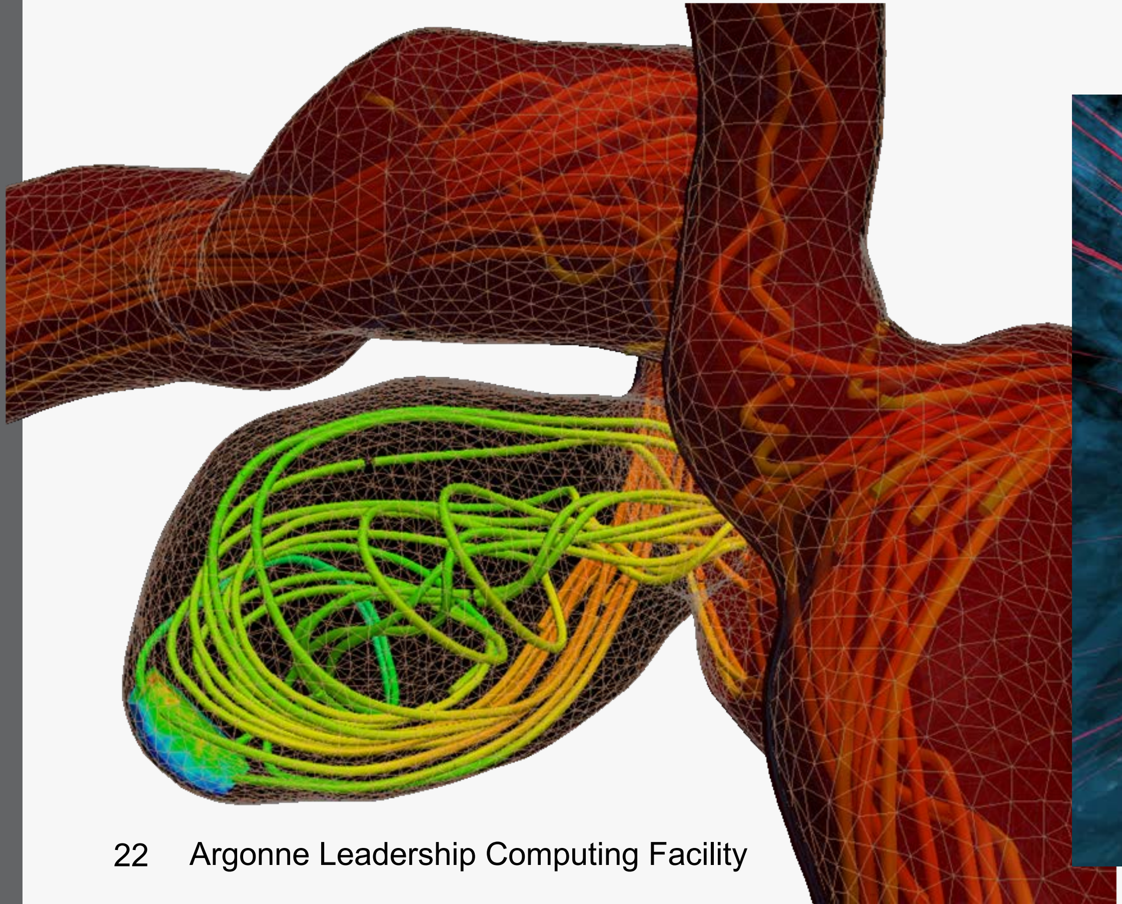
Data Representations: Streamlines

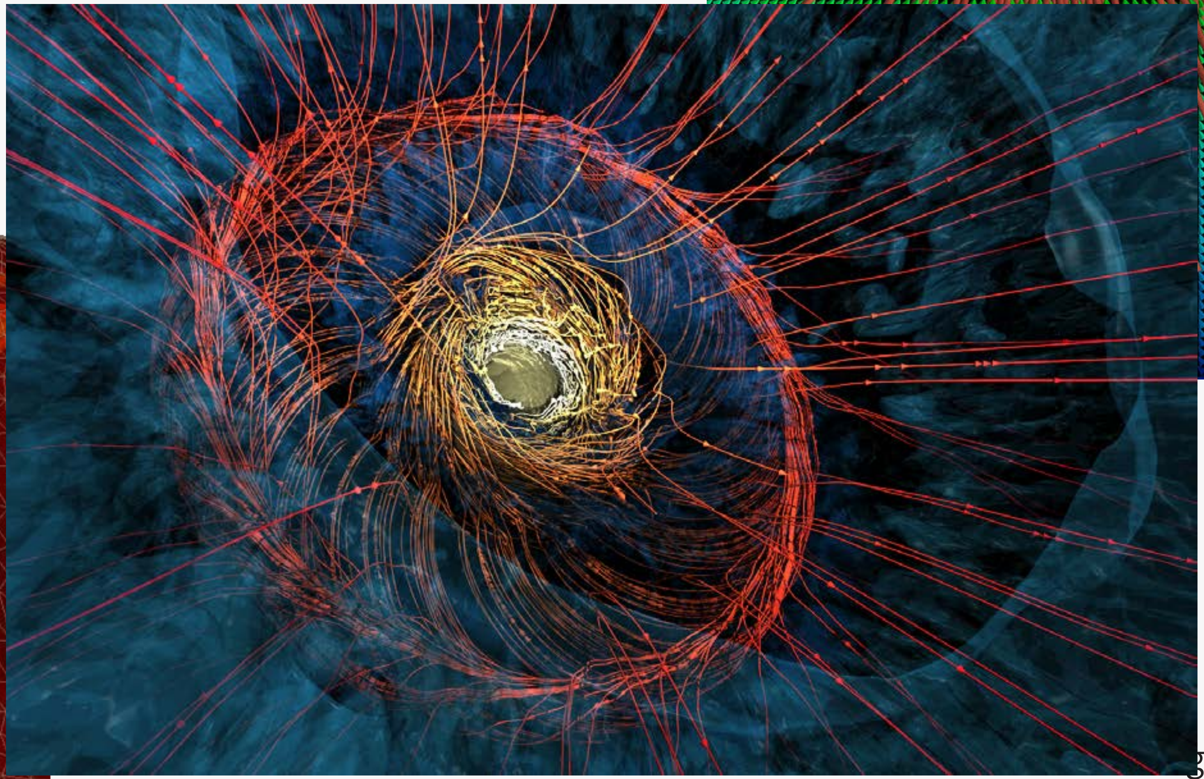
From vector field on a mesh (needs connectivity)

– Show the direction an element will travel in at any point in time.

VisIt & ParaView & vtk good at this







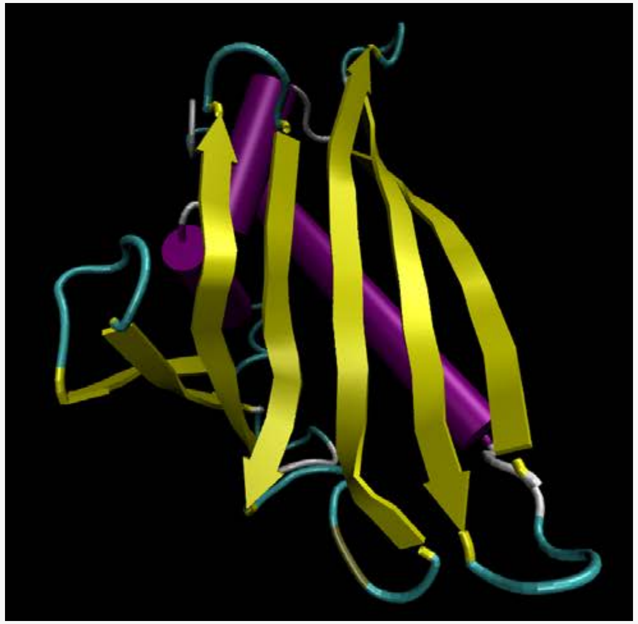
Data Representations: Pathlines

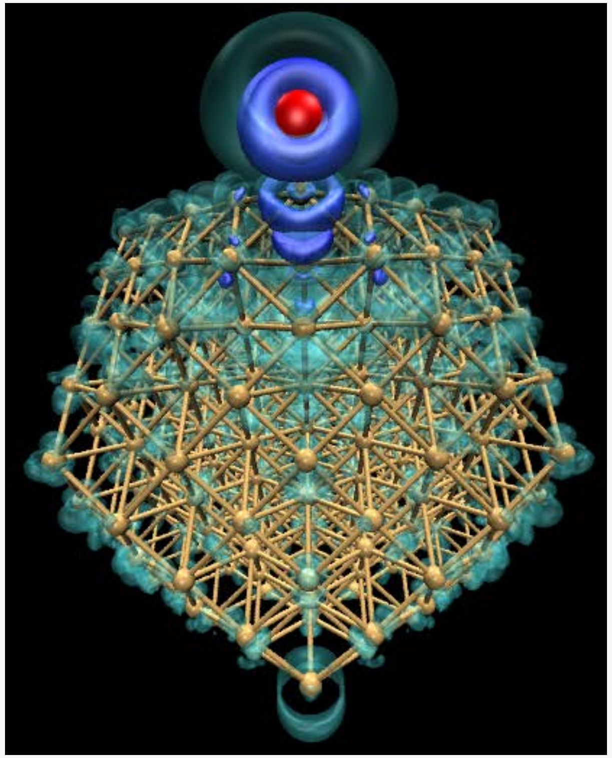
From vector field on a mesh (needs connectivity)

– Trace the path an element will travel over time.

VisIt & ParaView & vtk good at this

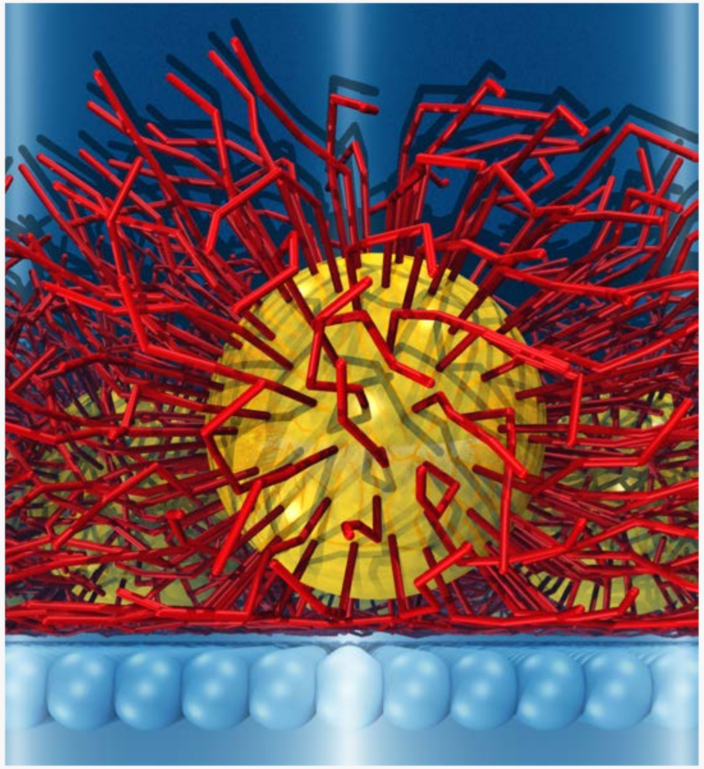
Molecular Dynamics Visualization

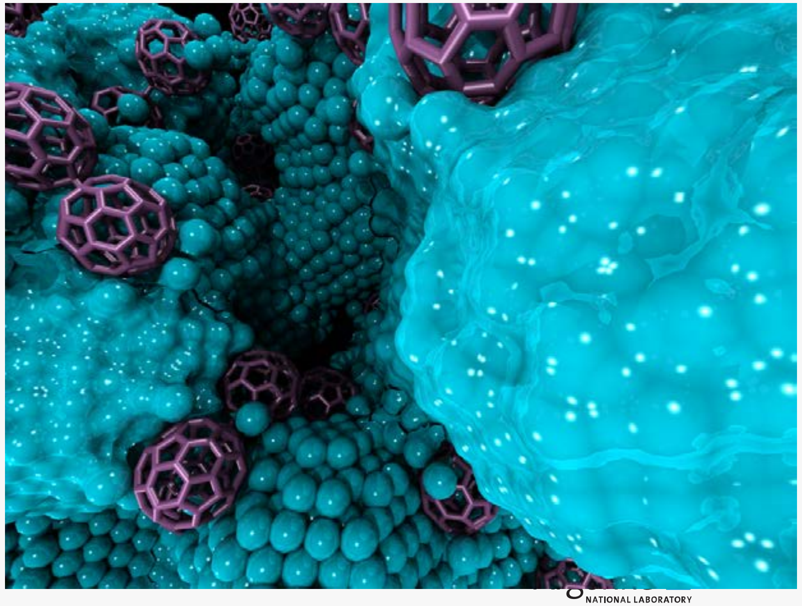




VMD:

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





VisIt & ParaView:

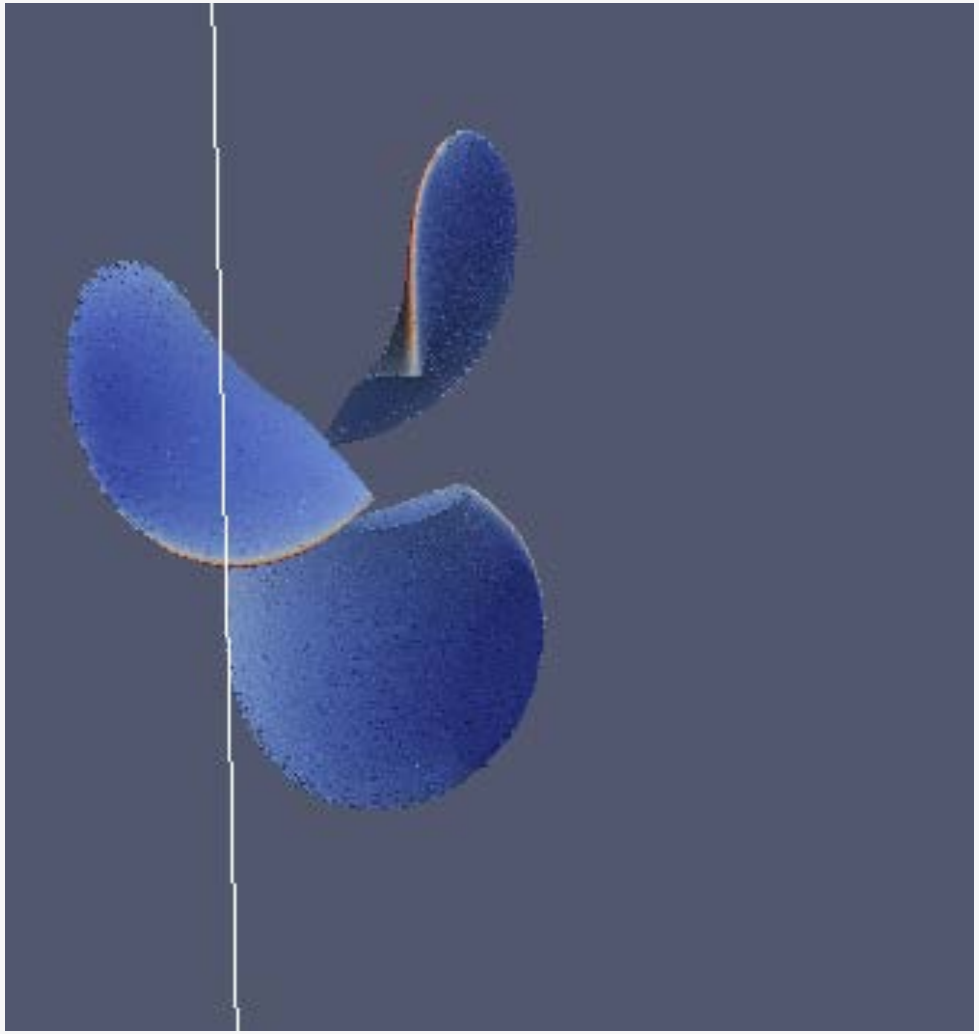
– Limited support for these types of representations, but improving

VTK:

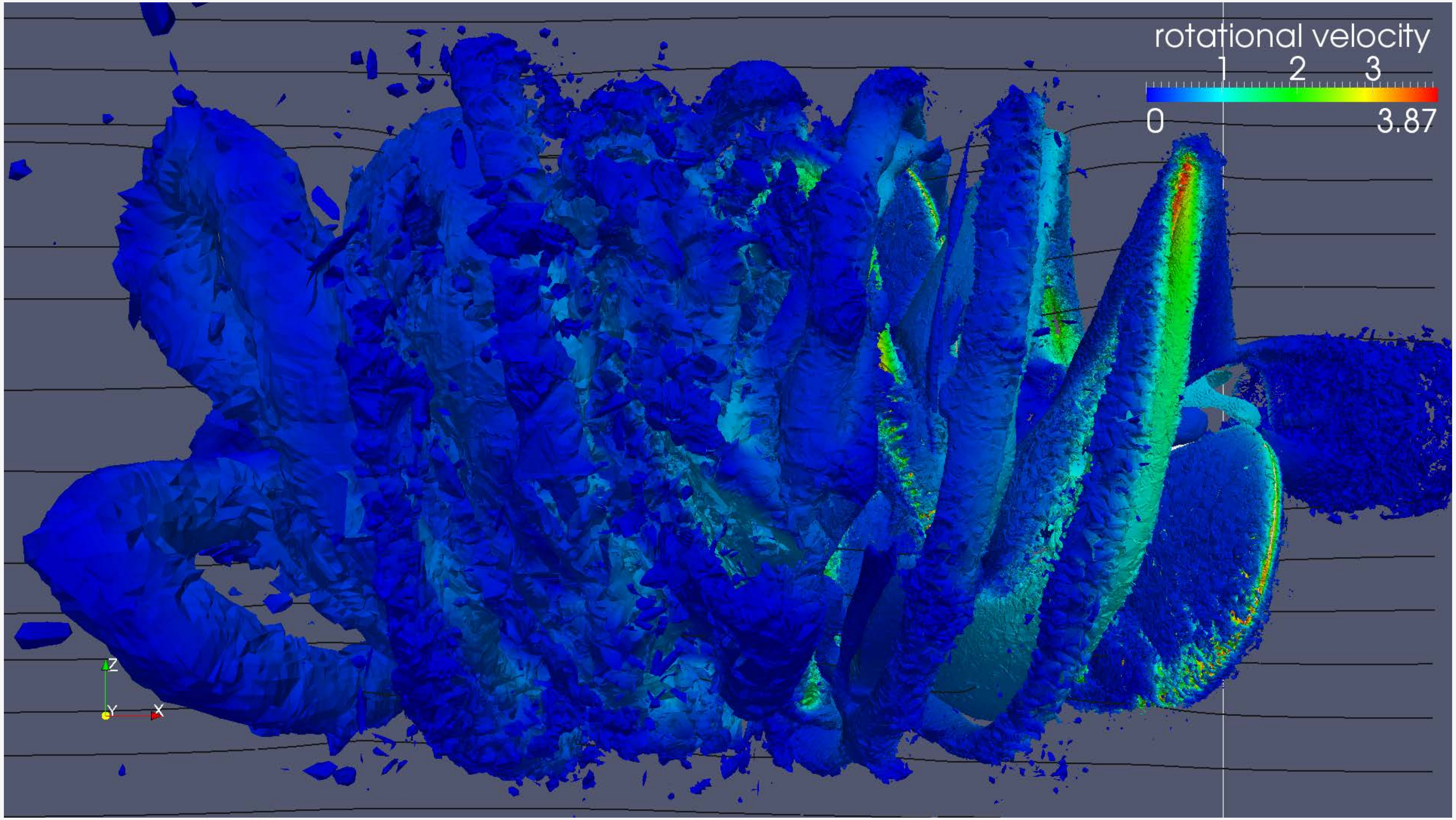
– Anything's possible if you try hard enough

Visualization for Debugging

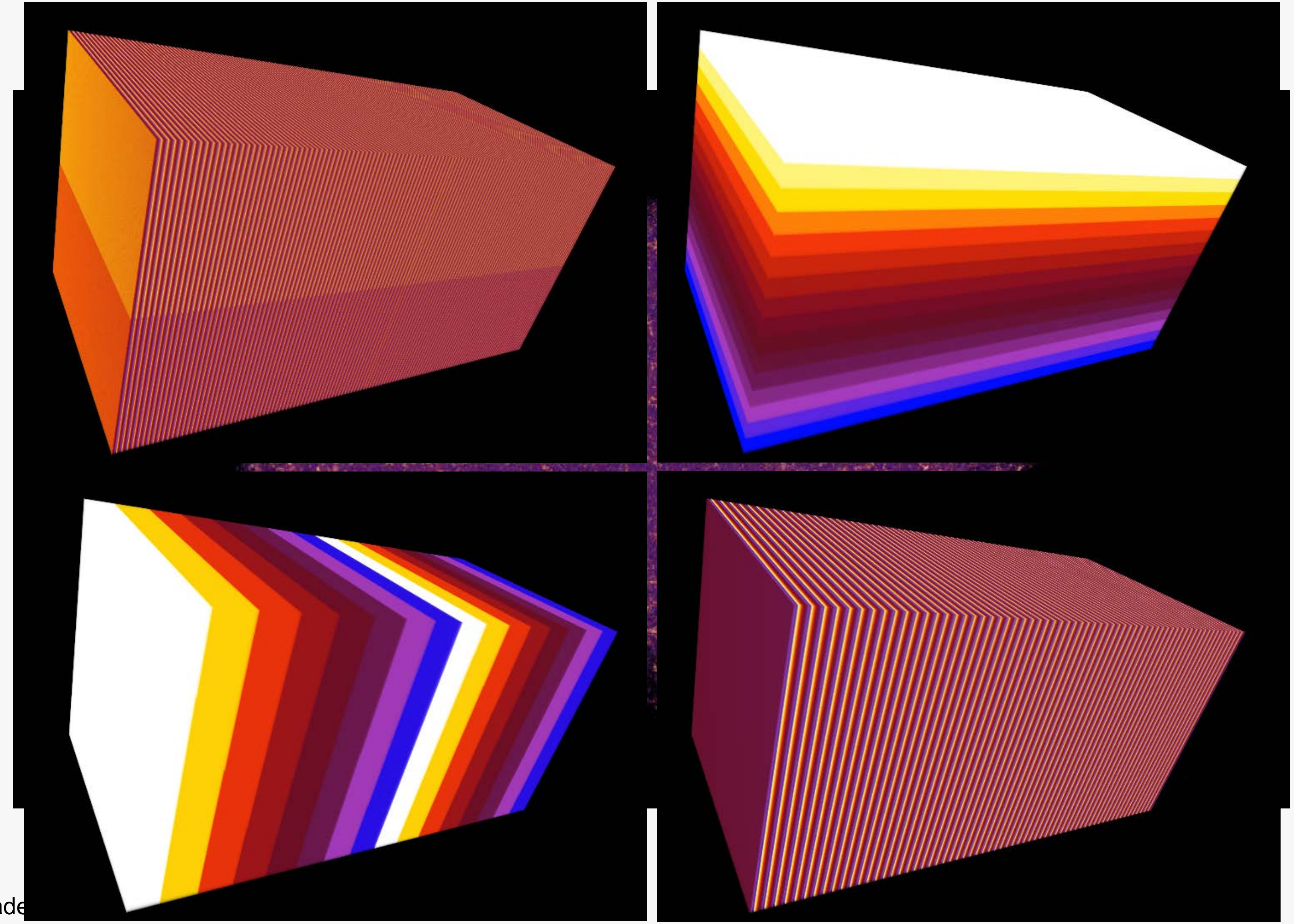
Visualization for Debugging



Visualization for Debugging

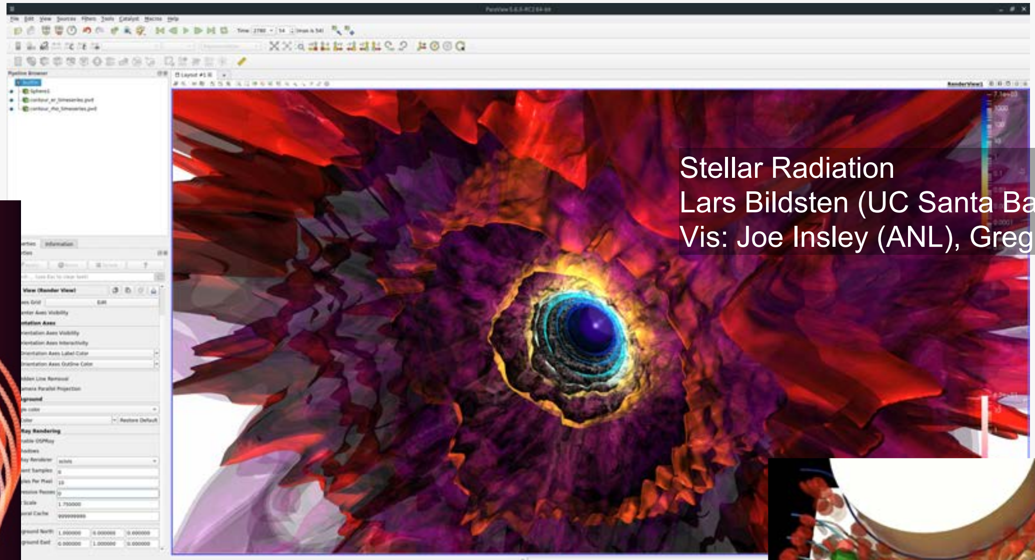


Visualization as Diagnostics: Color by Thread ID

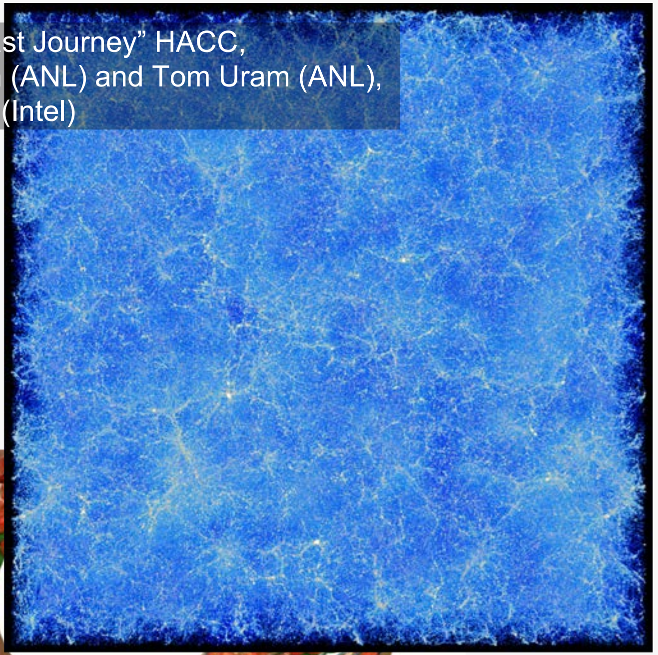


Advanced Rendering

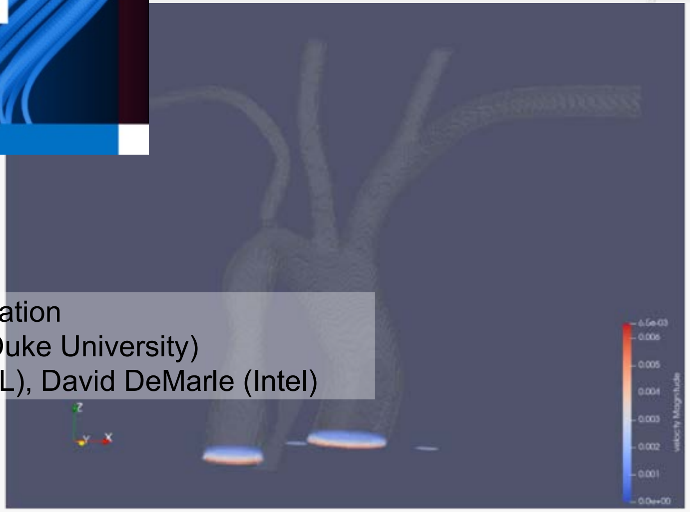
Intel® oneAPI Rendering Toolkit (“Render Kit”/"Render Framework")
Open Source Software for Advanced Rendering and Visualization



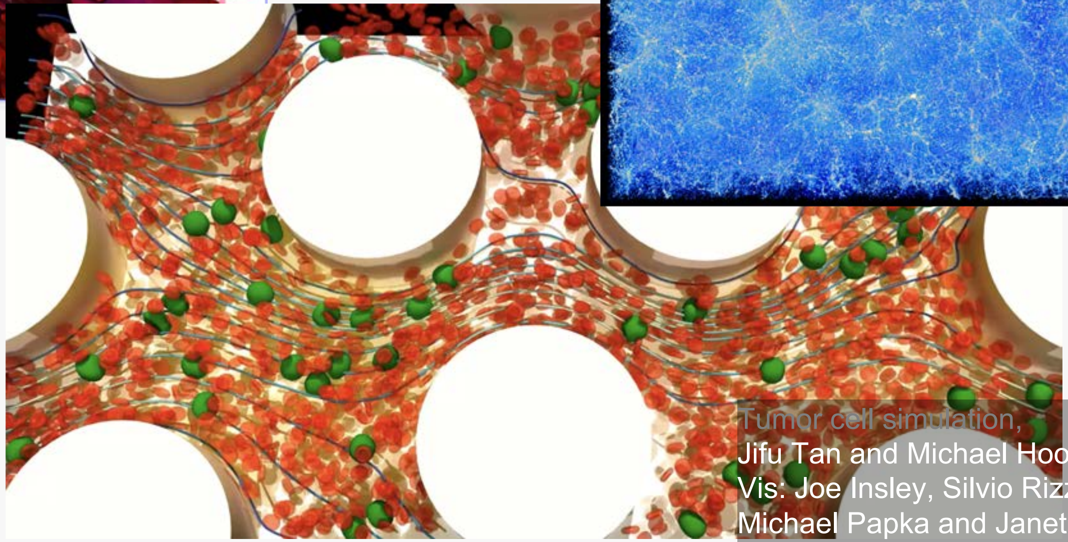
Stellar Radiation
Lars Bildsten (UC Santa Barbara),
Vis: Joe Insley (ANL), Greg Johnson (Intel)



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)

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

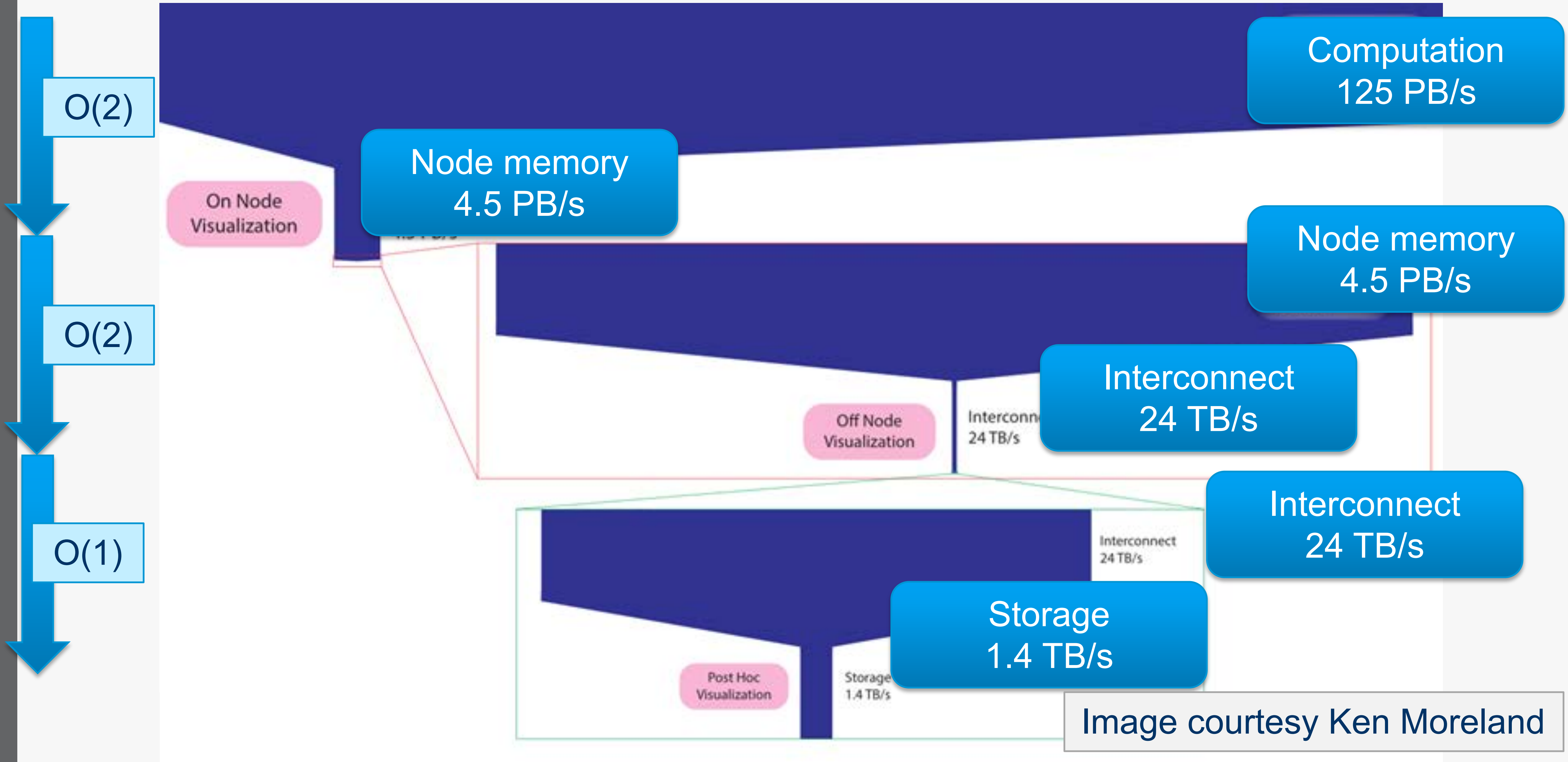


Image courtesy Ken Moreland

What are the problems?

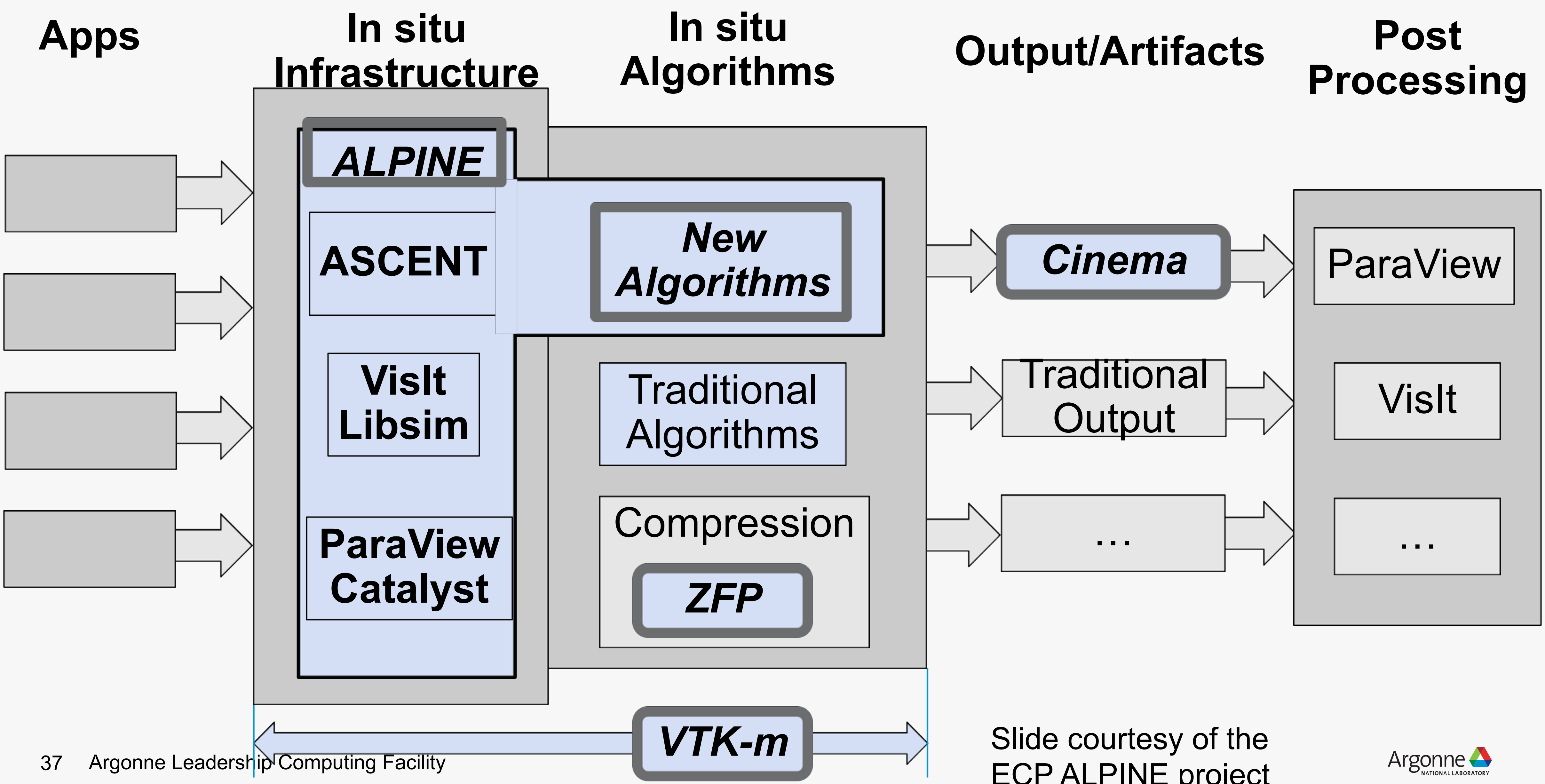
- Not enough I/O capacity on current HPC systems, and the trend is getting worse.
- If there's not enough I/O, you can't write data to storage, so you can't analyze it: lost science.
- Energy consumption: it costs a lot of power to write data to disk.
- Opportunity for doing better science (analysis)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

Slide courtesy the SENSEI team www.sensei-insitu.org

In Situ Frameworks and Infrastructures at ALCF

Name	Description	Contact person at ATPESC
ALPINE	In Situ algorithms and infrastructure for the Exascale Computing Project	Silvio Rizzi, Cyrus Harrison
ASCENT	A flyweight in situ visualization and analysis runtime for multi-physics HPC simulations	Cyrus Harrison
SENSEI	Write once run anywhere. Multiple backends. MxN in transit communication patterns	Silvio Rizzi, Joe Insley
ParaView/Catalyst	*In situ* use case library, with an adaptable application programming interface (API), that orchestrates the delicate alliance between simulation and analysis and/or visualization tasks	Dan Lipsa
Libsim	Originally developed to facilitate interactive connections from VisIt to running simulations	Cyrus Harrison
SmartSim	SmartSim is a software framework that facilitates the convergence of numerical simulations and AI workloads on heterogeneous architectures	Silvio Rizzi

Exascale Computing Project
Software Technology Data and Visualization



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

srizzi — srizzi@thetalogin6:~ — thetalogin6 — ssh -Y theta — 78×42

```
srizzi@thetagpu23:/lus/grand/projects/visualization/srizzi/WOIV22$ ls
adios_transport_sst.xml  consumer.sh    log.lammps    producer.simg
adios_write_bp4.xml      consumer.simg  mpich         vtk_io.xml
adios_write_sst.xml      in_lj          producer.sh
srizzi@thetagpu23:/lus/grand/projects/visualization/srizzi/WOIV22$ hostname -f
thetagpu23.mcp.alcf.anl.gov
srizzi@thetagpu23:/lus/grand/projects/visualization/srizzi/WOIV22$ ./producer.
sh
```

srizzi — srizzi@cc018:/lus/grand/projects/visualization/srizzi/WOIV22 — ssh -Y cooley — 76×42

```
[srizzi@cc018 WOIV22]$ ls
adios_transport_sst.xml  consumer.sh    log.lammps    producer.simg
adios_write_bp4.xml      consumer.simg  mpich         vtk_io.xml
adios_write_sst.xml      in_lj          producer.sh
[srizzi@cc018 WOIV22]$ hostname -f
cc018.fst.alcf.anl.gov
[srizzi@cc018 WOIV22]$ ./consumer.sh
```

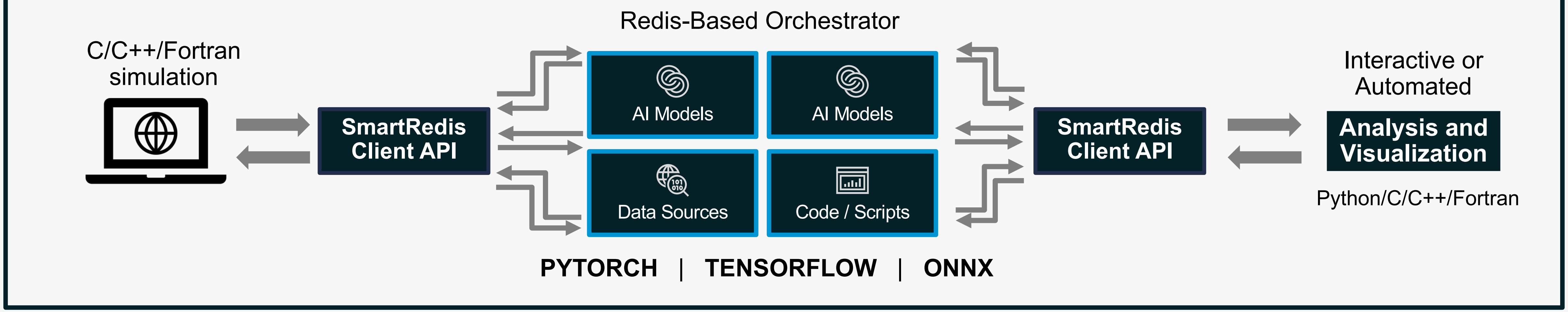
SmartSim Overview

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**

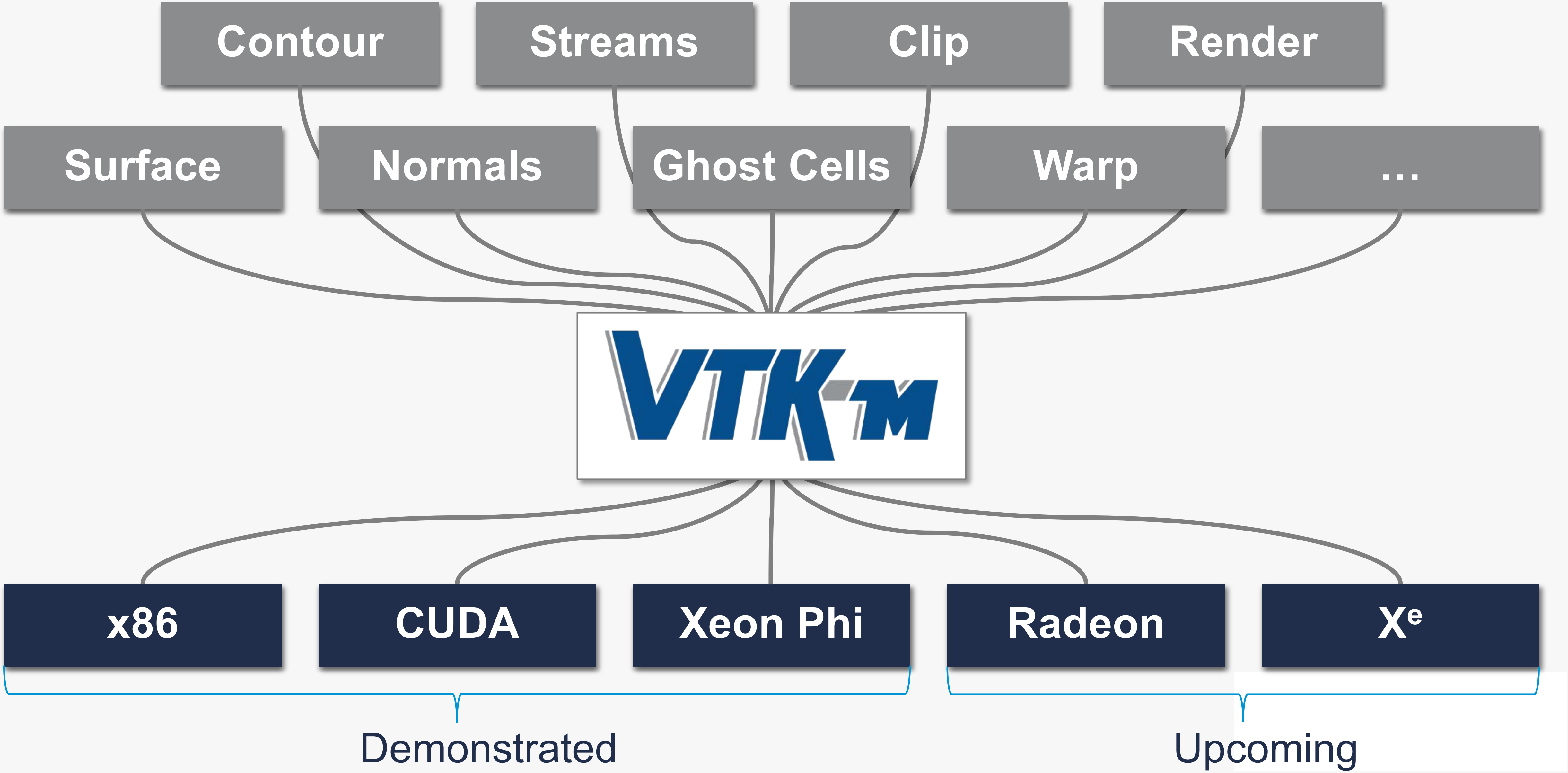
All of these can be done without touching the filesystem, i.e. **data-in-motion**



slide courtesy of the HPE SmartSim team

Infrastructures

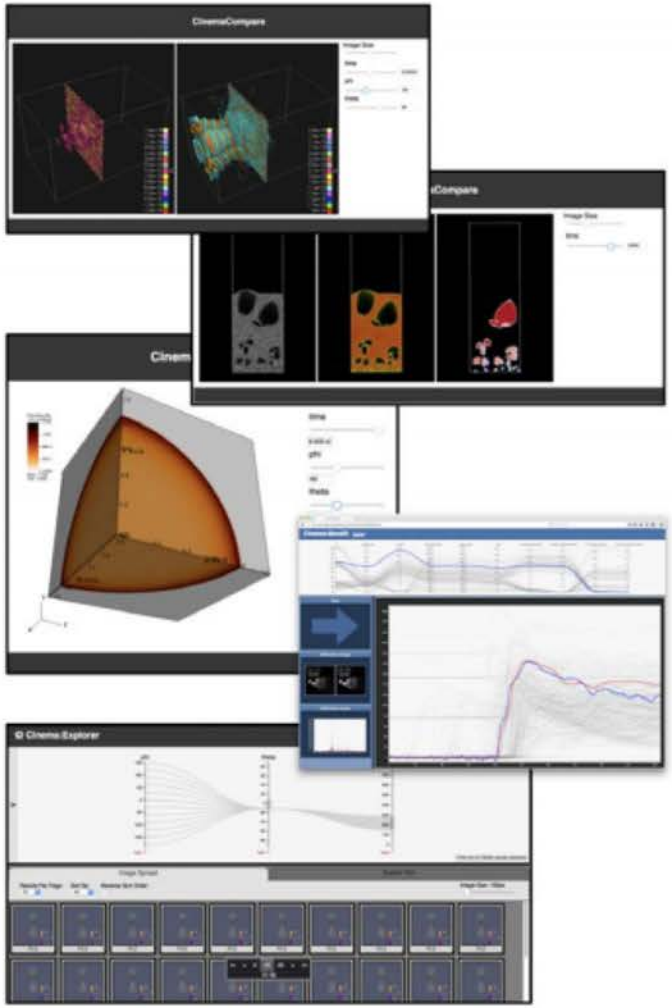
VTK-m's main thrust: a write-once-run-everywhere framework



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.



Slide courtesy of Los Alamos National Lab and the CINEMA project

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

Argonne NATIONAL LABORATORY

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