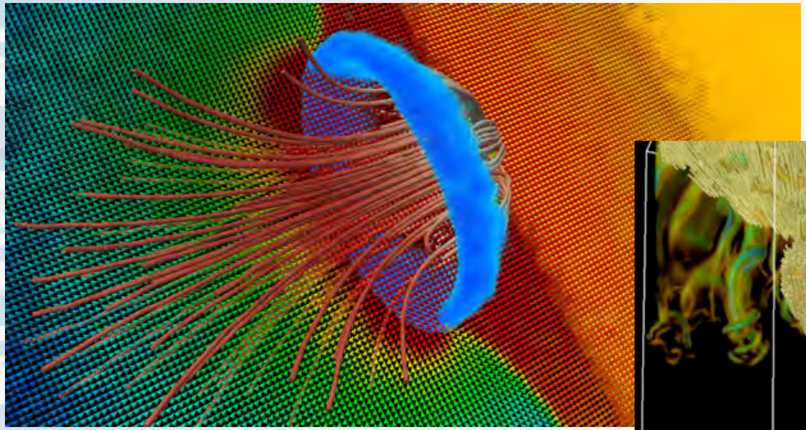
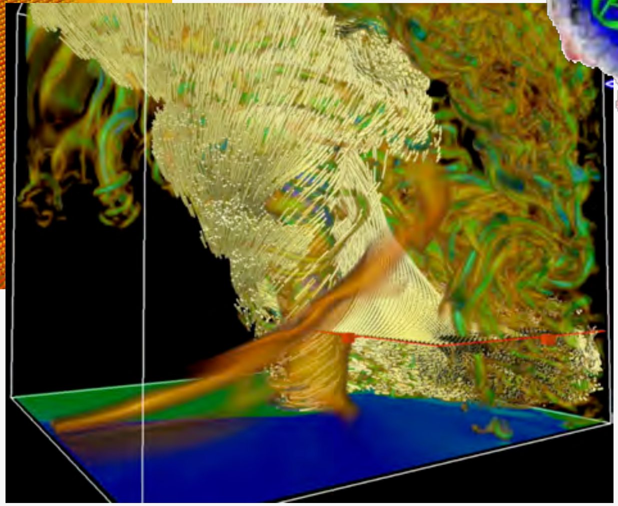
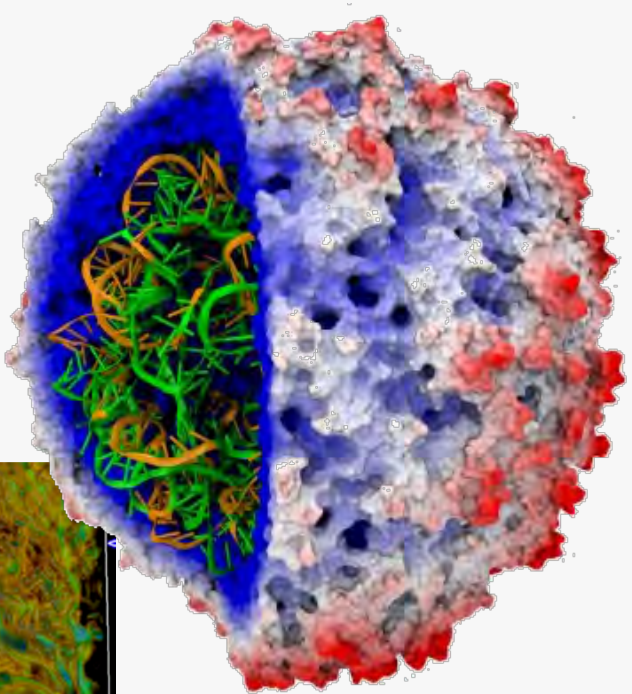
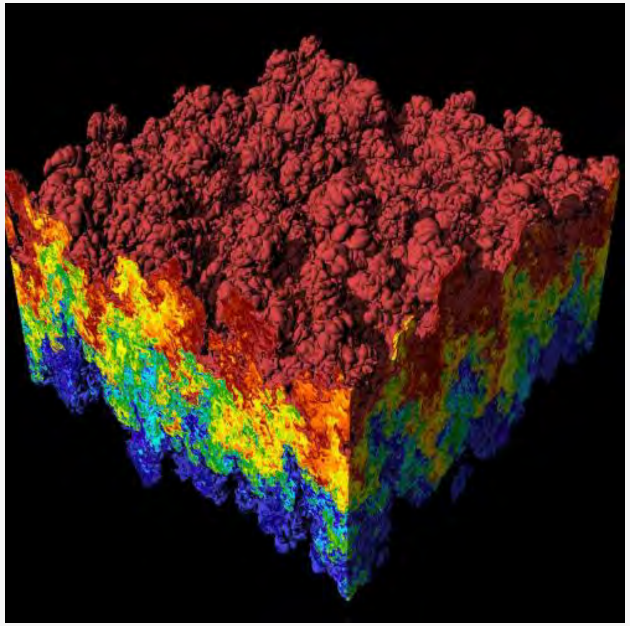
Argonne Training Program on Extreme-Scale Computing (ATPESC)

Data Analysis and Visualization









EXASCALE COMPUTING PROJECT

U.S. DEPARTMENT OF ENERGY | Office of Science

National Nuclear Security Administration

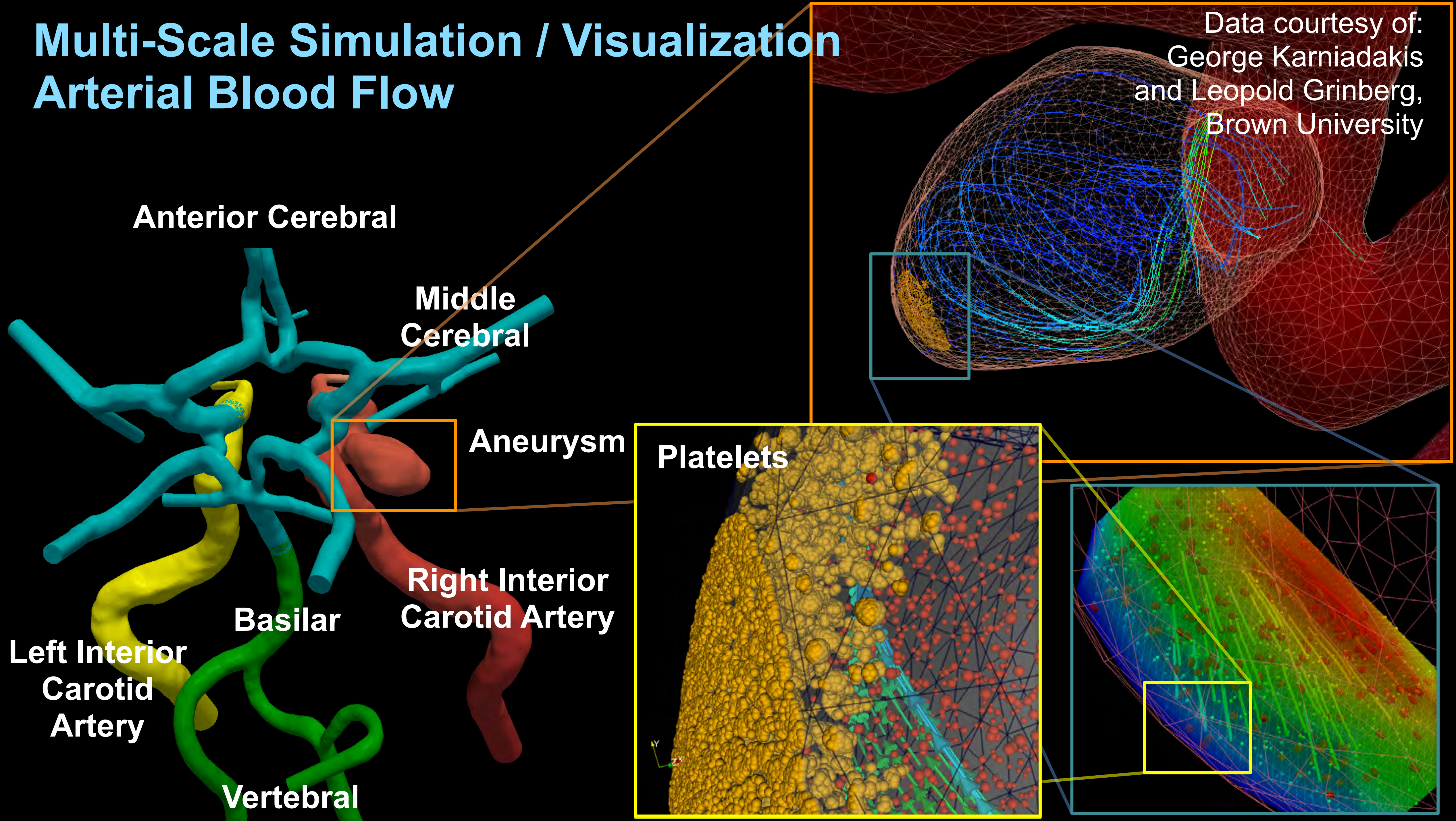
Visualization & Data Analysis

Time	Title of presentation	Lecturer
9:30 am	**Data Analysis and Visualization Introduction**	Mike Papka *ANL/NIU*, Joe Insley *ANL/NIU*, Silvio Rizzi, *ANL*
10:15 am	**Scalable Molecular Visualization and Analysis Tools in VMD**	John Stone *UIUC*
11:00 am	*Break*	
11:15 am	**Large Scale Visualization with ParaView**	Dan Lipsa *Kitware*
12:30 pm	*Lunch*	
1:30 pm	**Visualization and Analysis of HPC Simulation Data with VisIt**	Cyrus Harrison *LLNL*
2:45 pm	**Vapor**	Scott Pearse *UCAR*
3:30 pm	*Break*	
3:45 pm	**Exploring Visualization with Jupyter Notebooks**	• Tommy Marrinan *St. Thomas / ANL* • David Koop *NIU* • Cyrus Harrison *LLNL*, Matt Larsen *LLNL*
5:00 pm	*Adjourn*	

Here’s the plan…

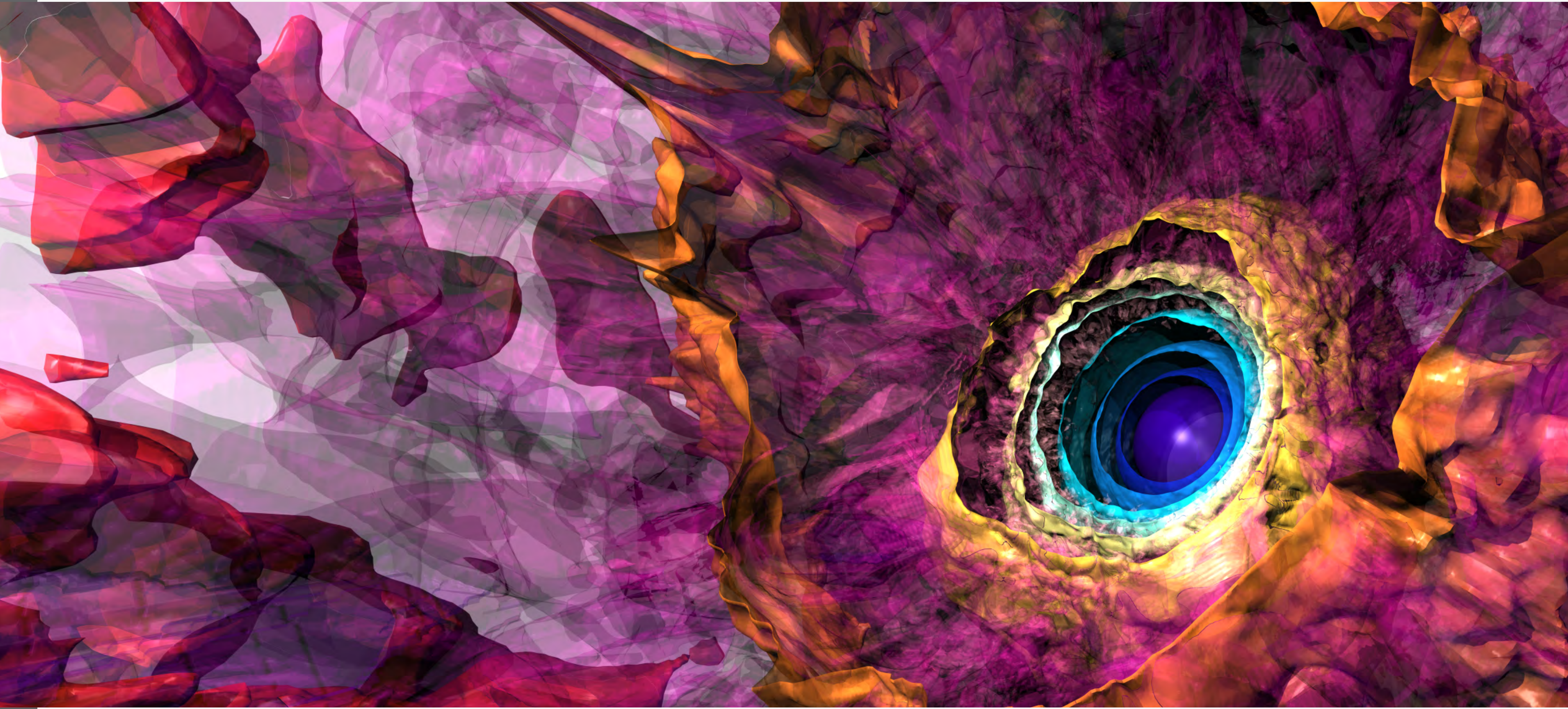
- Examples of visualizations
- Visualization resources
- Visualization tools and formats
- Data representations
- Visualization for debugging
- In Situ Visualization and Analysis

Multi-Scale Simulation / Visualization
Arterial Blood Flow

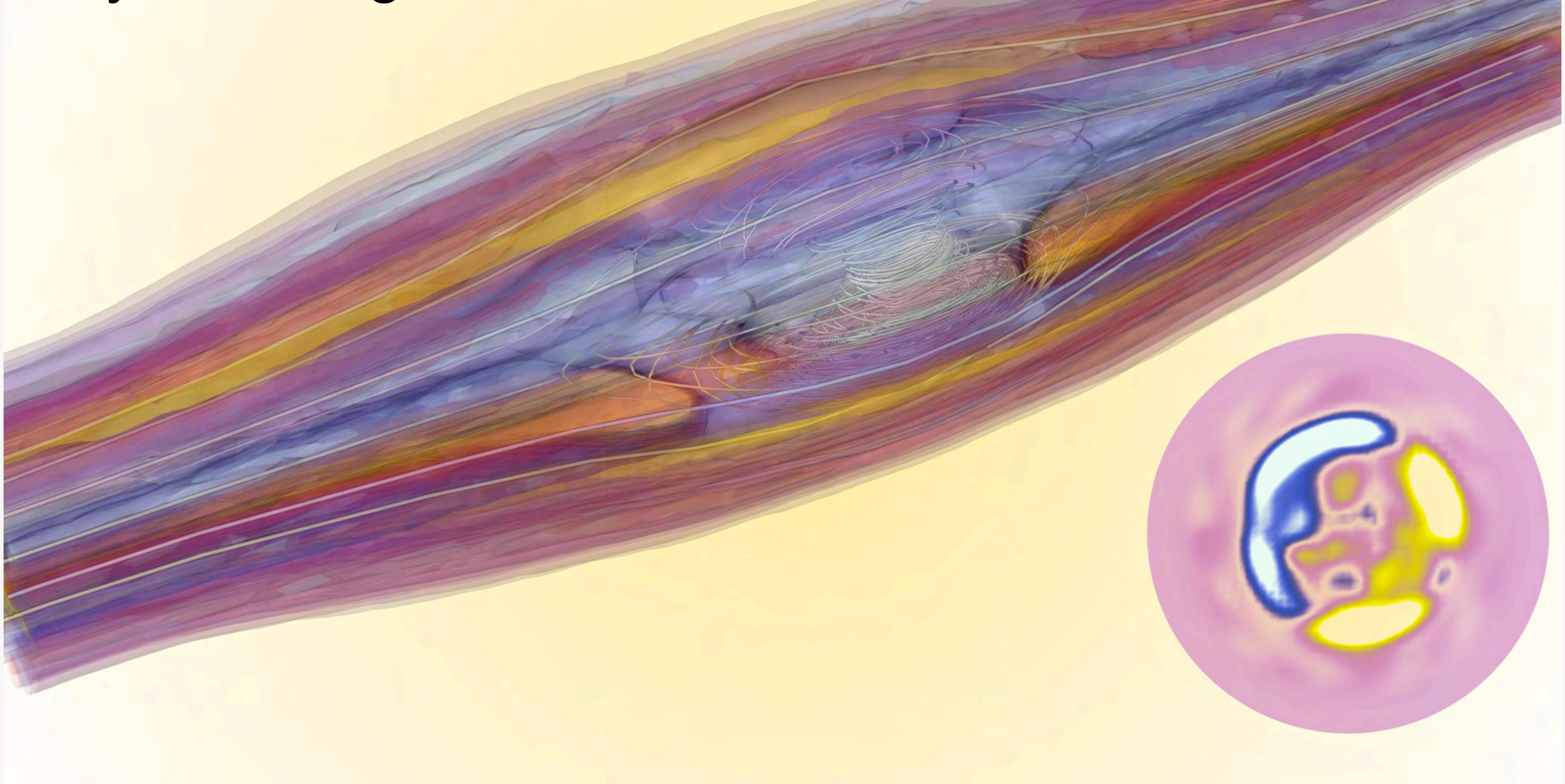


Physics: Stellar Radiation

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

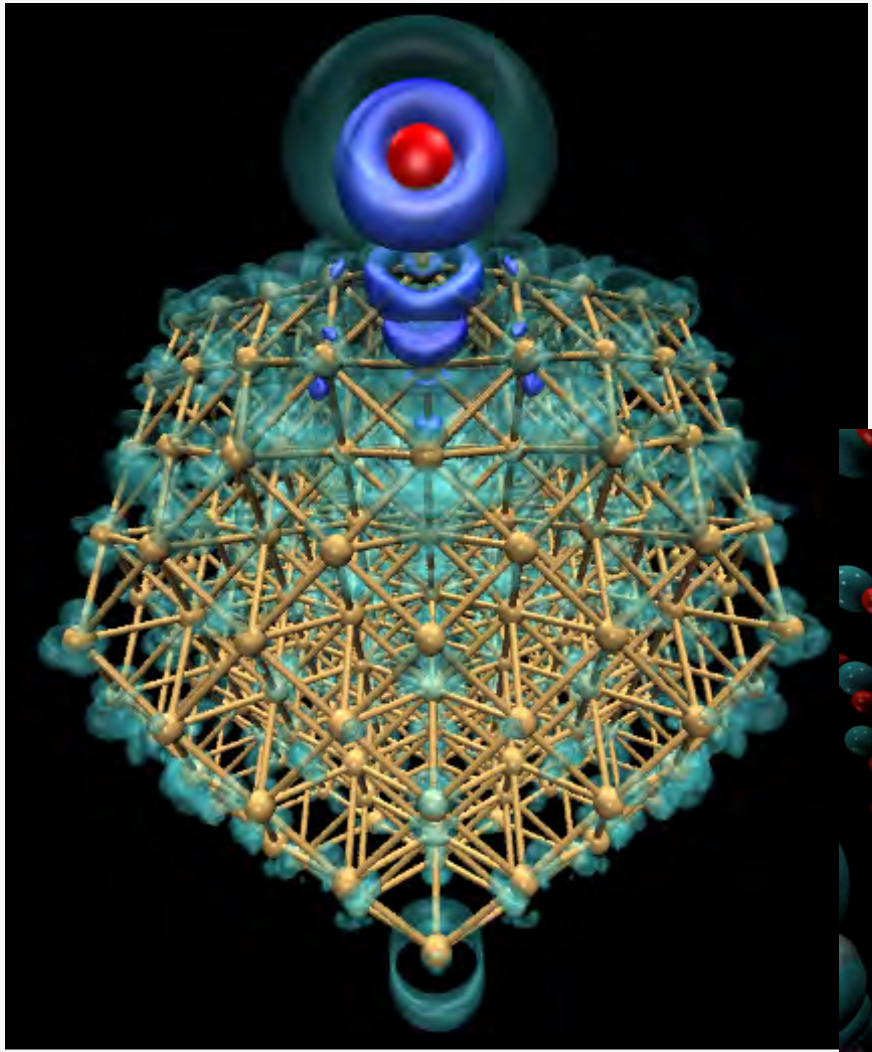


Physics: Magnetic Confinement Fusion



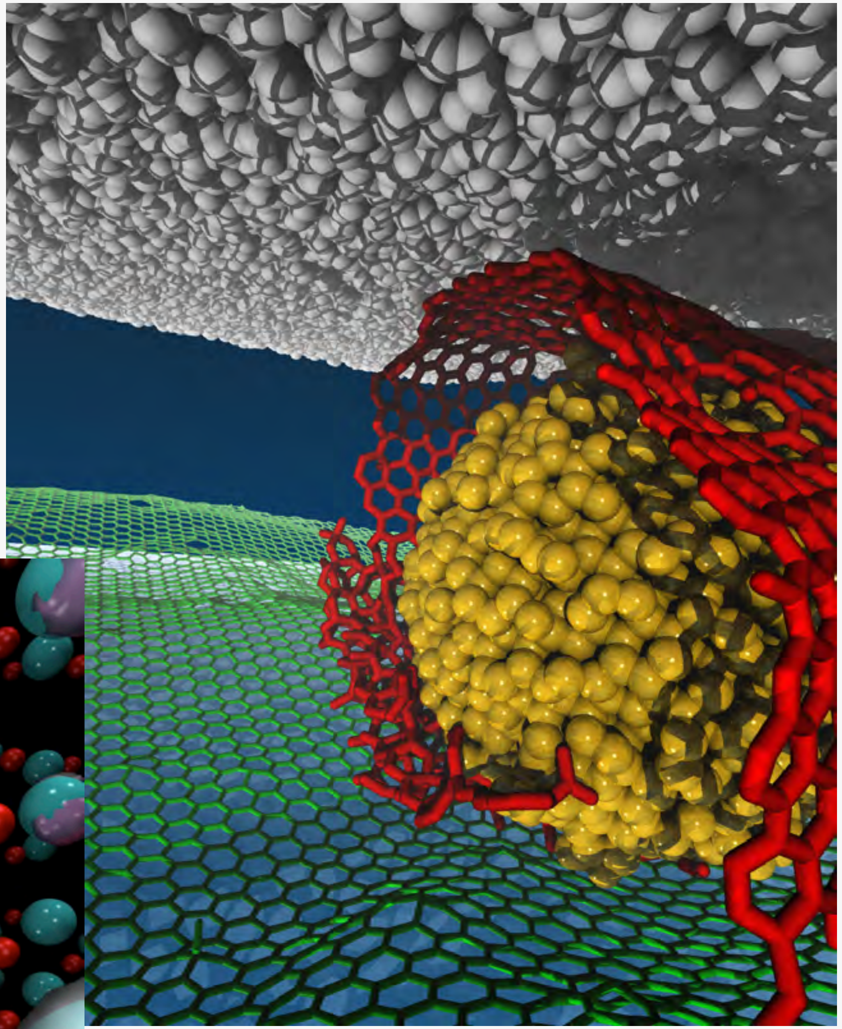
Data courtesy of Sean Dettrick, TAE Technologies, Inc.

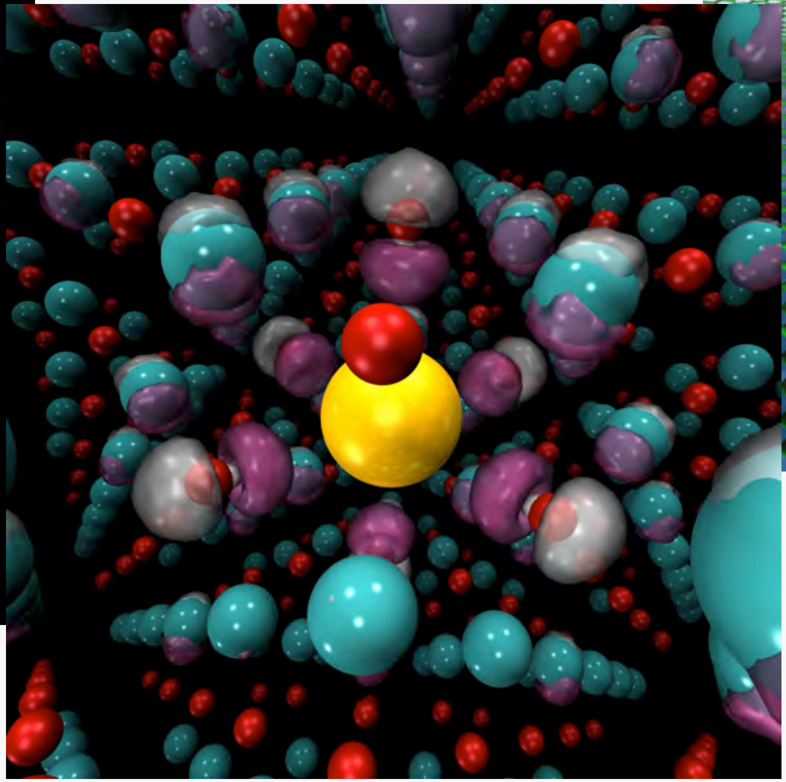
Materials Science / Molecular



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

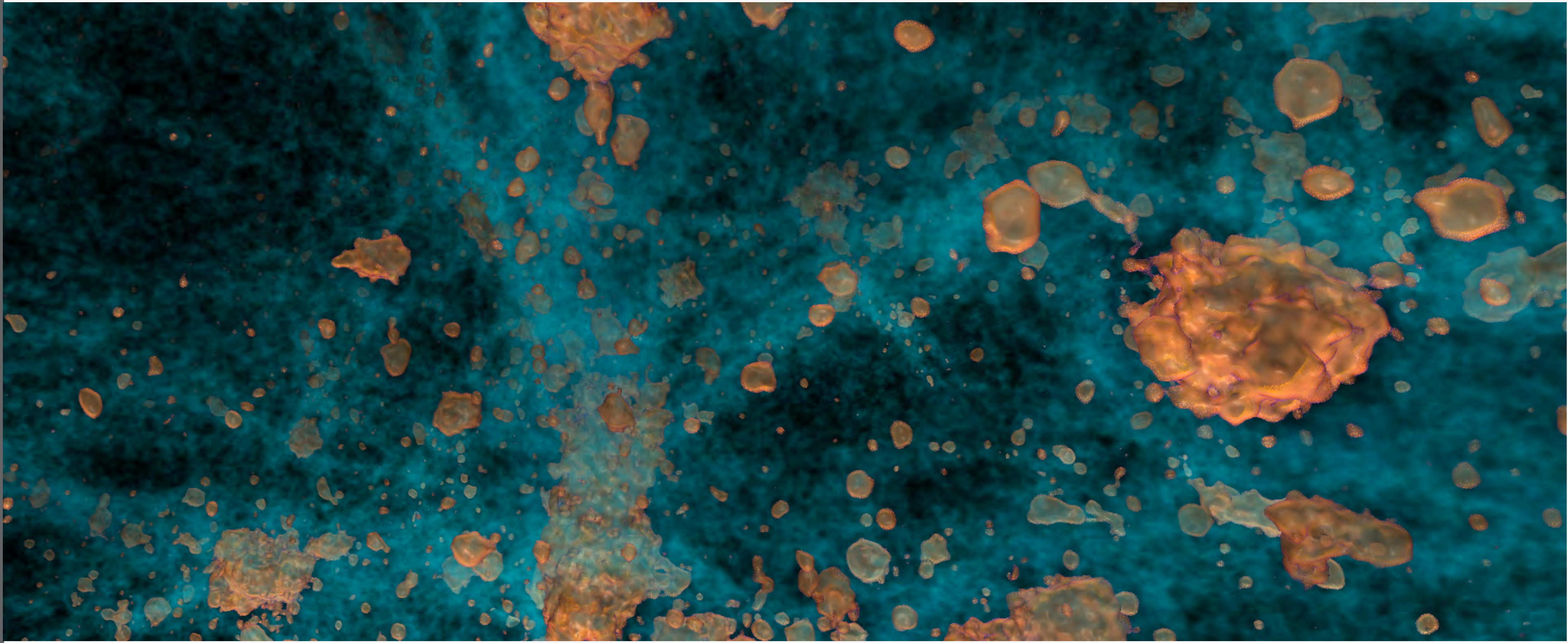
Data courtesy of: Subramanian Sankaranarayanan, Argonne National Laboratory





Data courtesy of: Paul Kent, Oak Ridge National Laboratory, Anouar Benali, Argonne National Laboratory

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

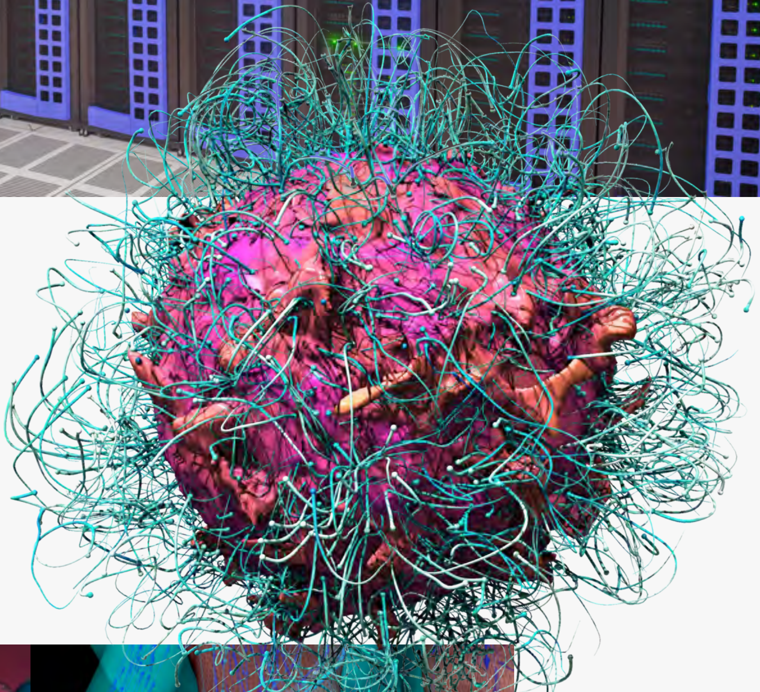
Aggregate RAM of 47 TB

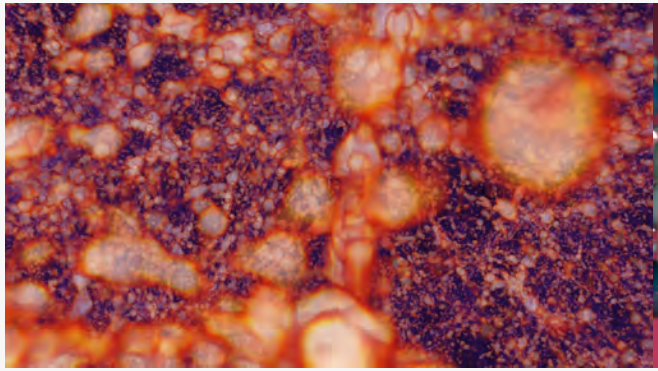
Aggregate GPU memory of ~3TB

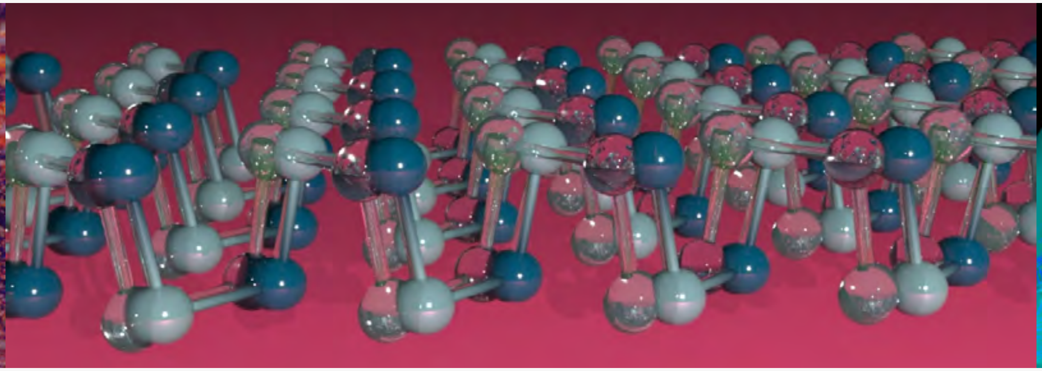
Cray CS System

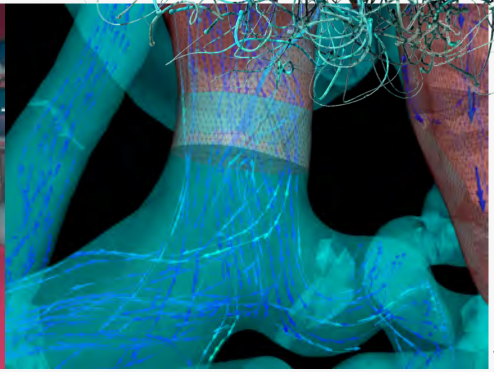
216 port FDR IB switch with uplinks to our QDR infrastructure

Mounts the 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

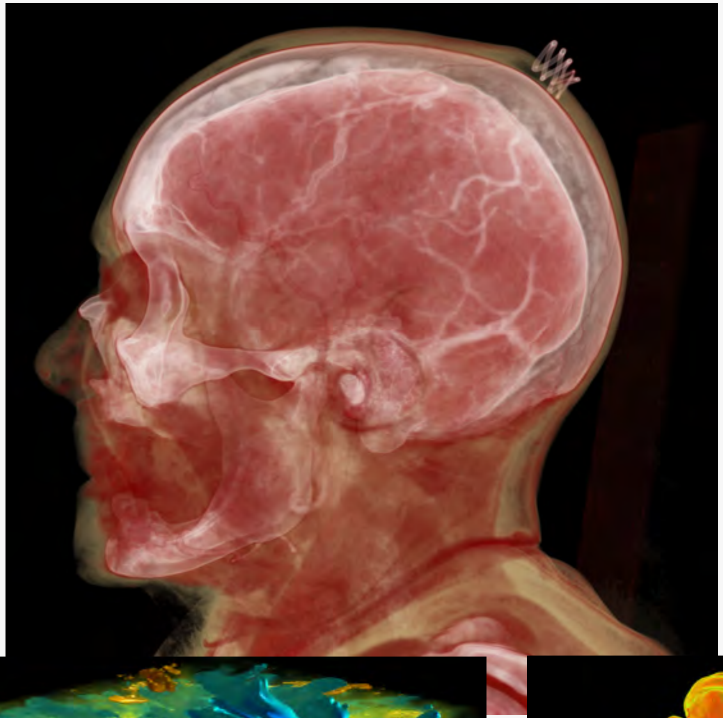
- Programming environment / API
- Additional capabilities, finer control
- Smaller memory footprint
- Requires more expertise (build custom applications)

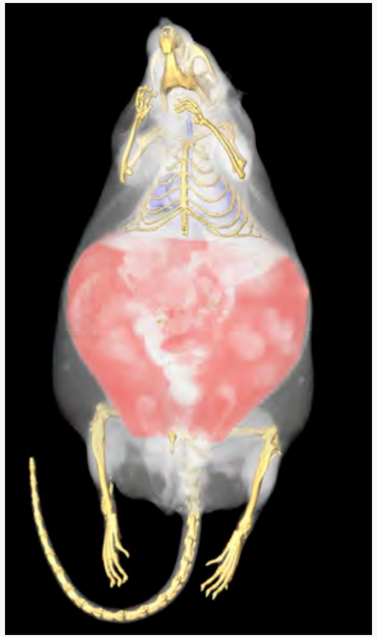
Data File Formats (ParaView & VisIt)

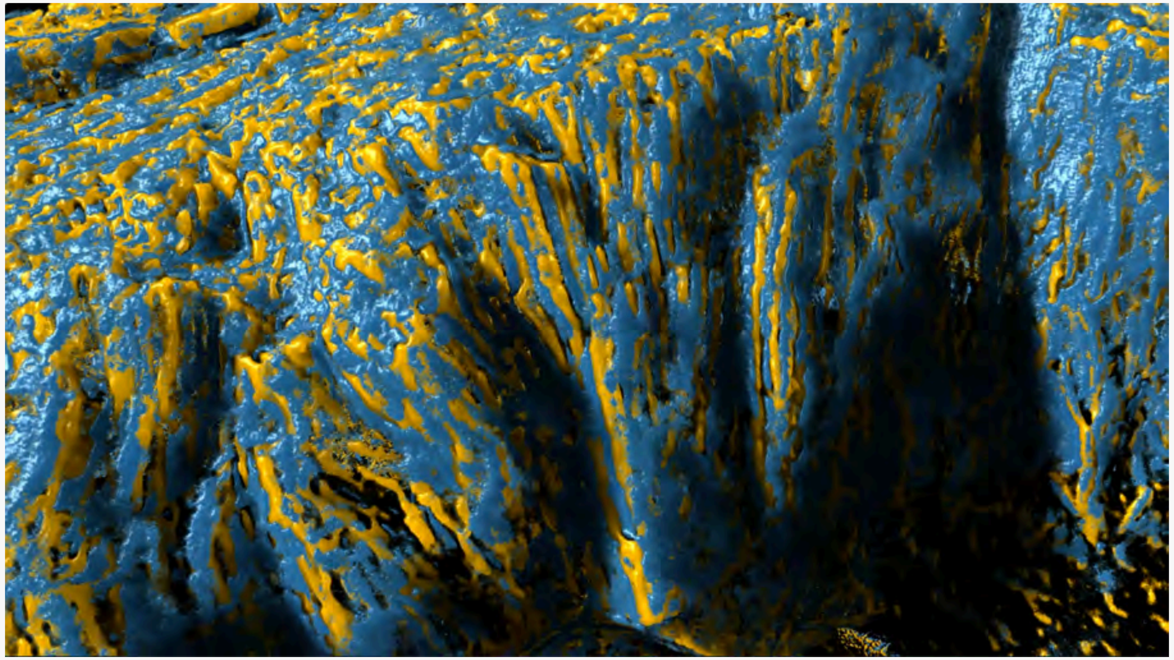
- VTK
- Parallel (partitioned) VTK
- VTK MultiBlock (MultiGroup, Hierarchical, Hierarchical Box)
- Legacy VTK
- Parallel (partitioned) legacy VTK
- EnSight files
- EnSight Master Server
- Exodus
- BYU
- XDMF
- PLOT2D
- PLOT3D
- SpyPlot CTH
- HDF5 raw image data
- DEM
- VRML
- PLY
- Polygonal Protein 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

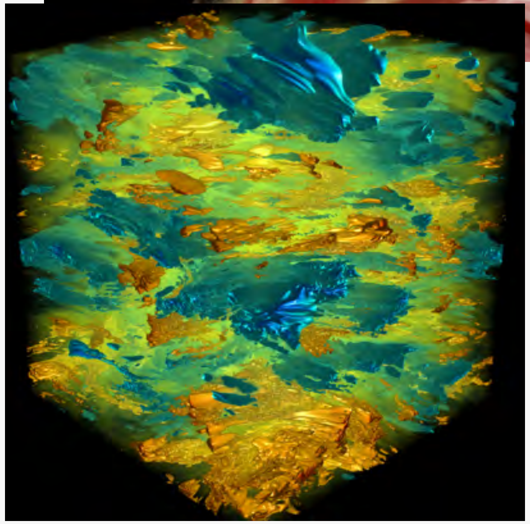
Data Representations

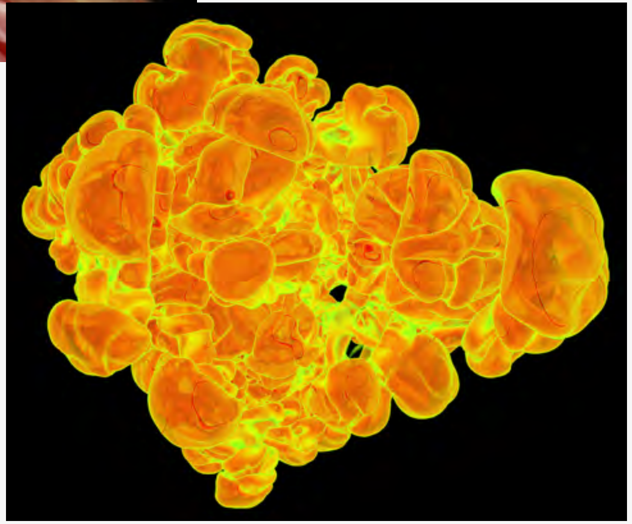
Data Representations: Volume Rendering

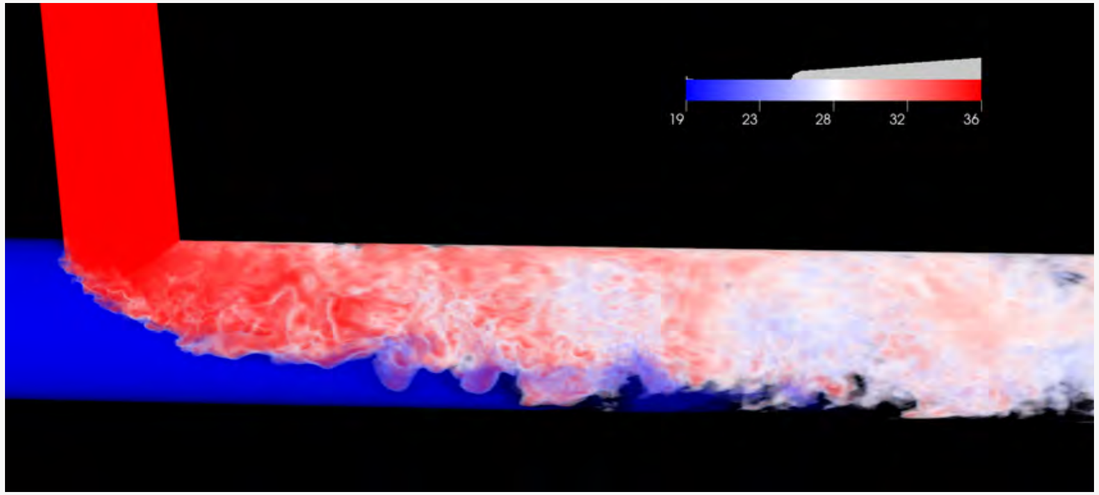












Data Representations: Glyphs

2D or 3D geometric object to represent point data

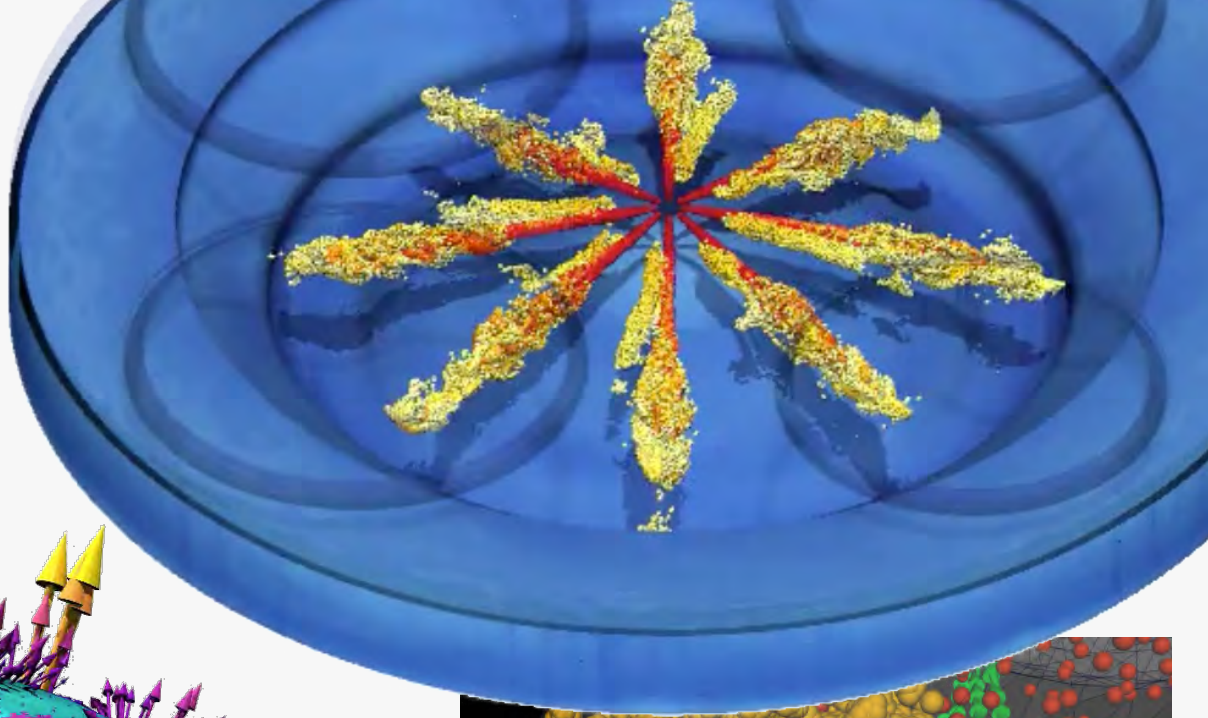
Location dictated by coordinate

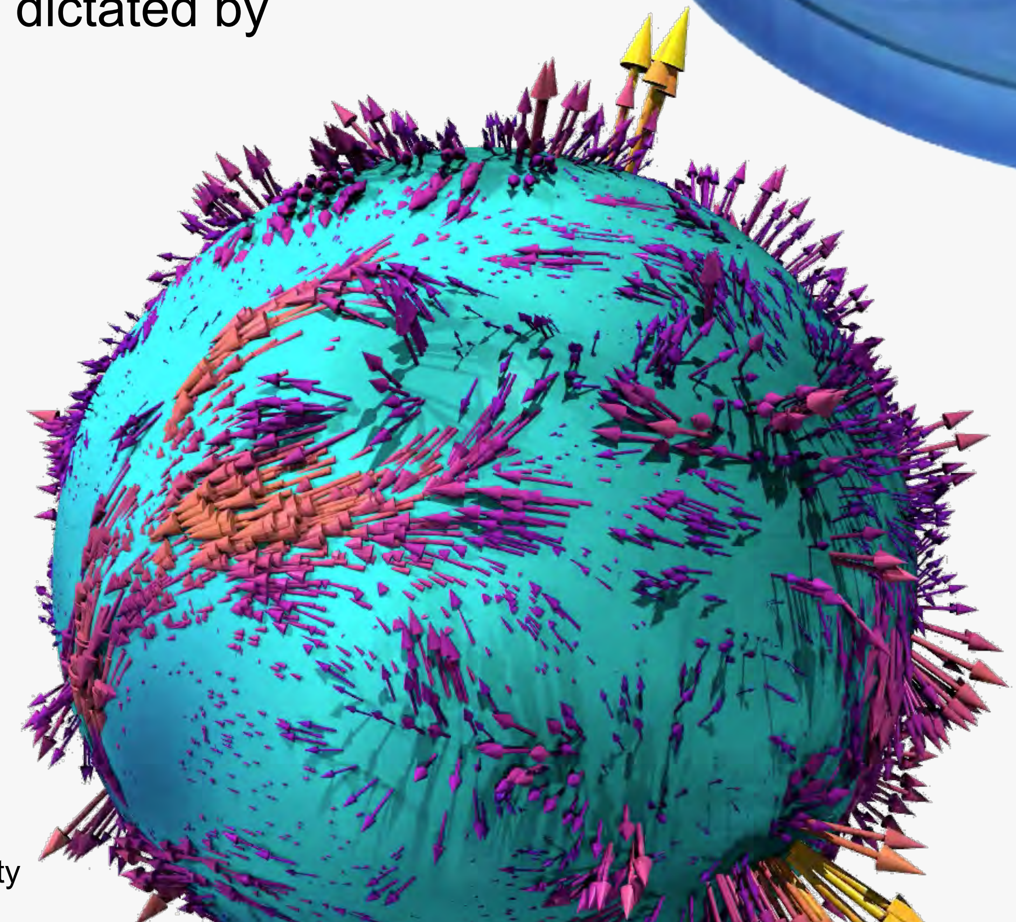
– 3D location on mesh

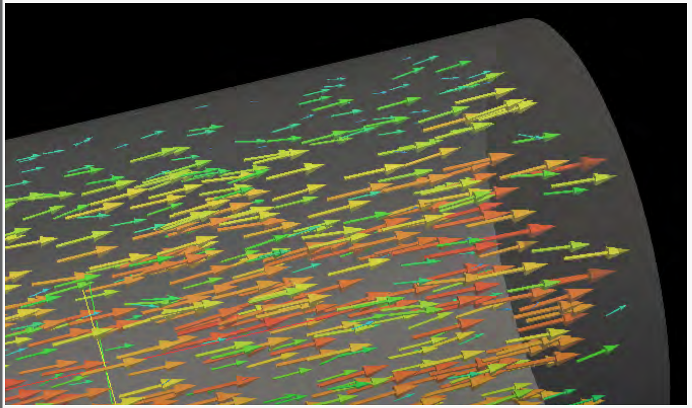
– 2D position in table/graph

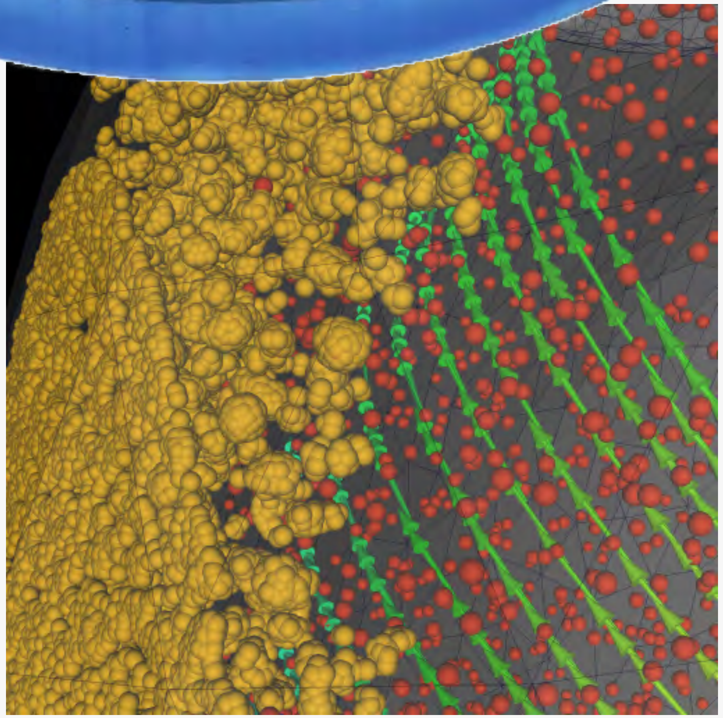
Attributes of graphical entity dictated by attributes of data

– color, size, orientation









Data Representations: Contours (Isosurfaces)

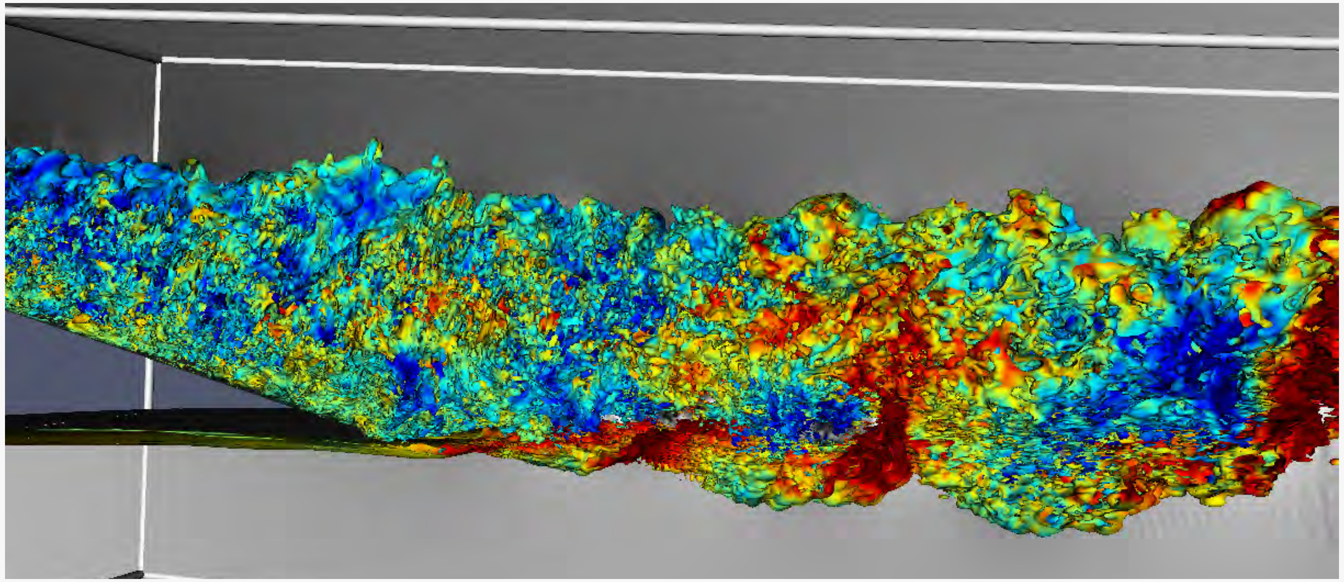
A Line (2D) or Surface (3D), representing a constant value

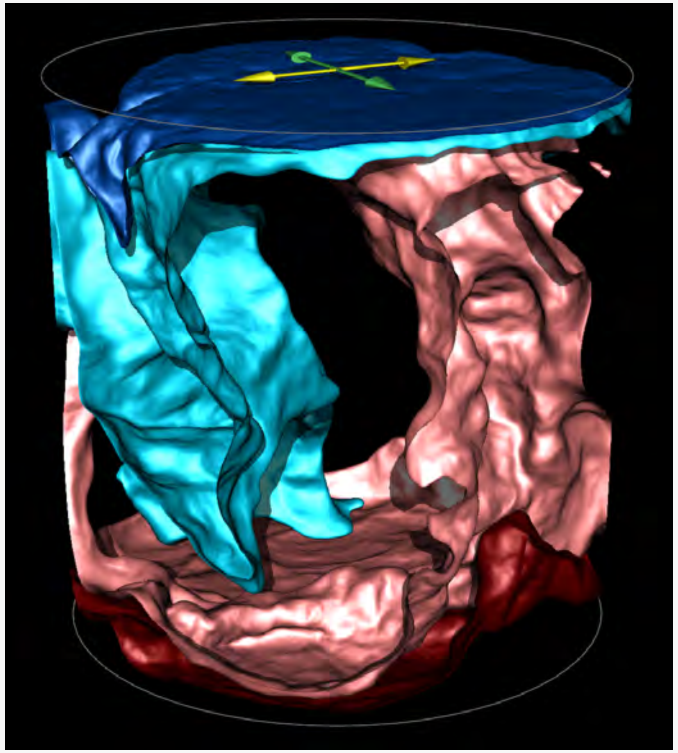
VisIt & ParaView:

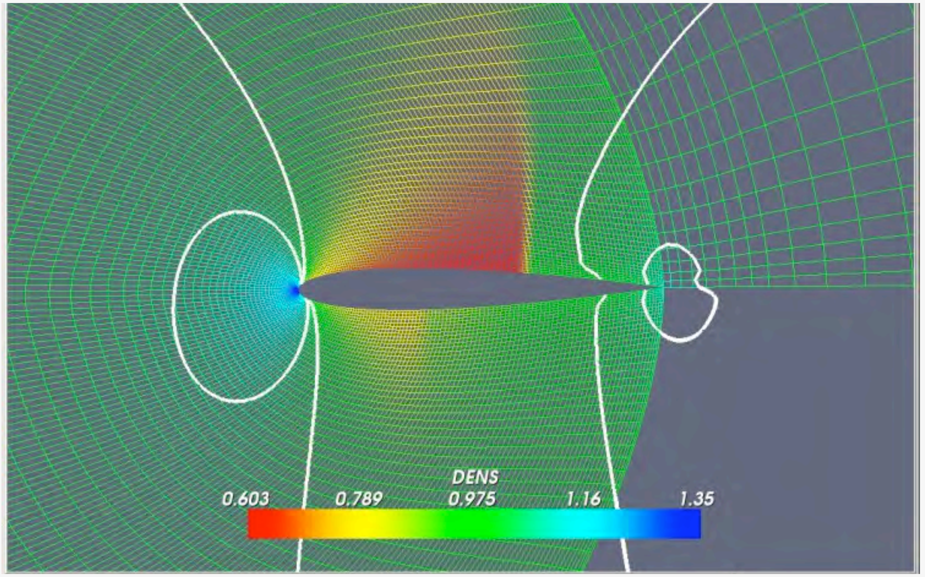
– good at this

vtk:

– same, but again requires more effort

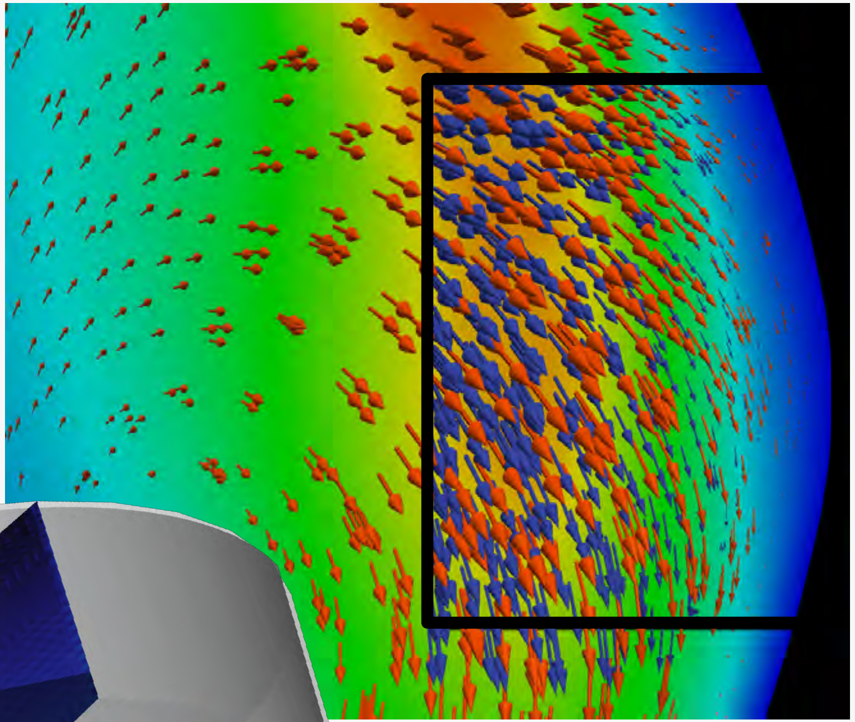


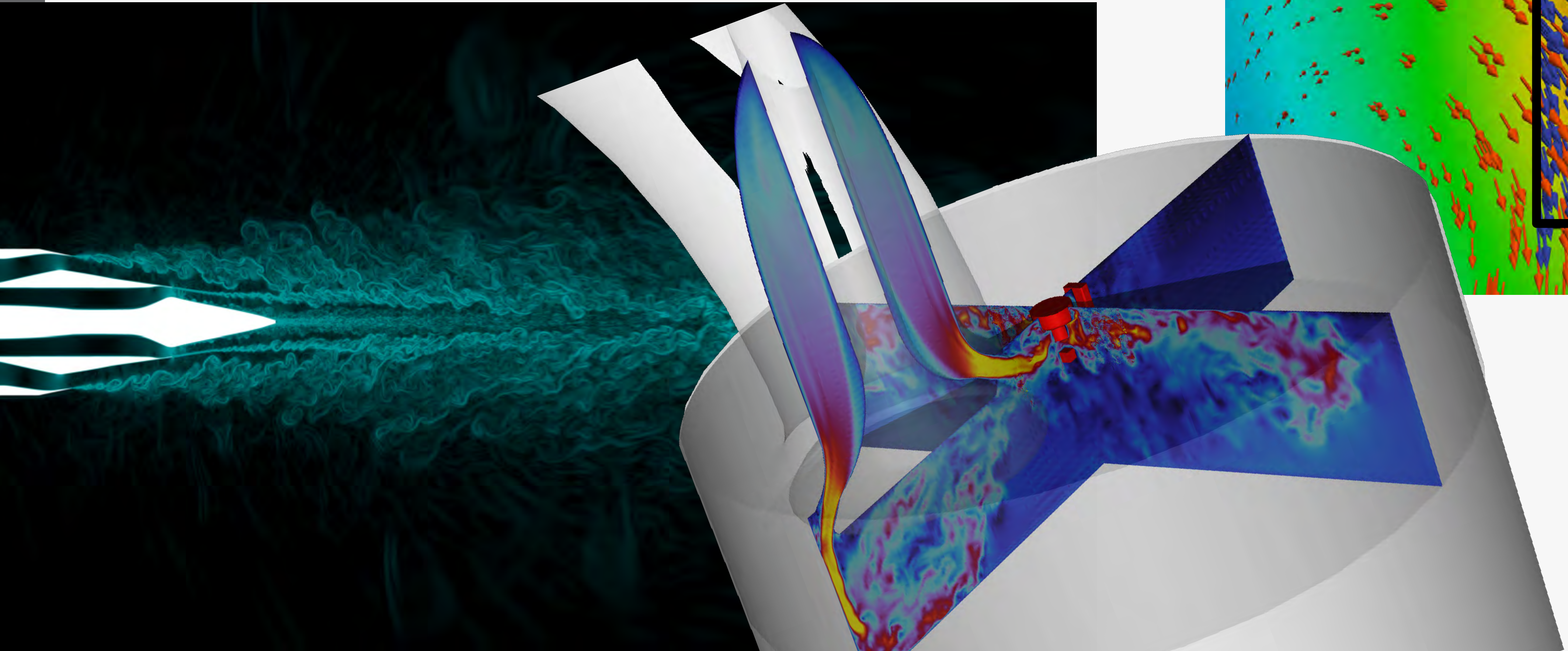




Data Representations: Cutting Planes

- 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



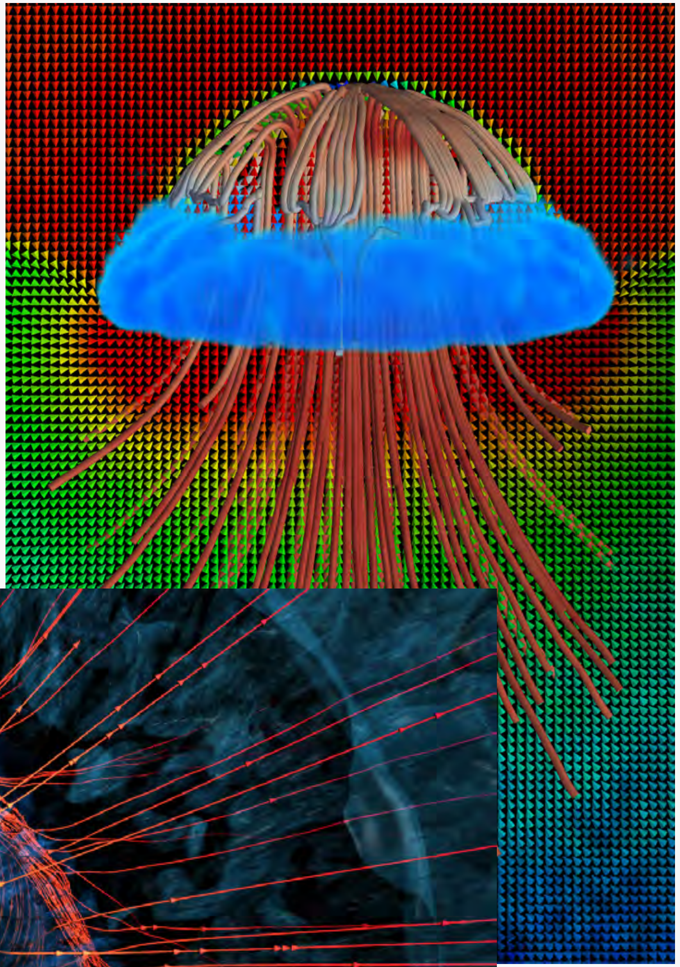


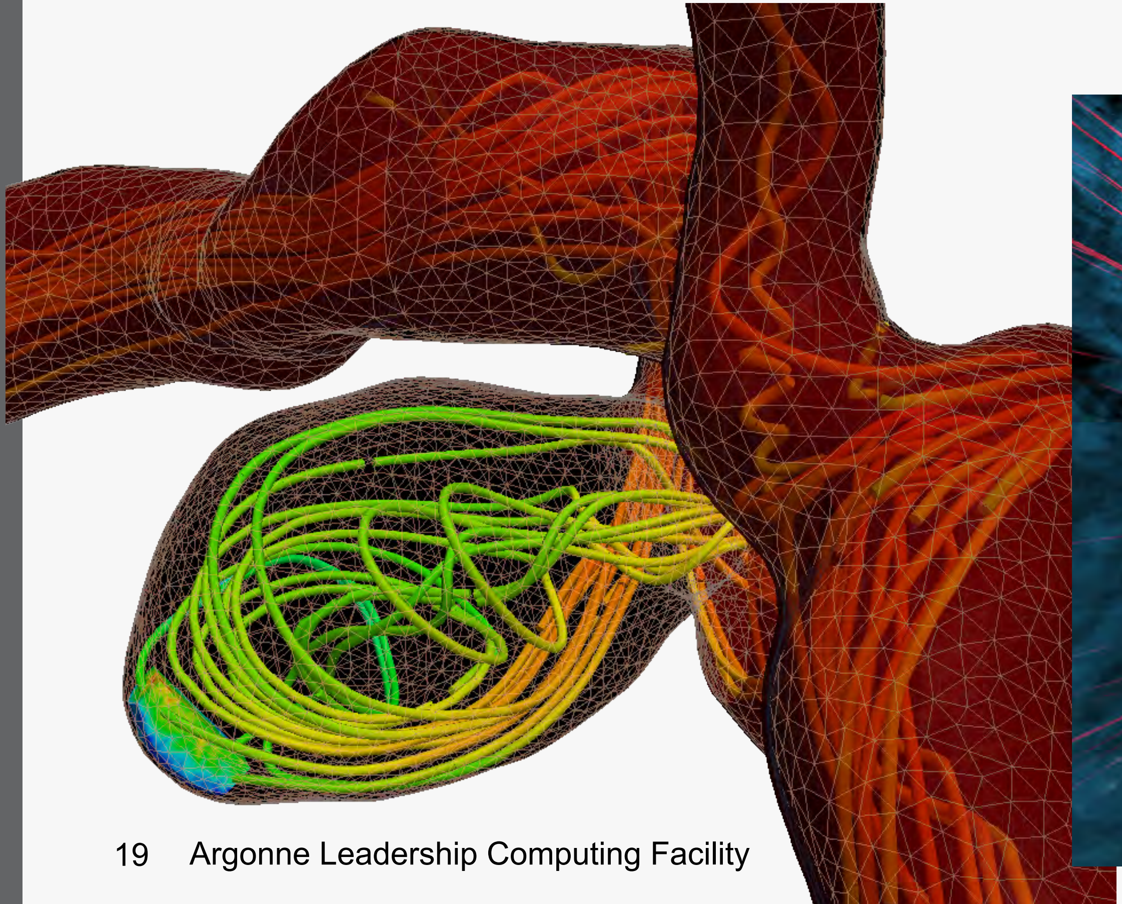
Data Representations: Streamlines

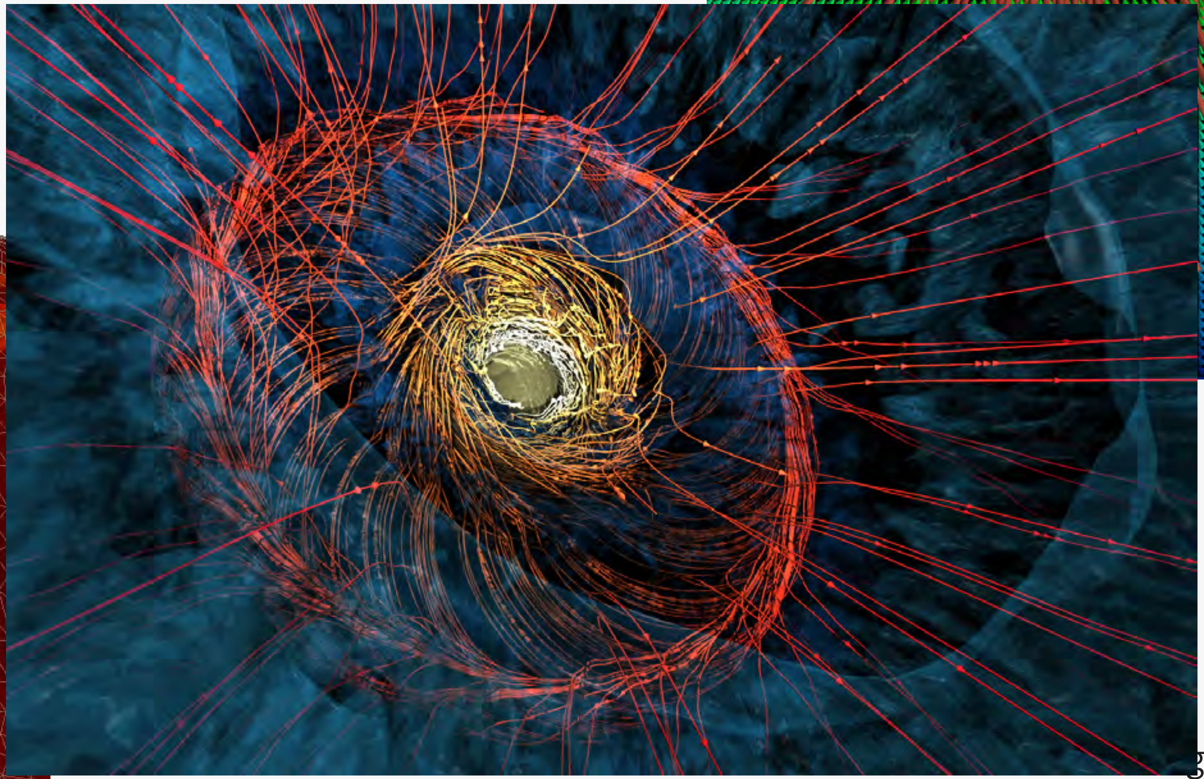
From vector field on a mesh (needs connectivity)

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

VisIt & ParaView & vtk good at this





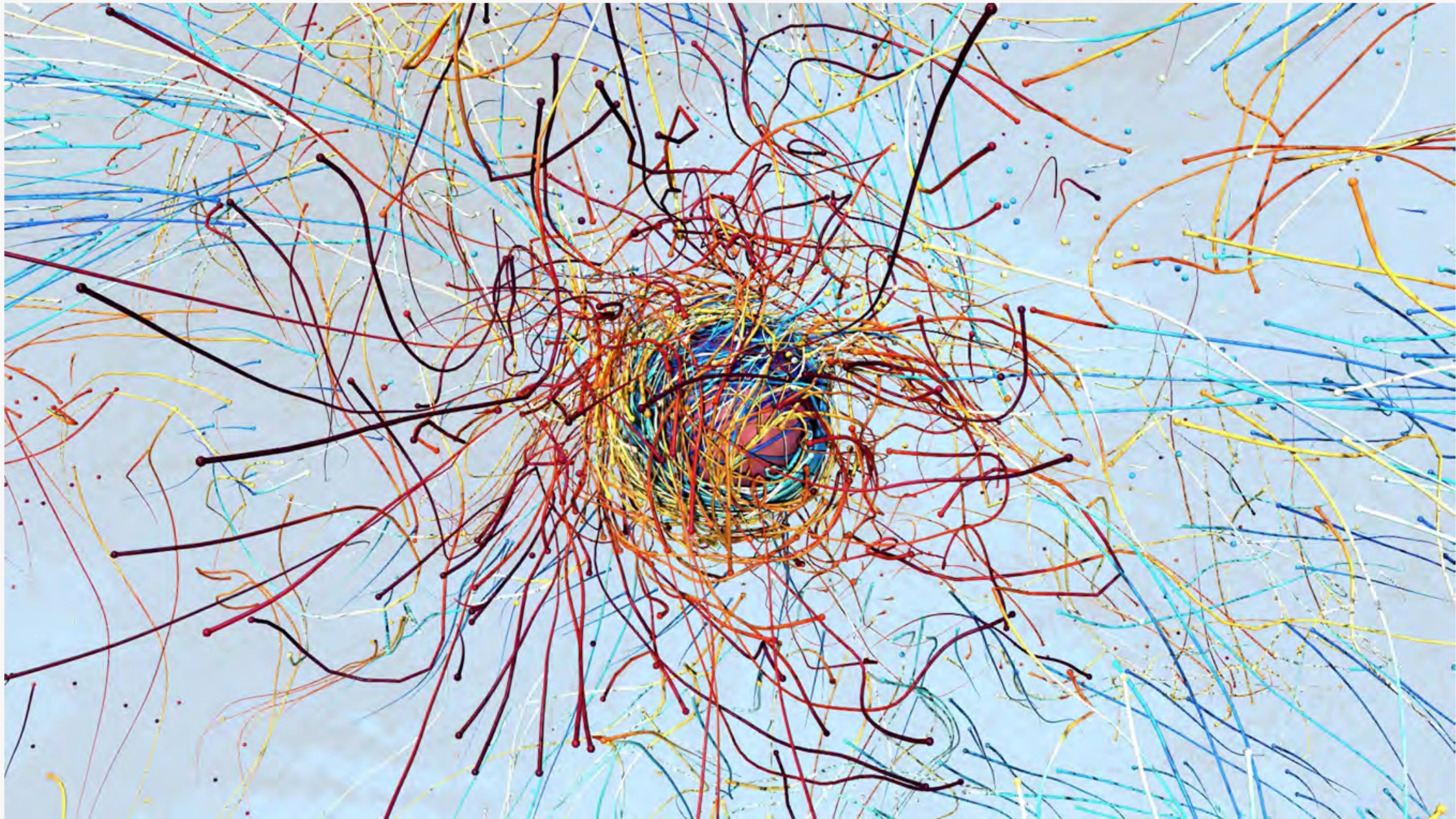


Data Representations: Pathlines

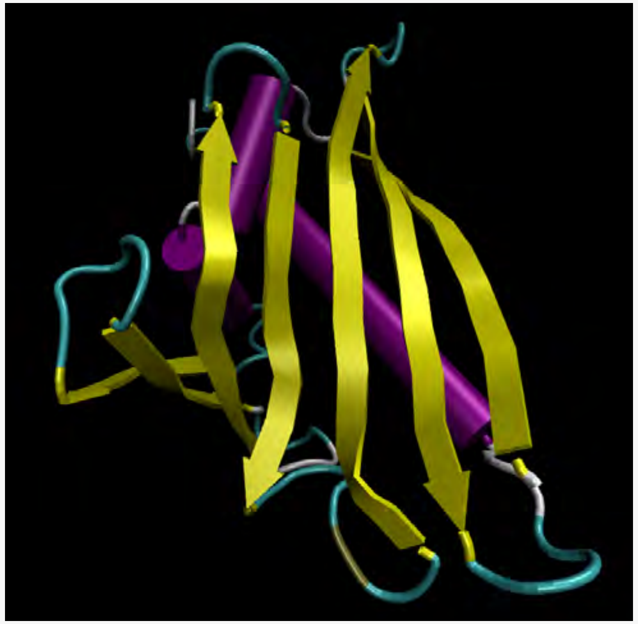
From vector field on a mesh (needs connectivity)

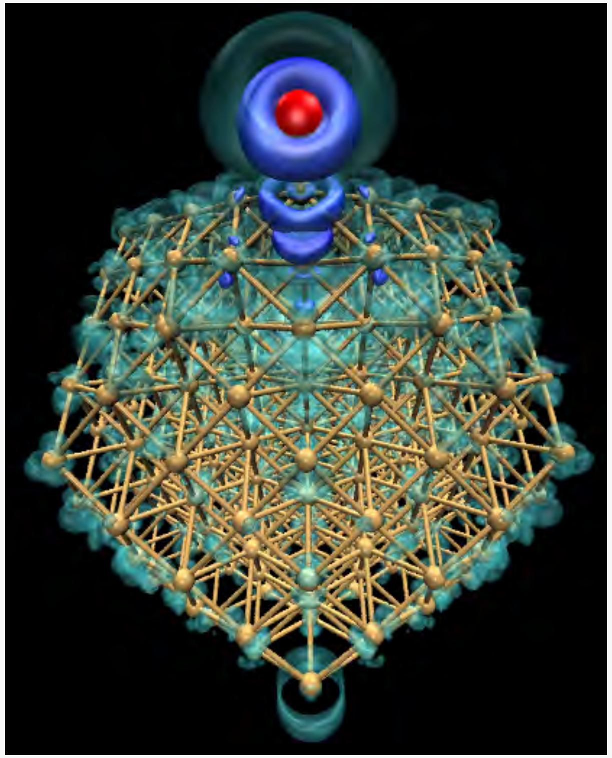
– Trace the path an element will travel over time.

VisIt & ParaView & vtk good at this



Molecular Dynamics Visualization



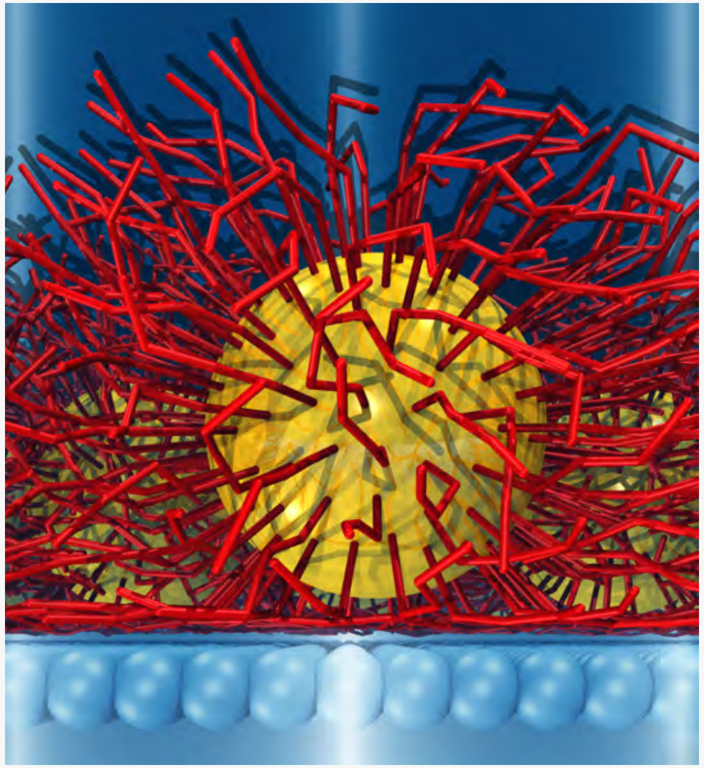


VMD:

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

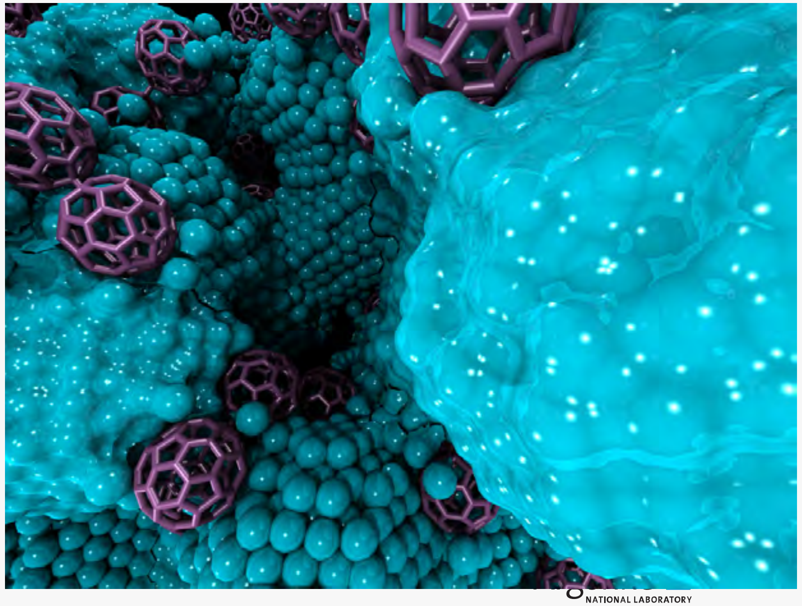
VisIt & ParaView:

– Limited support for these types of representations, but improving



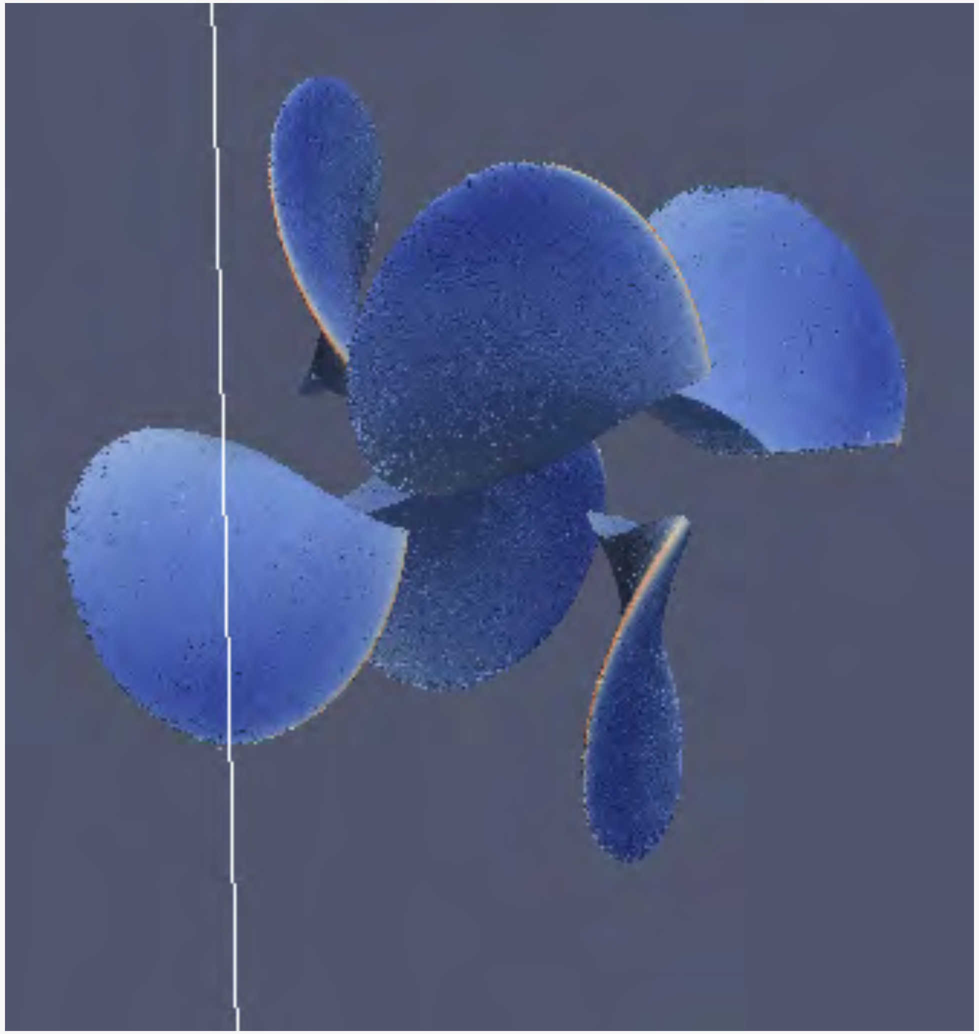
VTK:

– Anything's possible if you try hard enough

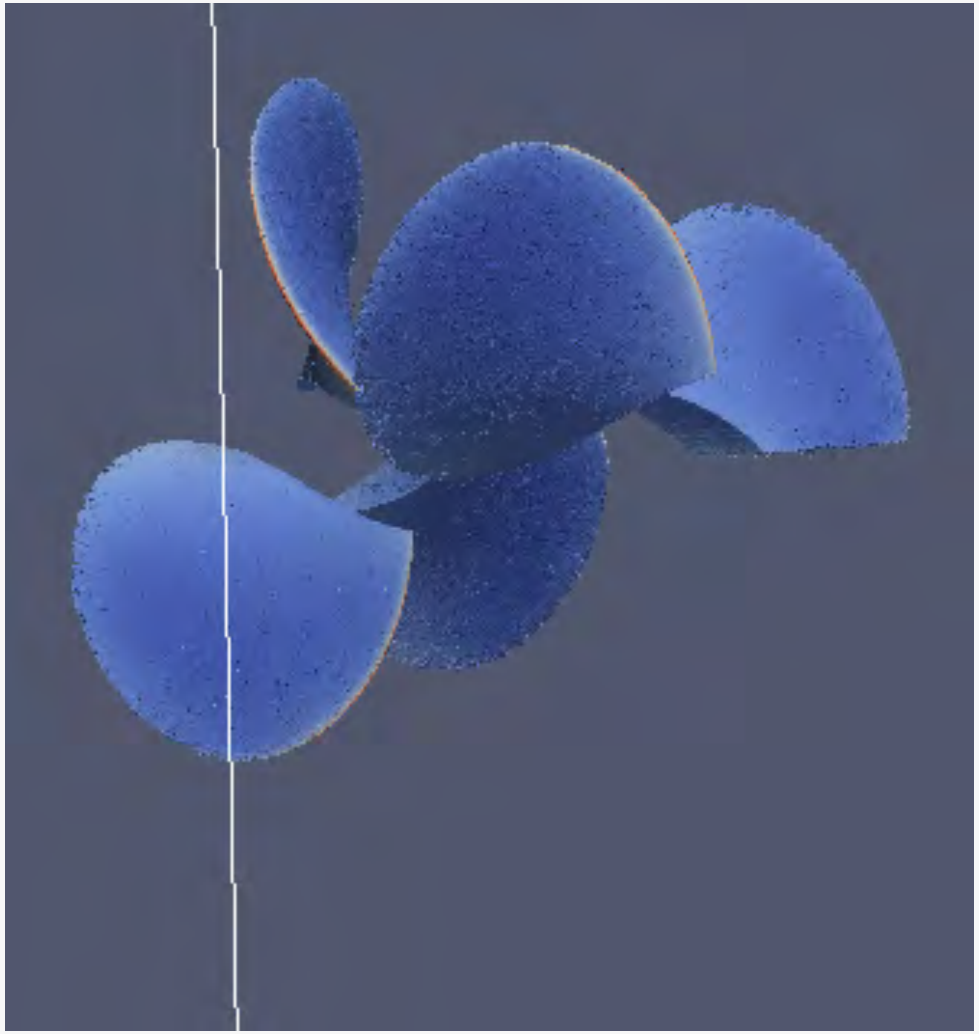


Visualization for Debugging

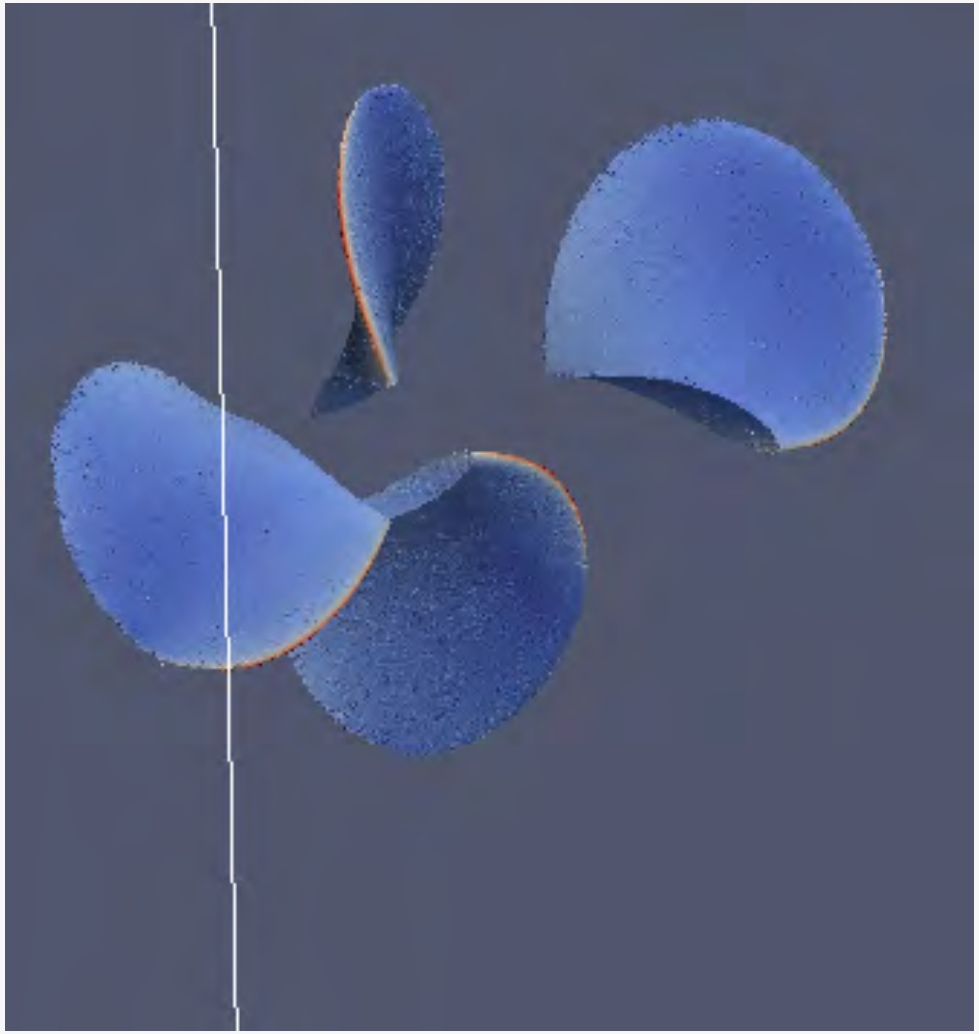
Visualization for Debugging



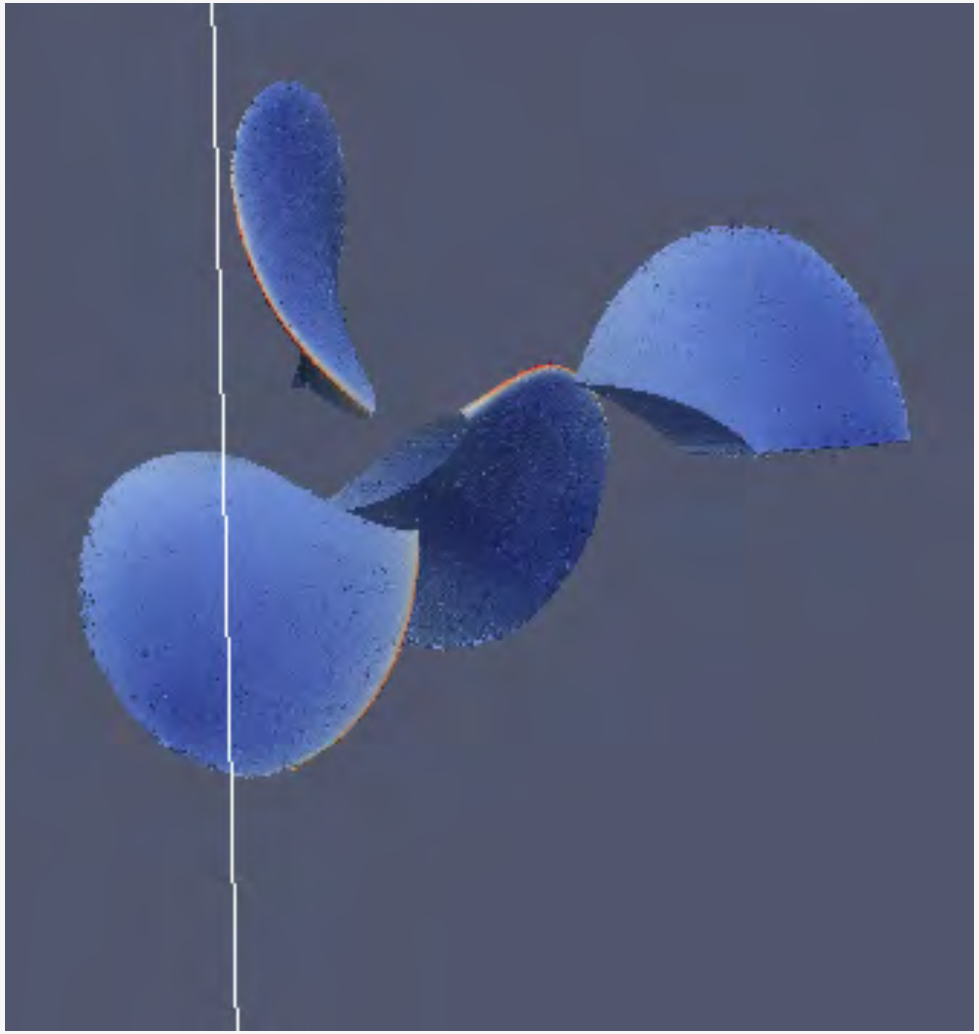
Visualization for Debugging



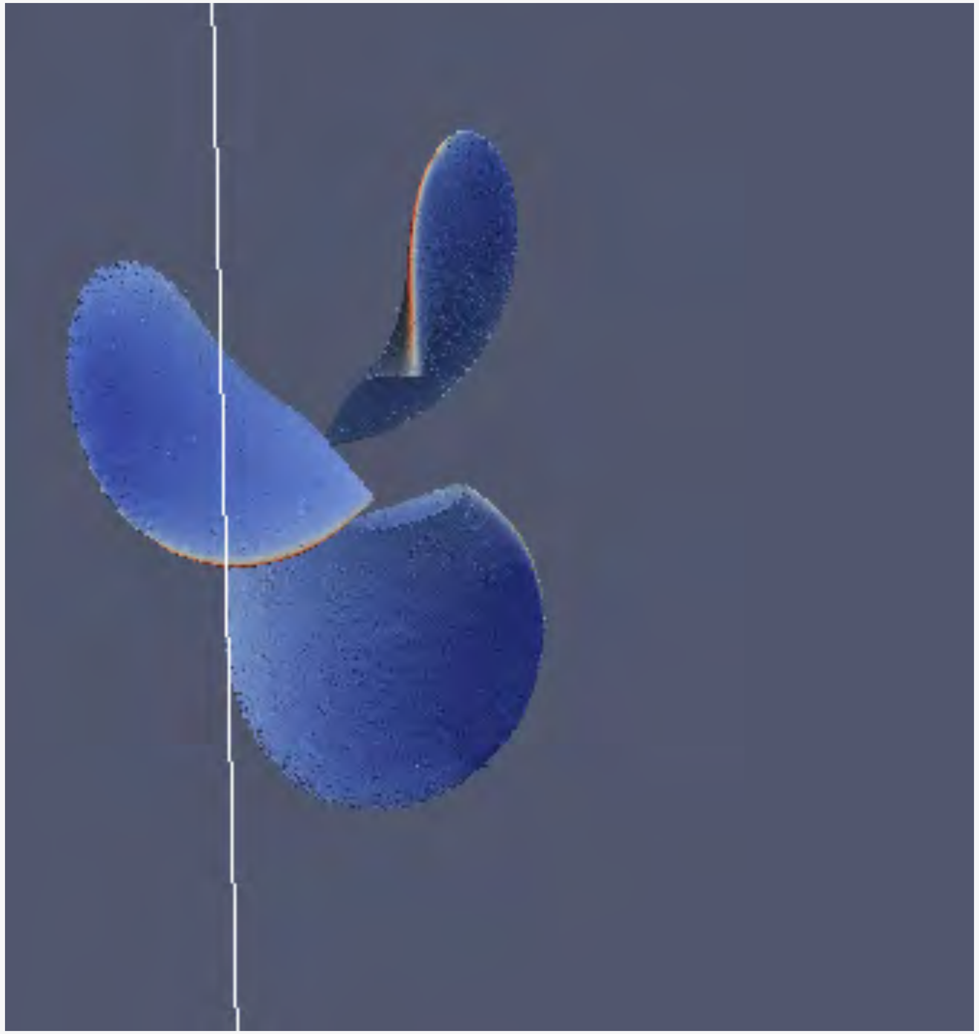
Visualization for Debugging



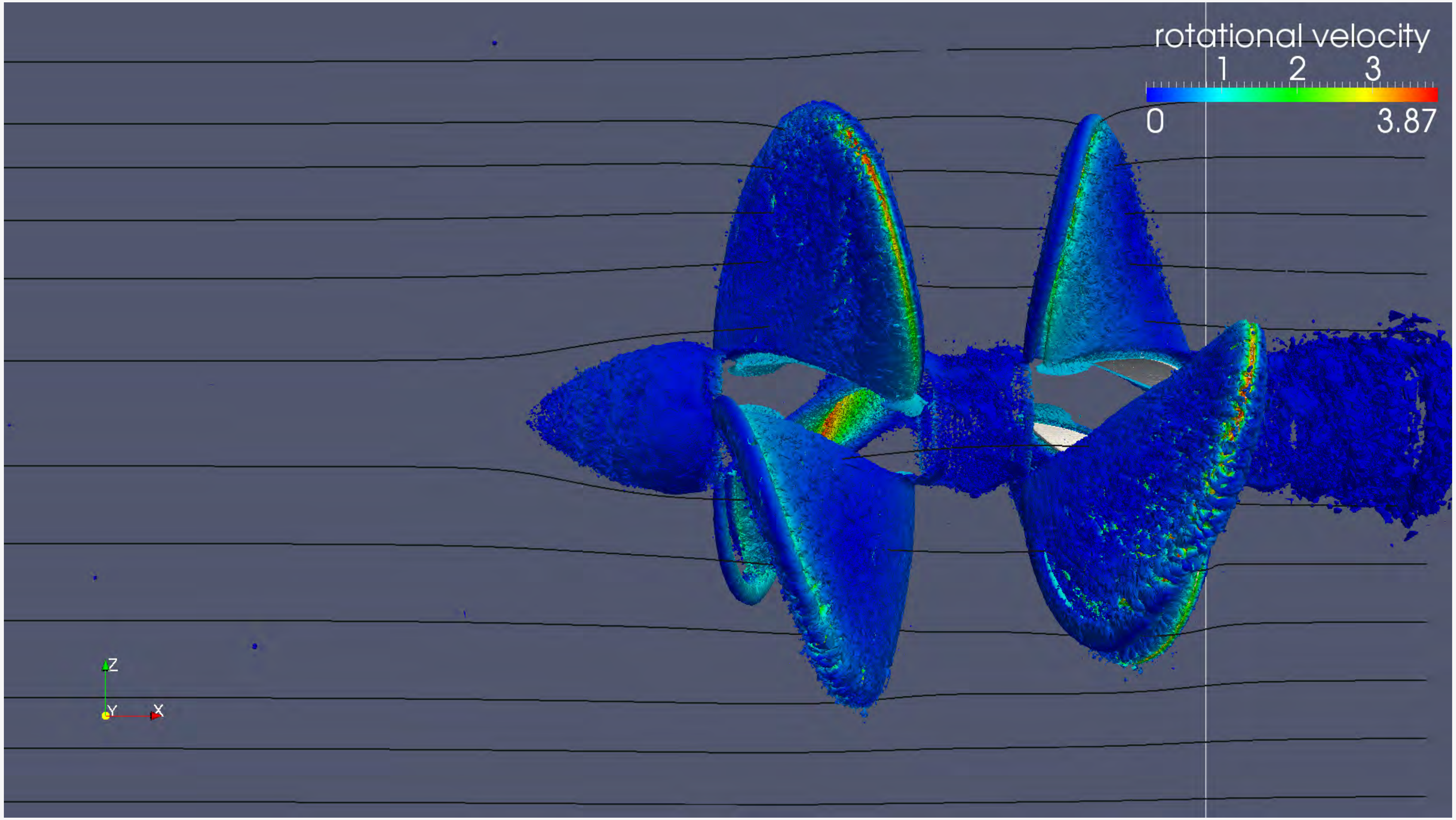
Visualization for Debugging



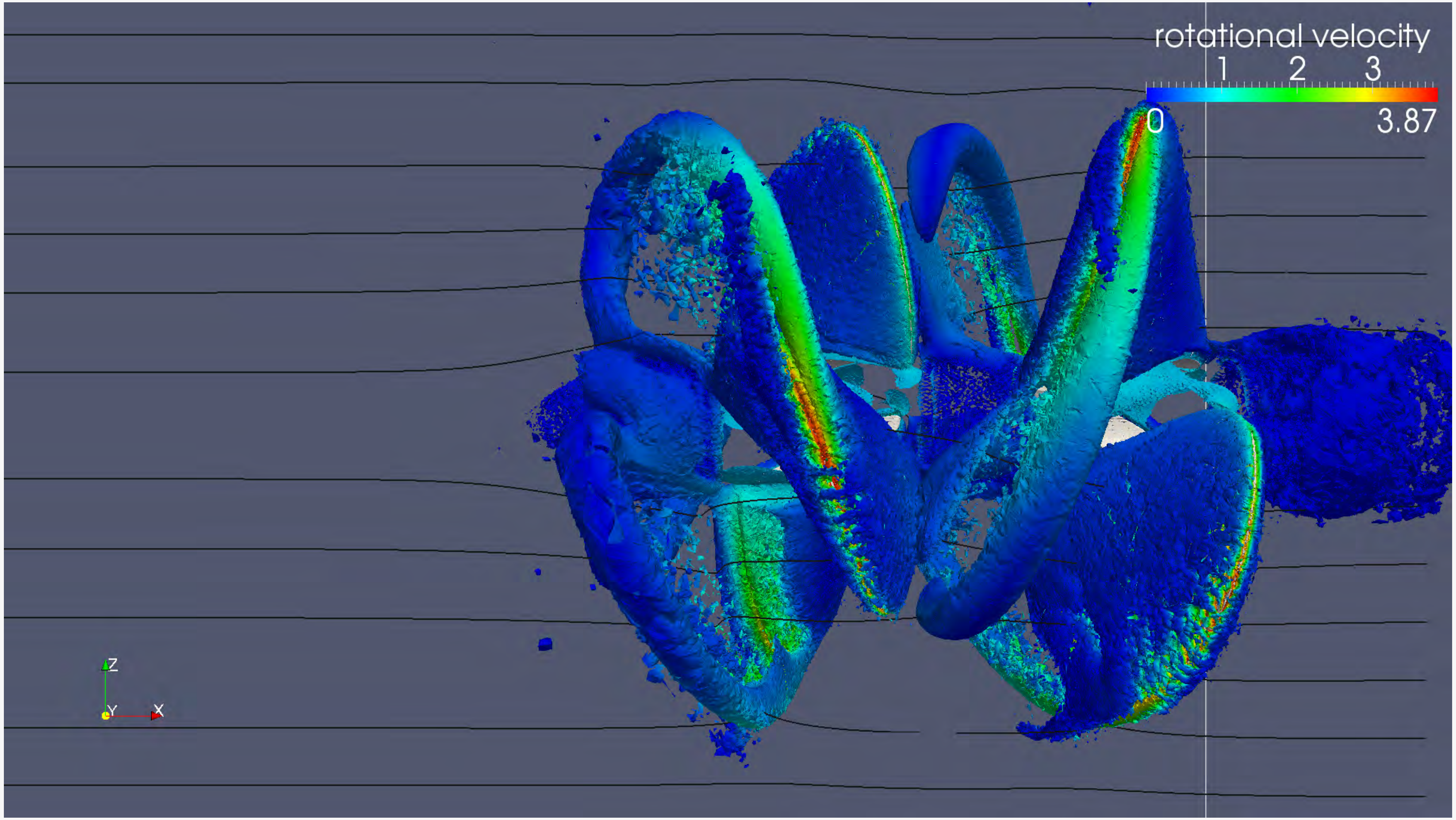
Visualization for Debugging



Visualization for Debugging

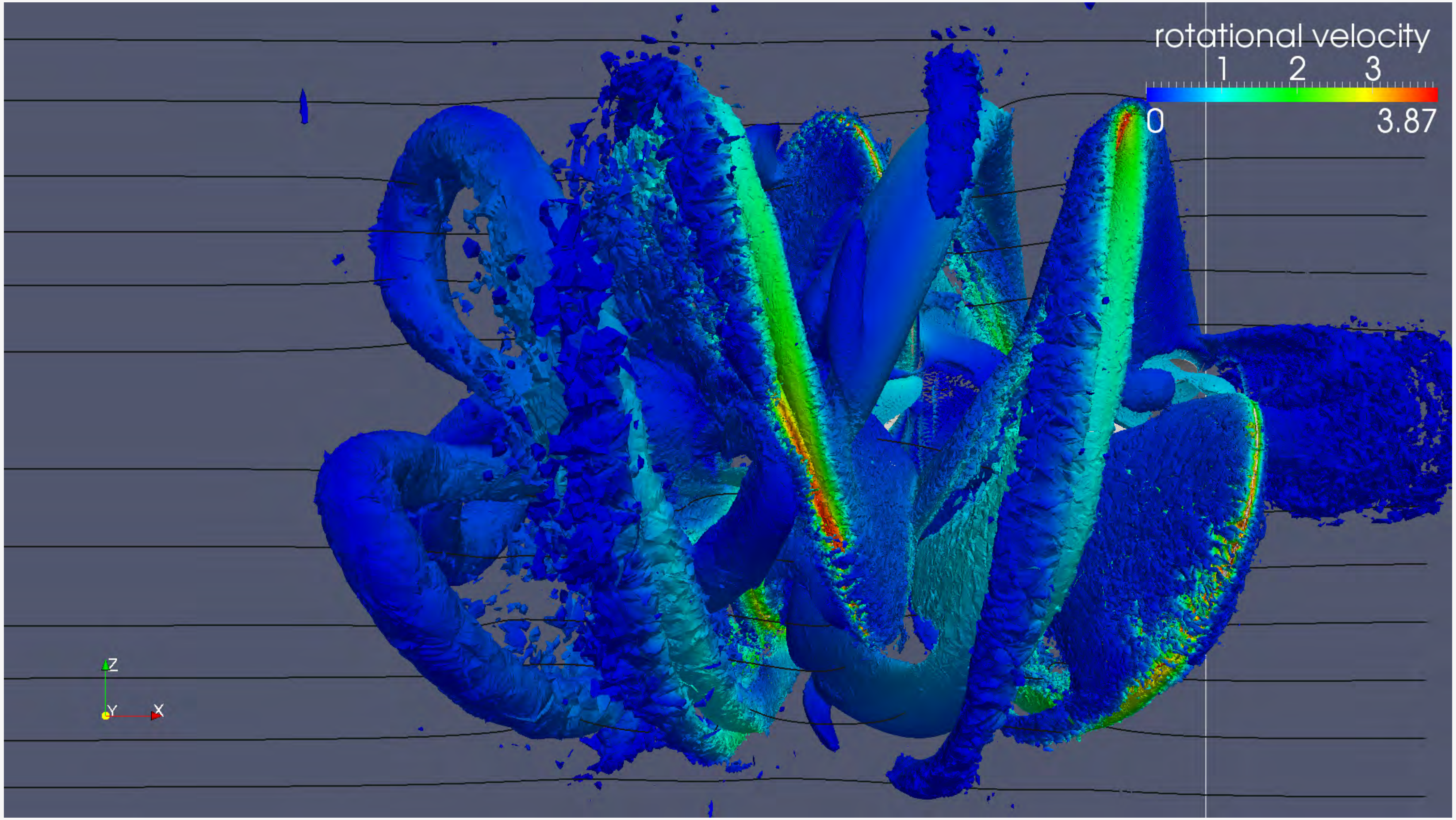


Visualization for Debugging

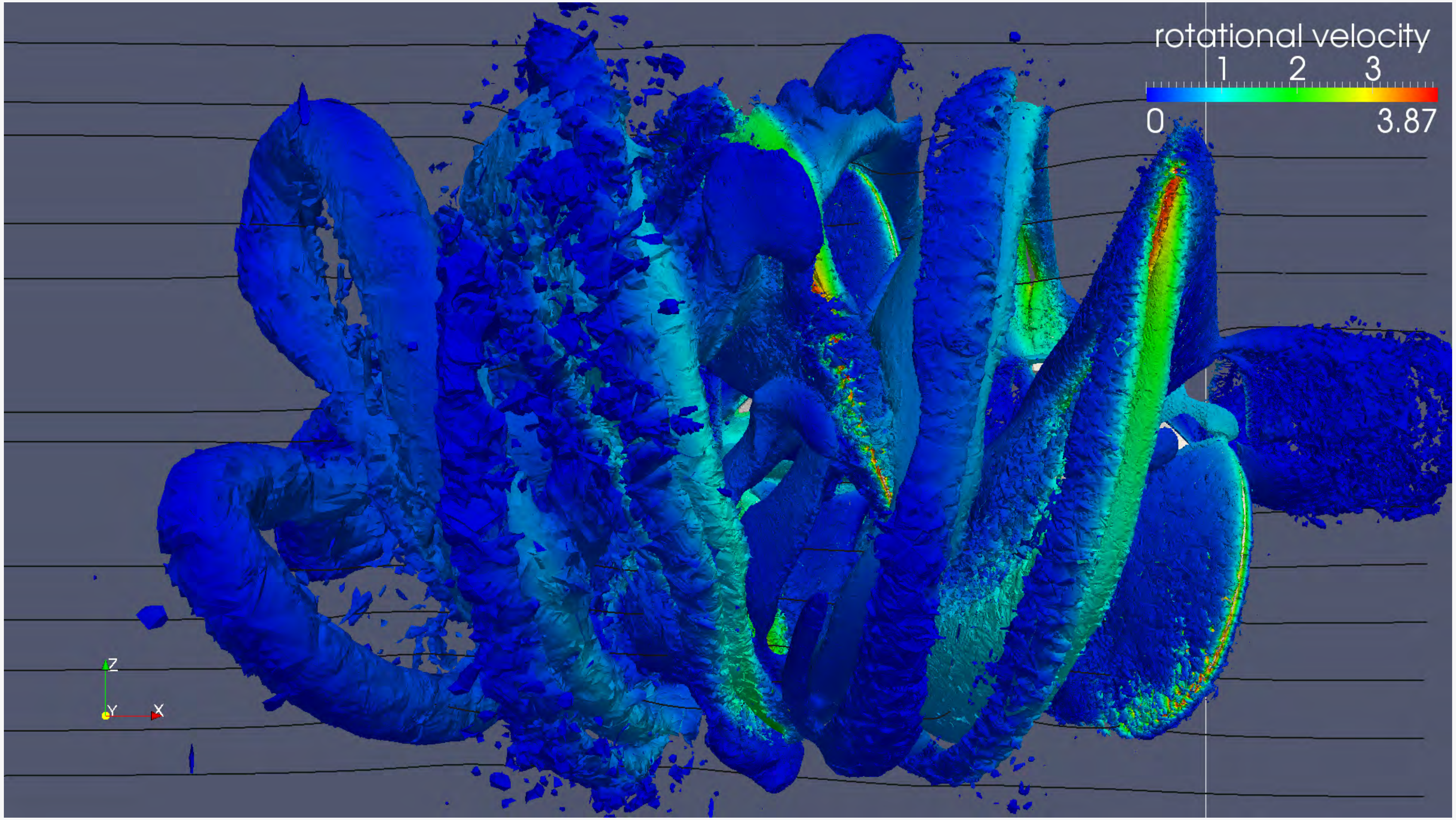


Argonne NATIONAL LABORATORY

Visualization for Debugging

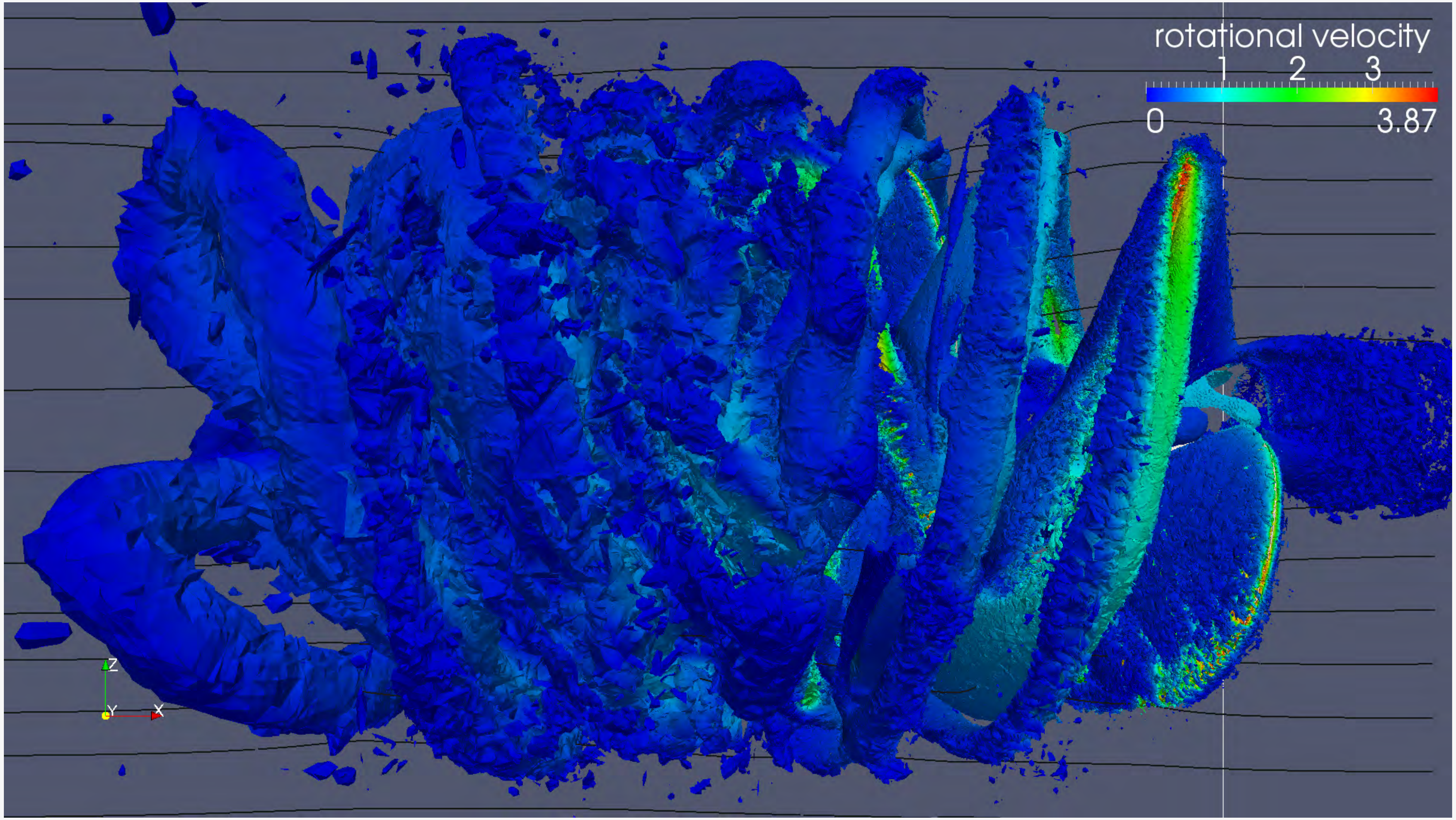


Visualization for Debugging

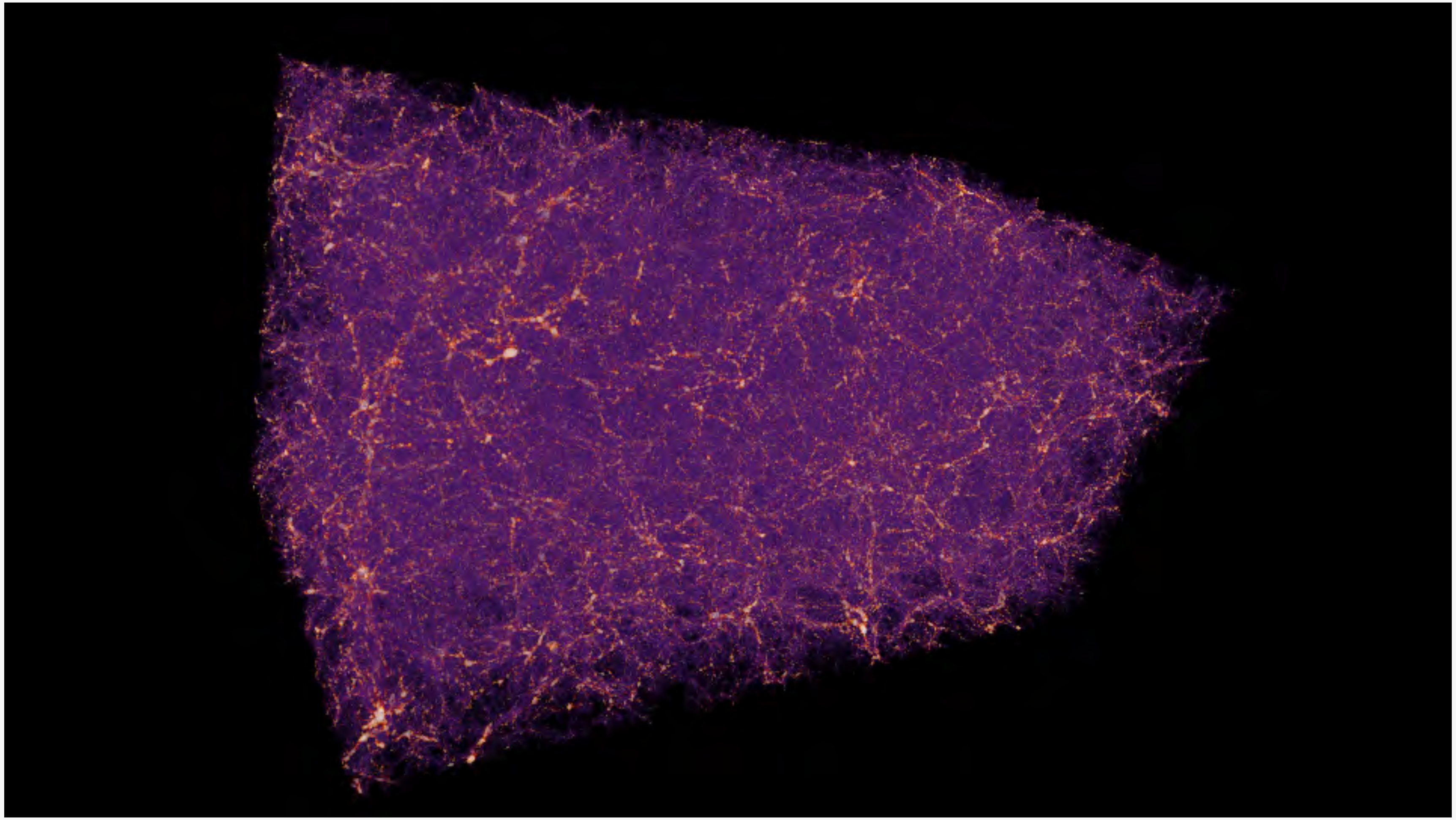


Argonne NATIONAL LABORATORY

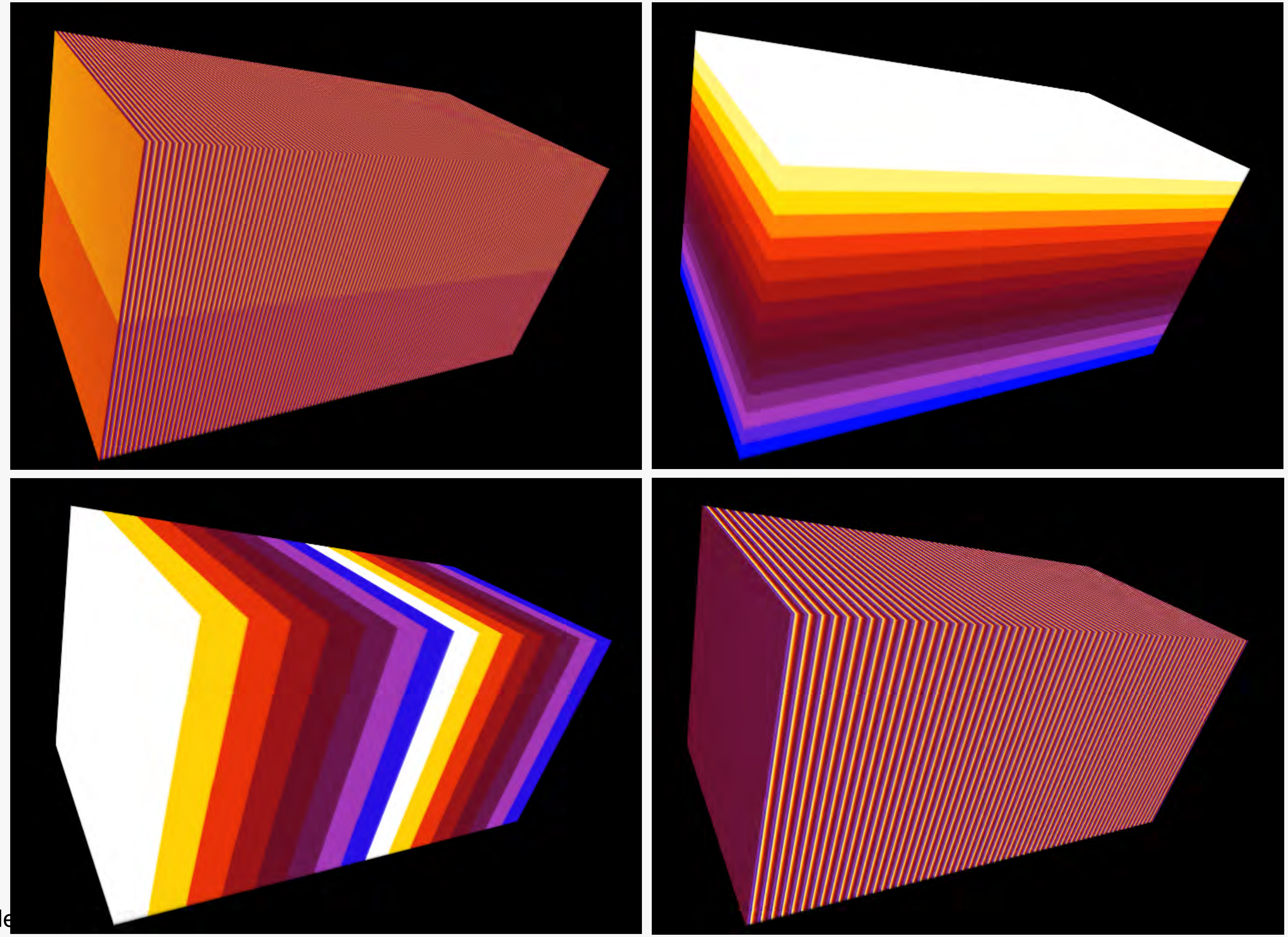
Visualization for Debugging



Visualization as Diagnostics: Color by Thread ID



Visualization as Diagnostics: Color by Thread ID



Argonne NATIONAL LABORATORY

In Situ Visualization and Analysis

Five orders of magnitude between compute and I/O capacity on Titan Cray system at ORNL

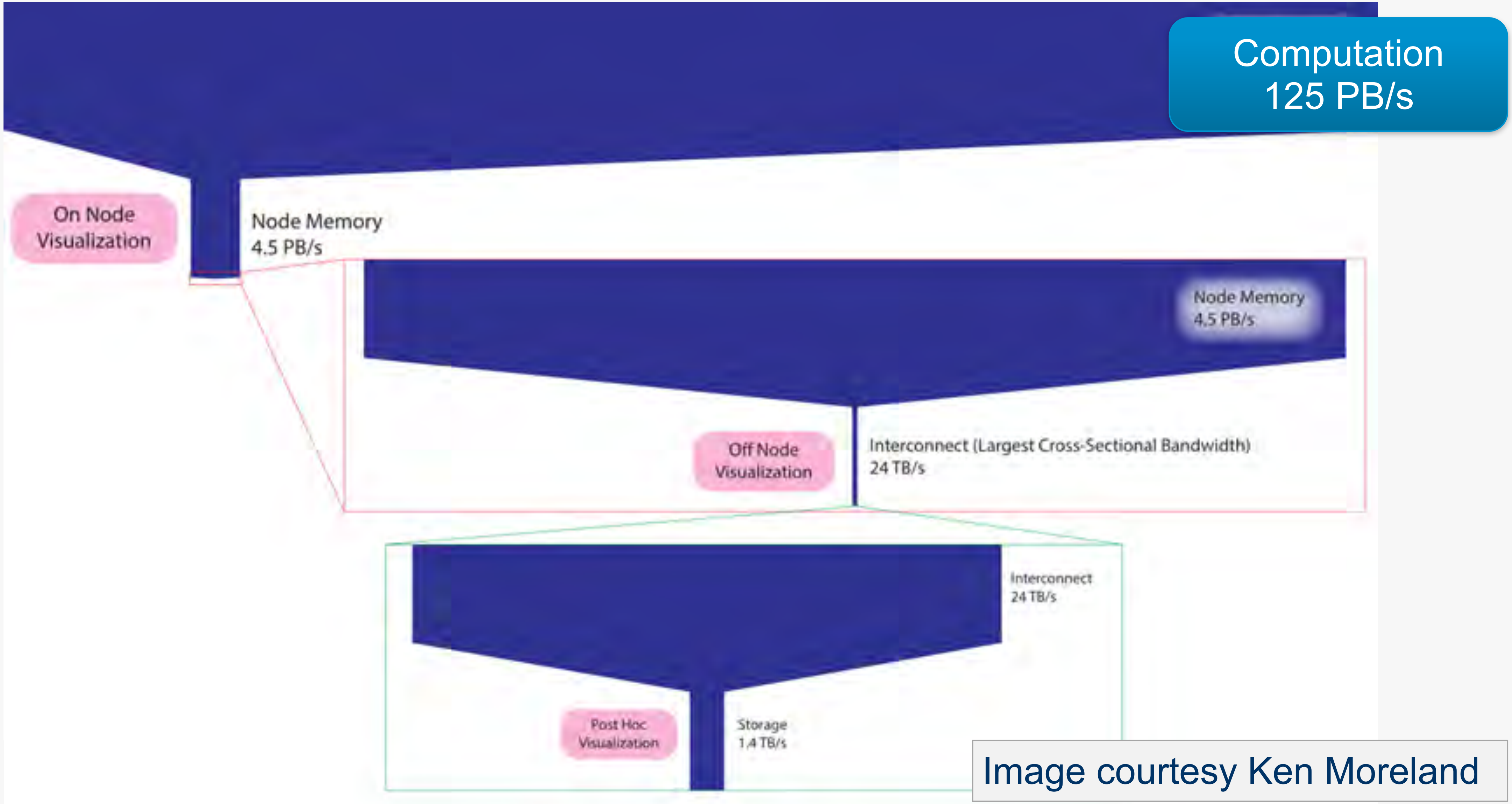


Image courtesy Ken Moreland

Five orders of magnitude between compute and I/O capacity on Titan Cray system at ORNL

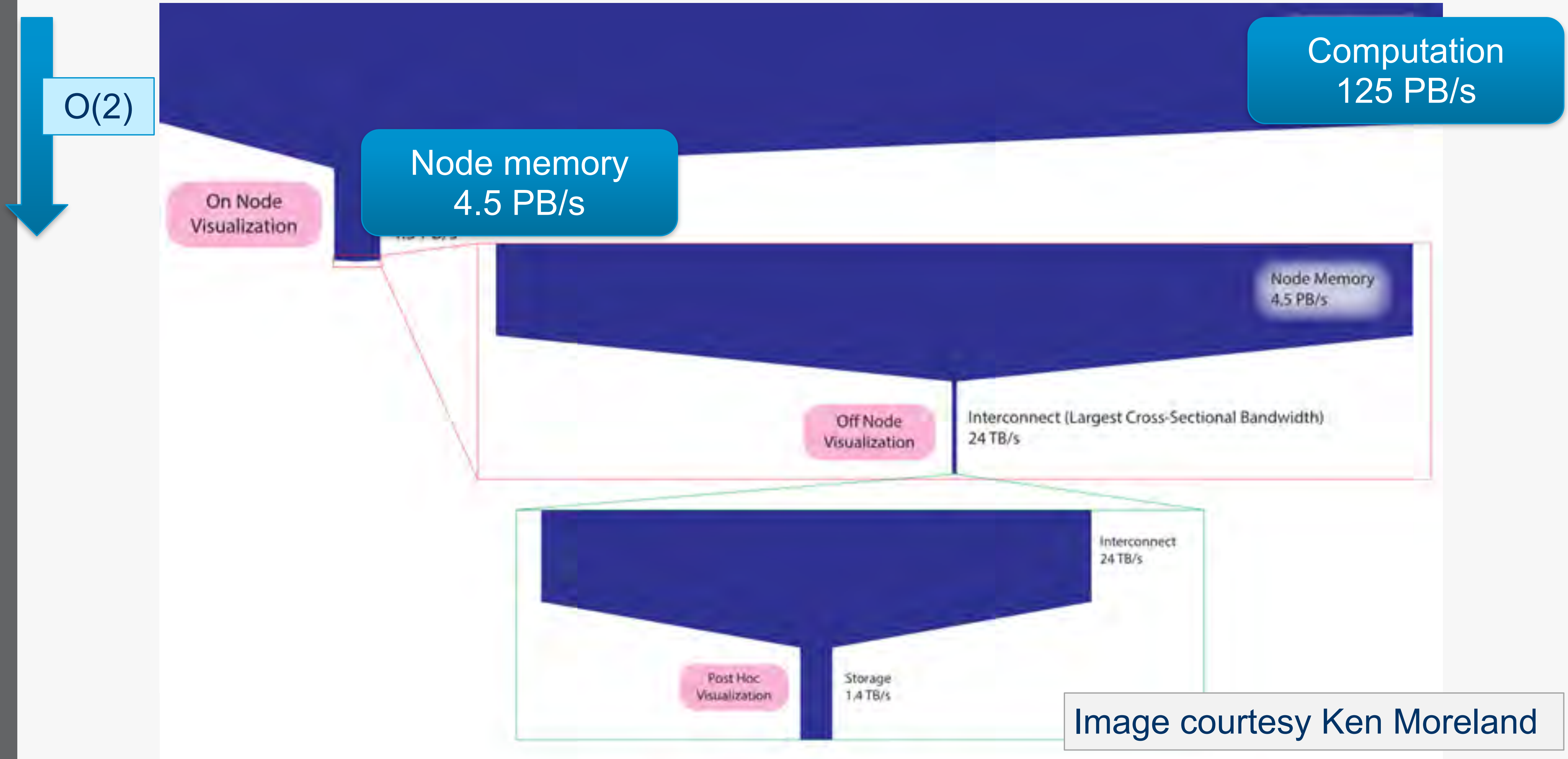


Image courtesy Ken Moreland

Five orders of magnitude between compute and I/O capacity on Titan Cray system at ORNL

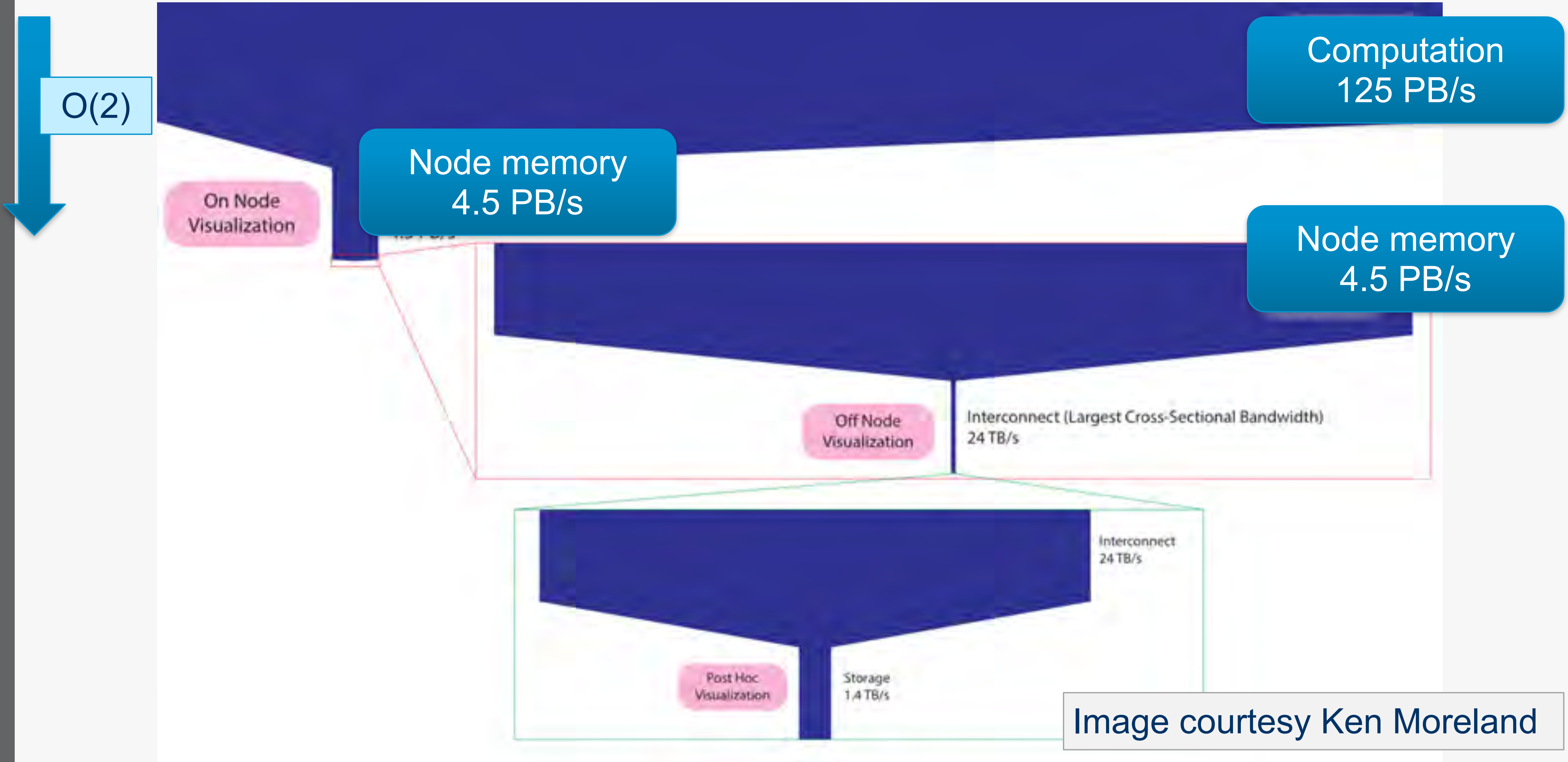


Image courtesy Ken Moreland

Five orders of magnitude between compute and I/O capacity on Titan Cray system at ORNL

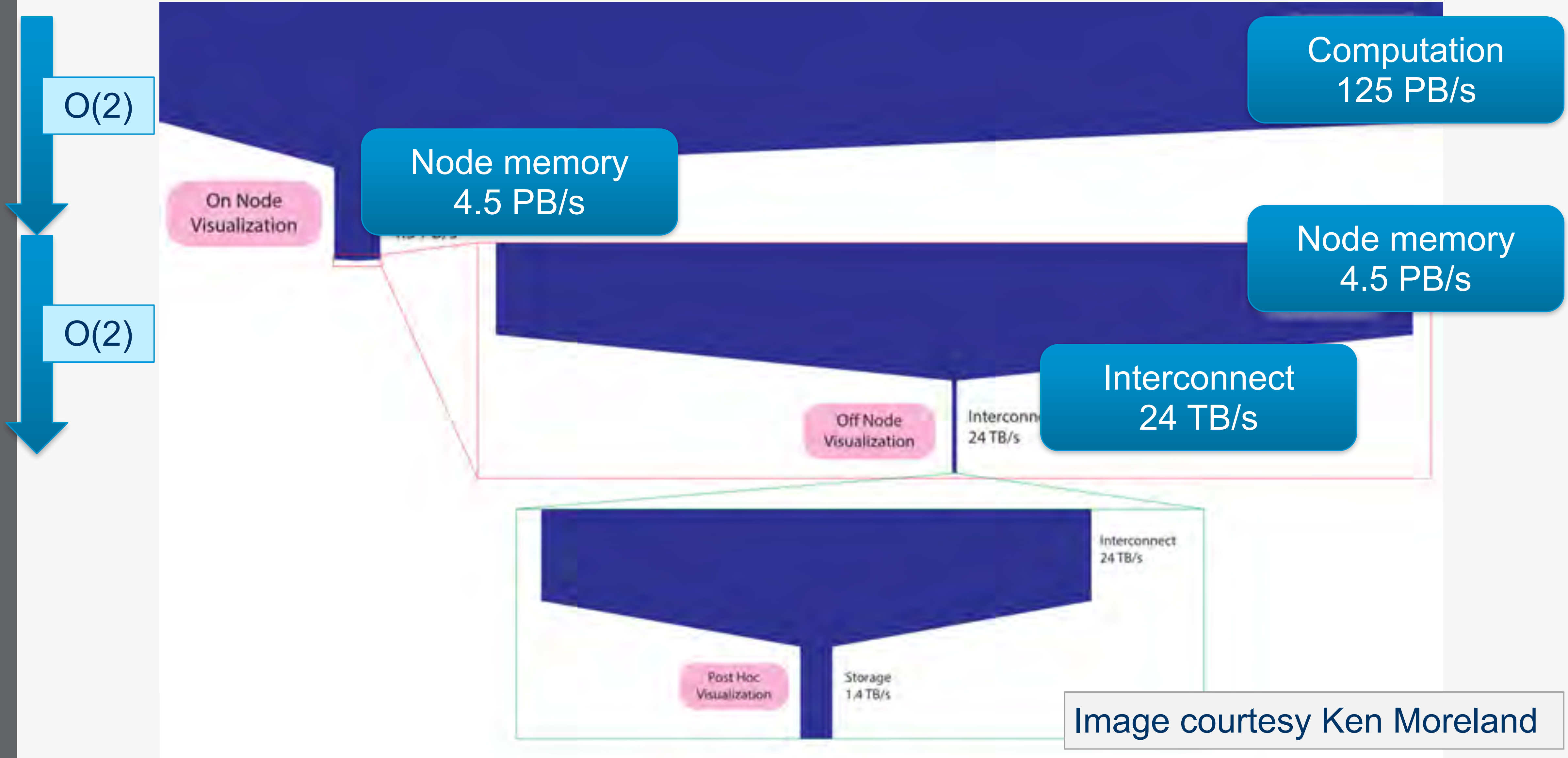


Image courtesy Ken Moreland

Five orders of magnitude between compute and I/O capacity on Titan Cray system at ORNL

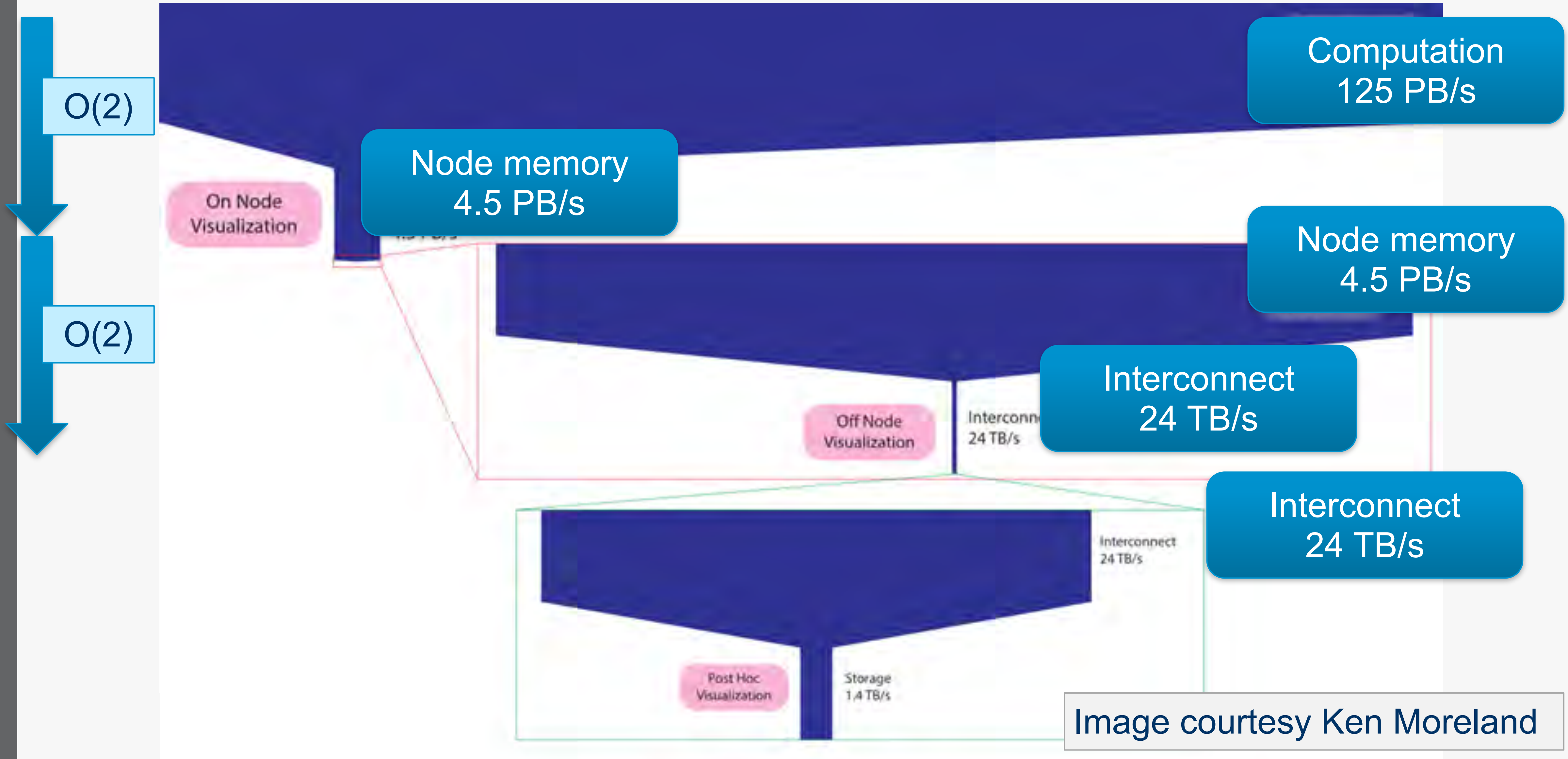


Image courtesy Ken Moreland

Five orders of magnitude between compute and I/O capacity on Titan Cray system at ORNL

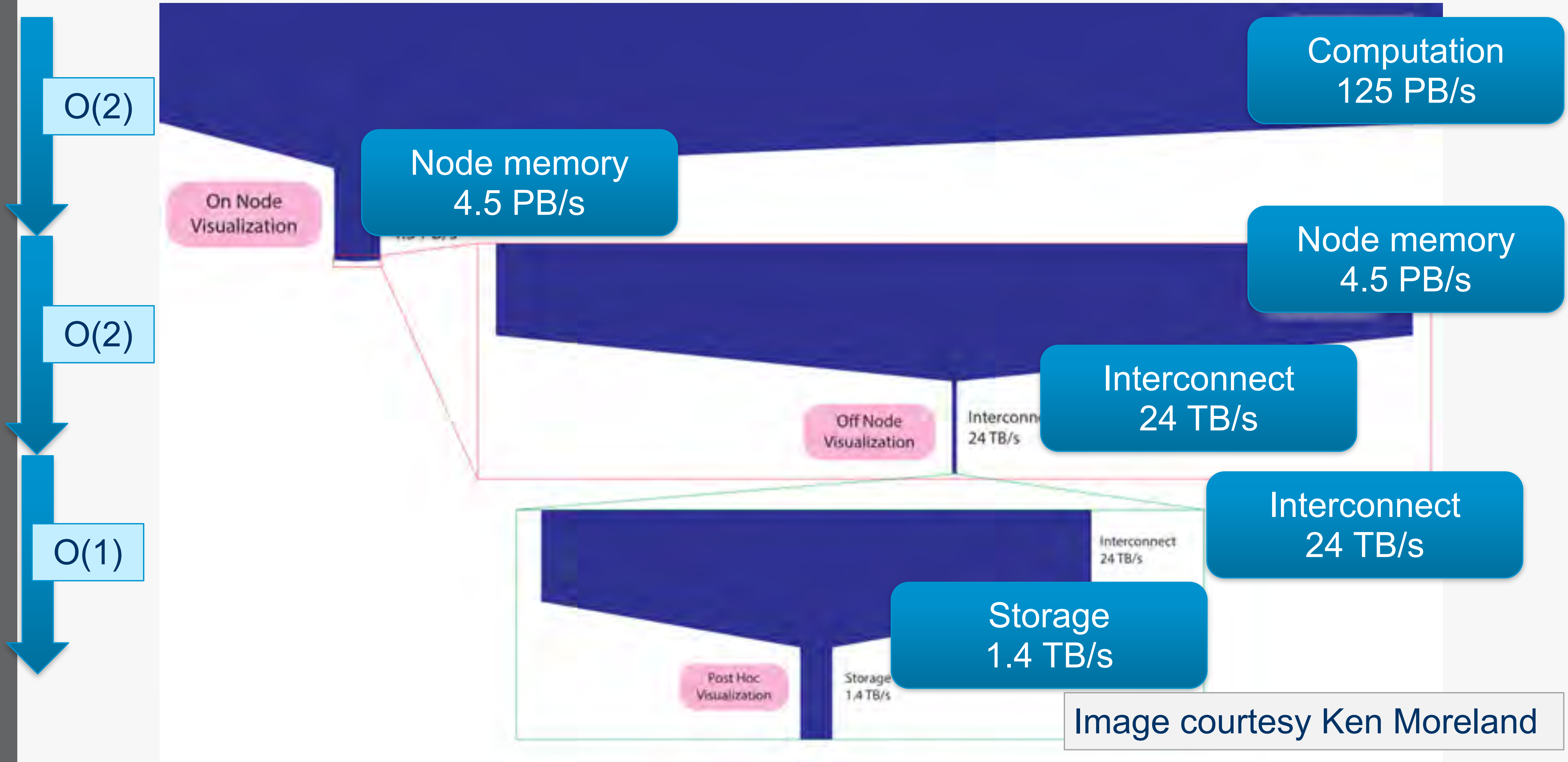


Image courtesy Ken Moreland

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

Argonne NATIONAL LABORATORY

Two Frameworks for In Situ Vis and Analysis at ALCF

SENSEI
in situ

- "Write once, run everywhere" design
- Data model based on VTK from Kitware
- Supports a variety of backends, including ParaView/Catalyst, VisIt/LibSim, ADIOS, Python

Ascent

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

Instrumenting Simulation Codes

SENSEI
in situ

Ascent

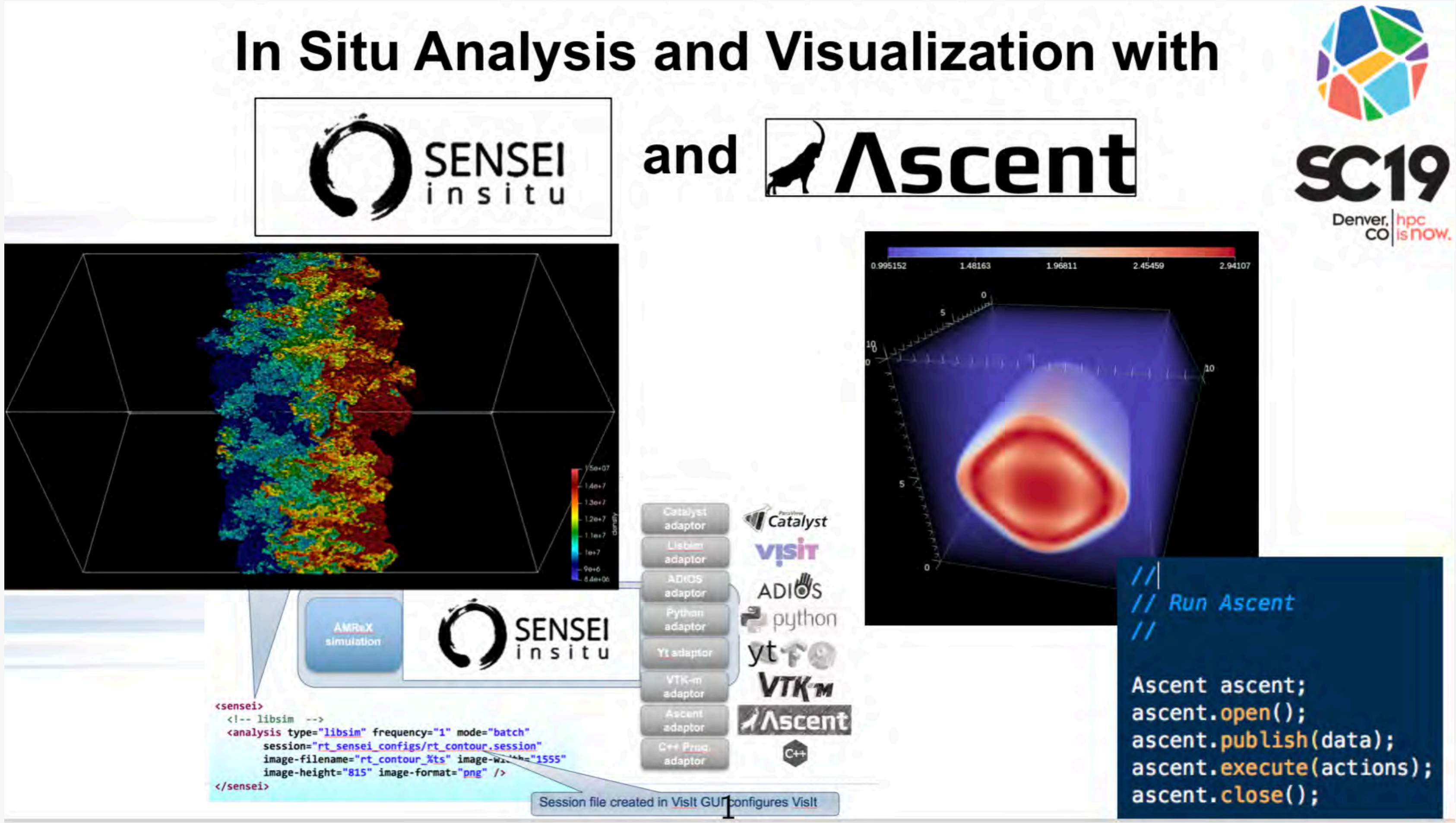
```
initialize sim
if do_insitu bridge::initialize
do
  compute new state
  if do_io write plot file
  if do_insitu bridge::execute
while !done
if do_insitu bridge::finalize
finalize sim
```

```
//
// Run Ascent
//

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

SENSEI + ASCENT tutorial at SC19 and SC20

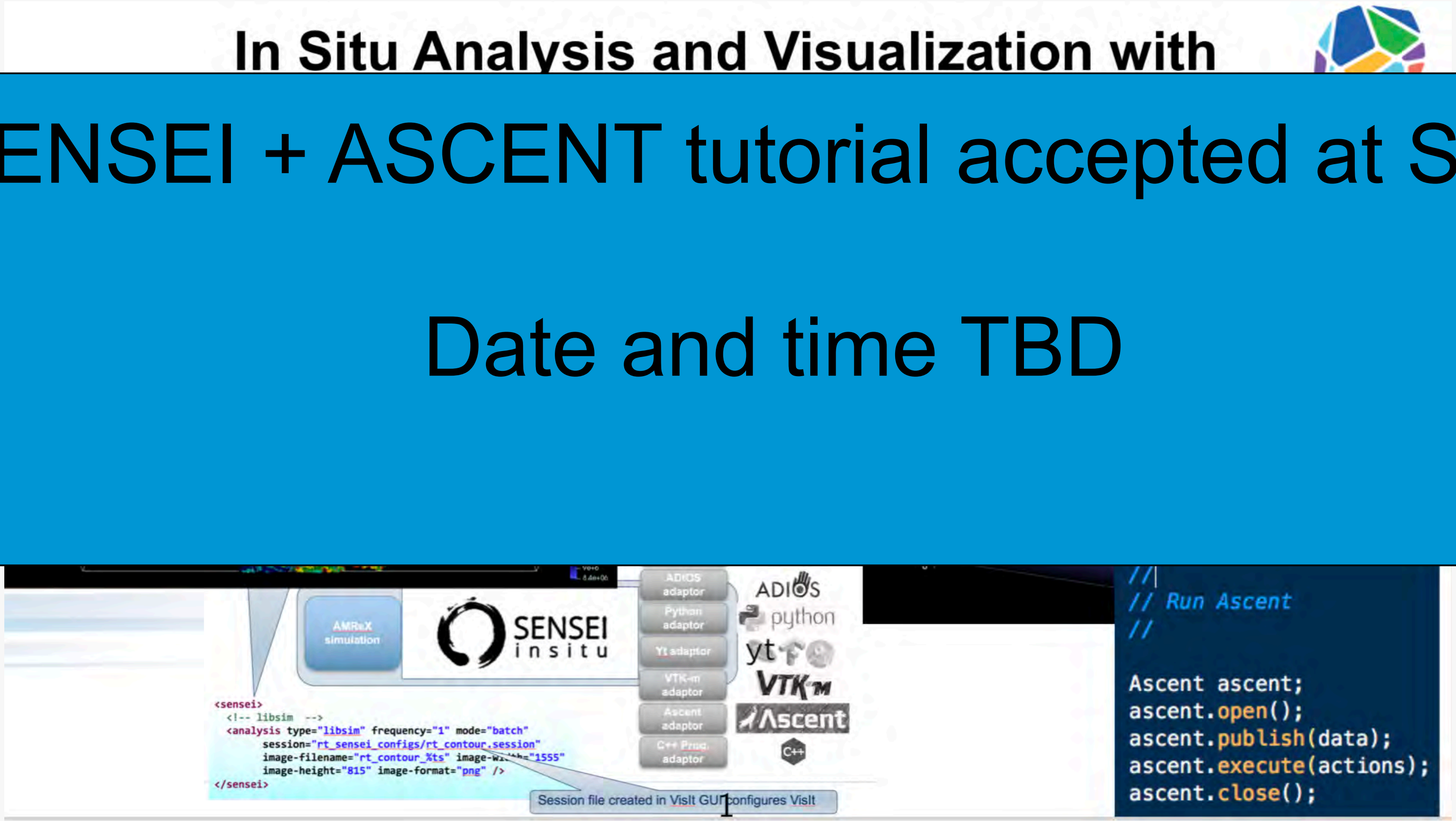
Slides and Virtual Machine available here:
https://sensei-insitu.org/tutorials/sc19.html
https://ix.cs.uoregon.edu/~hank/sc20/



Argonne NATIONAL LABORATORY

SENSEI + ASCENT tutorial at SC19 and SC20

Slides and Virtual Machine available here:
https://sensei-insitu.org/tutorials/sc19.html
https://ix.cs.uoregon.edu/~hank/sc20/

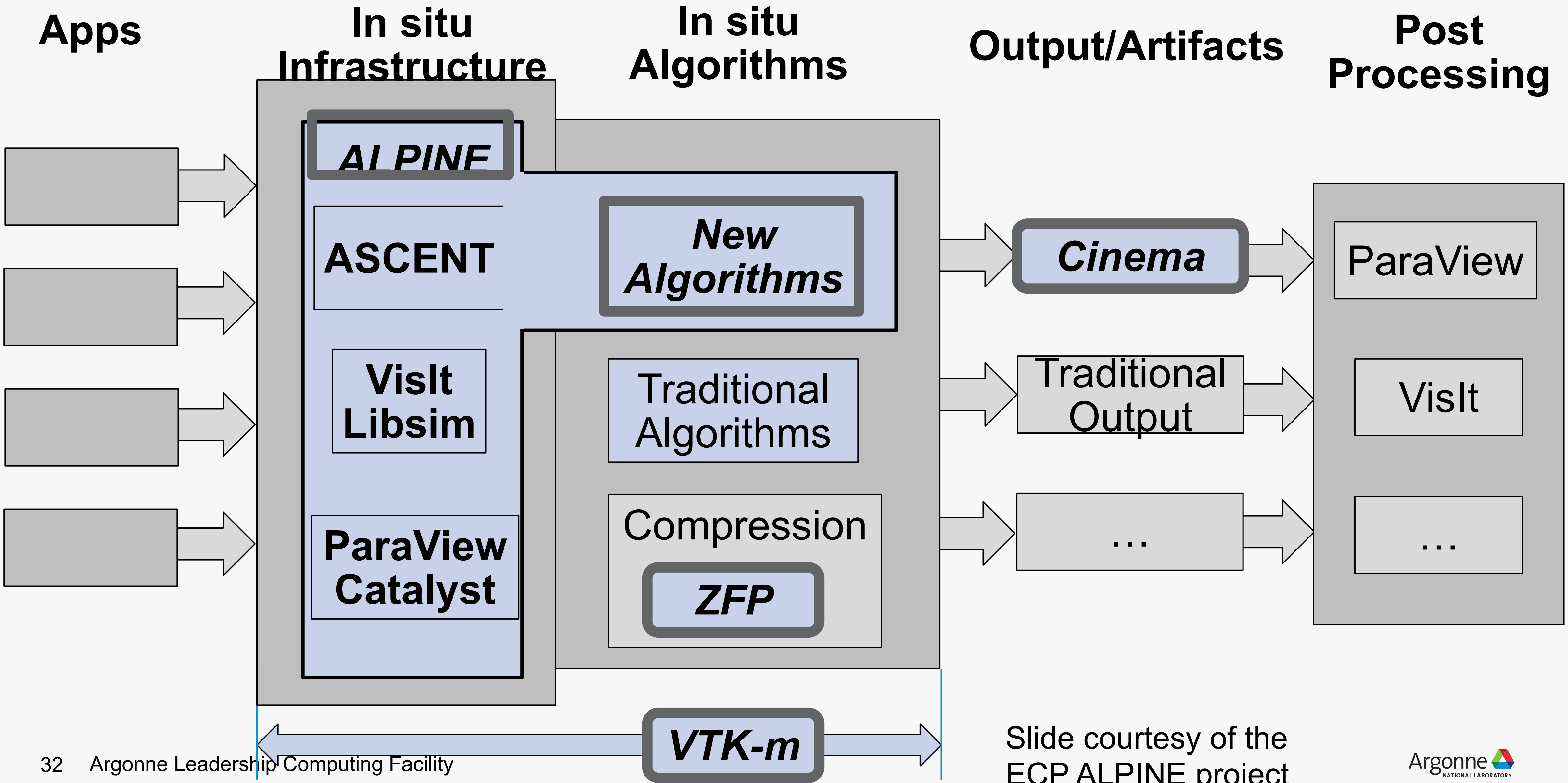


SENSEI + ASCENT tutorial accepted at SC21

Date and time TBD

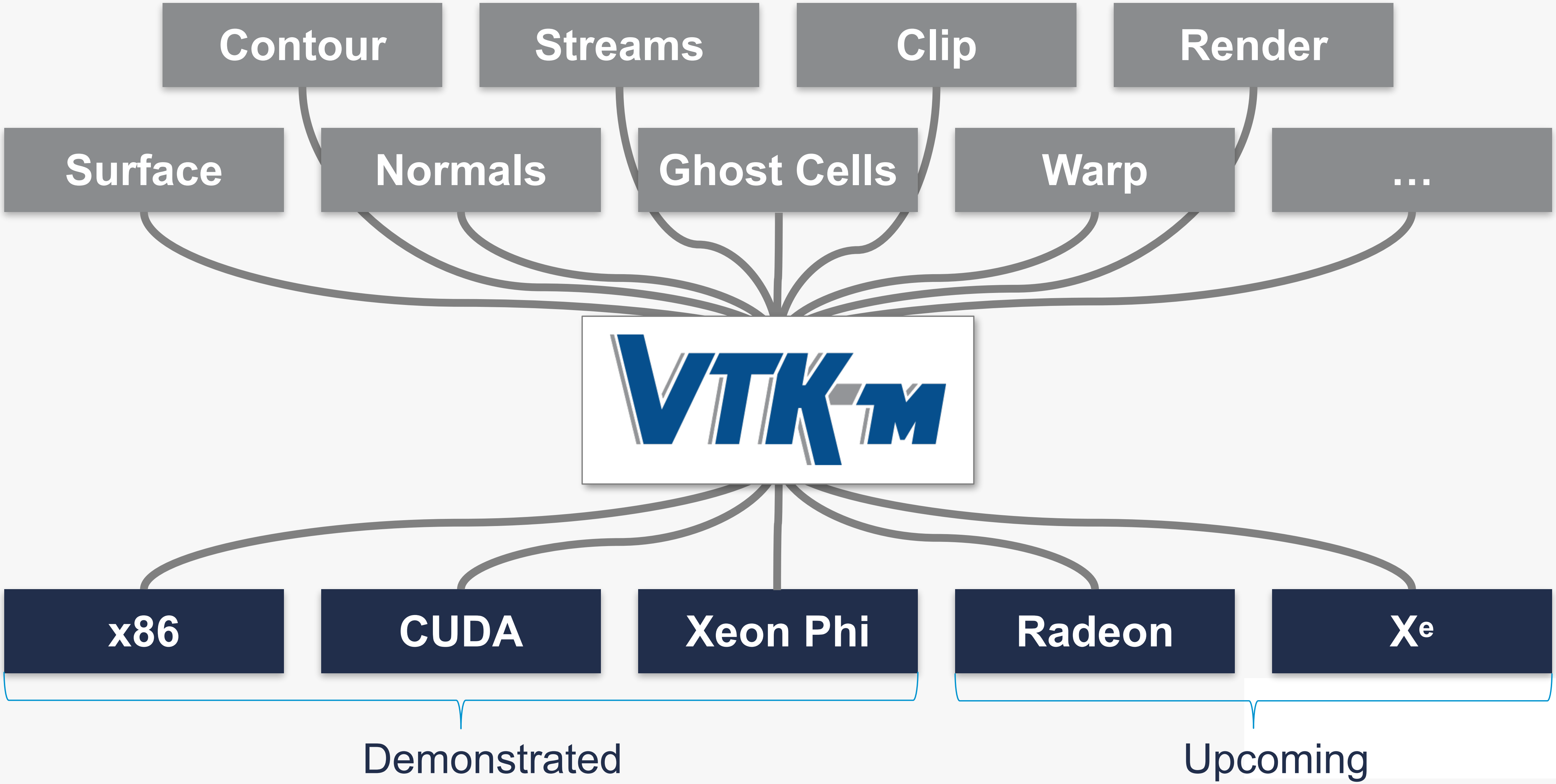
Argonne NATIONAL LABORATORY

Exascale Computing Project
Software Technology Data and Visualization



Slide courtesy of the ECP ALPINE project

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

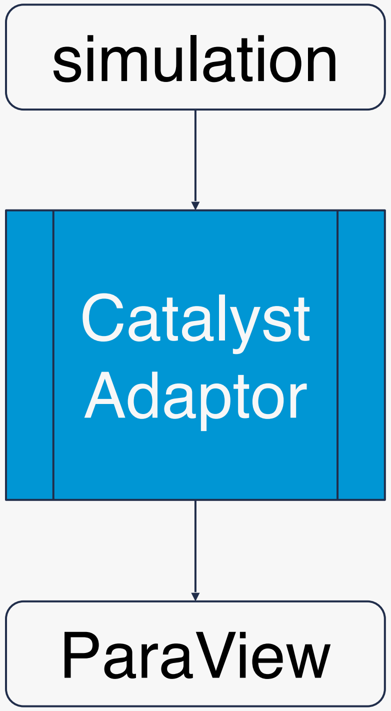


Slide courtesy of the ECP VTK-m project

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



Development challenges:

- Requires good understanding of VTK data model and APIs

Build/development challenges:

- Requires a CMake-based build system
- Requires ParaView SDK (cannot use distributed ParaView binaries)
- Simulation build tightly coupled with ParaView version used

Maintenance challenges:

- Changing APIs and data model
- Changing build system

Extracts from slide set courtesy Utkarsh Ayachit, Kitware Inc.

Catalyst Revised: the design

Simplifying the adaptor
----> switch to Conduit

- Avoid need to understand VTK data model
- Provide mechanism to provide data with zero-copy & meta-data to interpret it

Simplifying build and deployment

- Inspired by MPICH ABI compatibility initiative
- Simulations to link against a tiny stub and allow switching of implementation at runtime

Utkarsh Ayachit, Andrew Bauer, Ben Boeckel, Berk Geveci, Ken Moreland, Patrick O'Leary, and Tom Osika: *Catalyst Revised: Rethinking the ParaView In Situ Analysis and Visualization API*, WOIV 2021

QUESTIONS?

Joe Insley
insley@anl.gov

Silvio Rizzi
srizzi@anl.gov

Janet Knowles
jknowles@anl.gov

Victor Mateevitsi
vmateevitsi@anl.gov