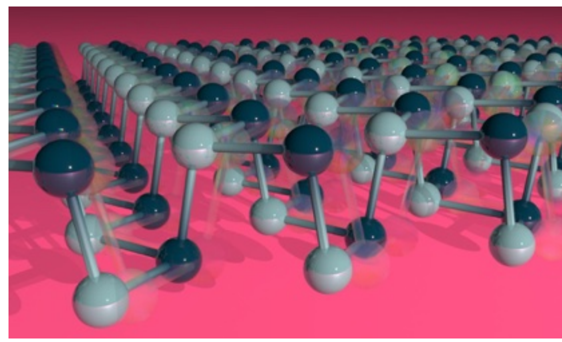
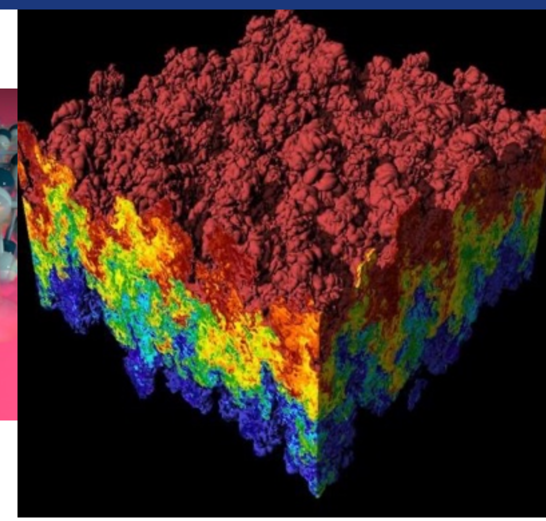
ARGONNE

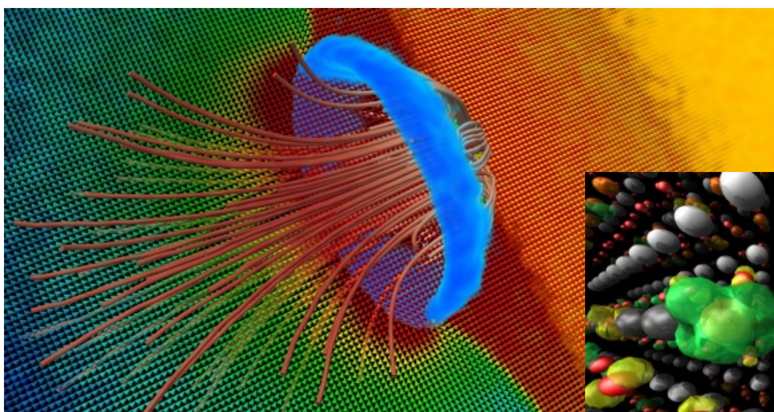
ATPESC2025

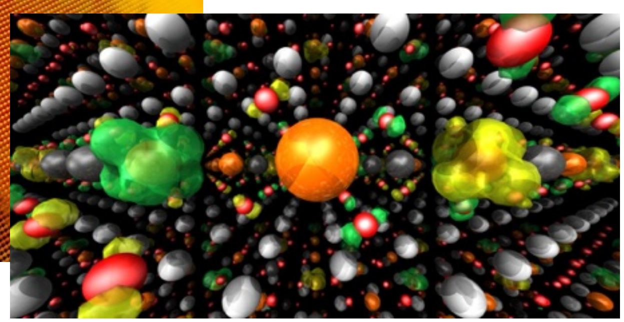
EXTREME-SCALE COMPUTING

Data Analysis and Visualization









extremecomputingtraining.anl.gov

Argonne NATIONAL LABORATORY

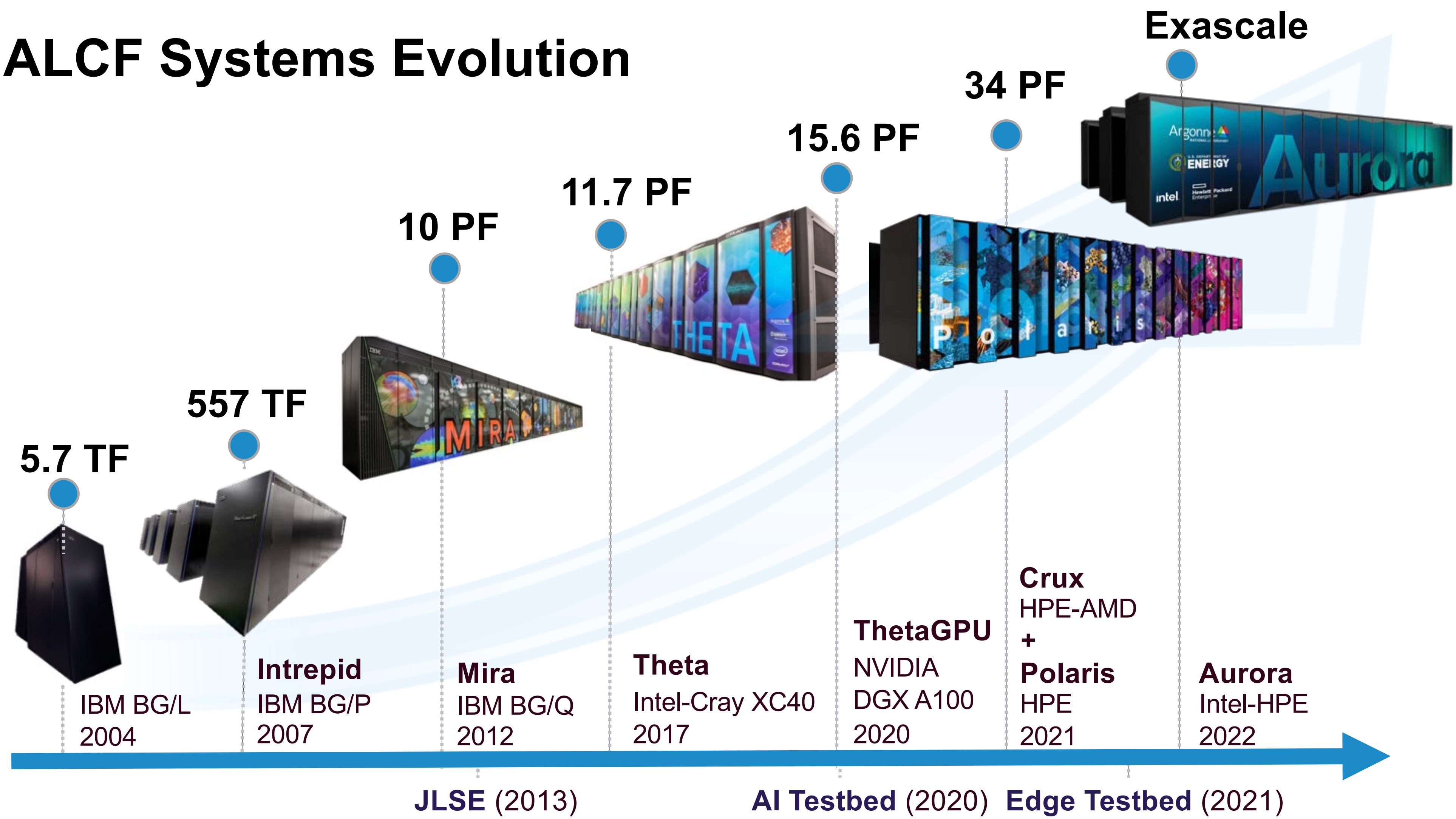
Visualization & Data Analysis

Time	Title of presentation	Lecturer
8:30 am	**Data Analysis and Visualization Introduction**	Joe Insley *ANL*, Silvio Rizzi *ANL*
9:15 am	**Visualization and Analysis of HPC Simulation Data with VisIt**	Cyrus Harrison *LLNL,* Justin Privitera *LLNL*
10:00 am	*Break*	
10:30 am	**Visualization and Analysis of HPC Simulation Data with VisIt (continued)**	Cyrus Harrison *LLNL,* Justin Privitera *LLNL*
11:00 am	**Exploring Visualization with Jupyter Notebooks**	David Koop *NIU*
11:45 am	**Large Scale Visualization with ParaView**	Dan Lipsa *Kitware*
12:30 pm	*Lunch*	
1:30 pm	**Large Scale Visualization with ParaView**	Dan Lipsa *Kitware*
2:00 pm	**Uncertainty Visualization**	Mengjiao Han *ANL*
2:45 pm	**Ascent**	Cyrus Harrison *LLNL,* Justin Privitera *LLNL*
3:30 pm	*Break*	
4:00 pm	**Trame**	Patrick Avery *Kitware*
4:45 pm	**AI/ML Visualization**	Shilpika, ANL
5:30 pm	*Hands-on*	All
6:30 pm	*Dinner*	
7:30 pm	*After-dinner talk*	Jack Dongarra

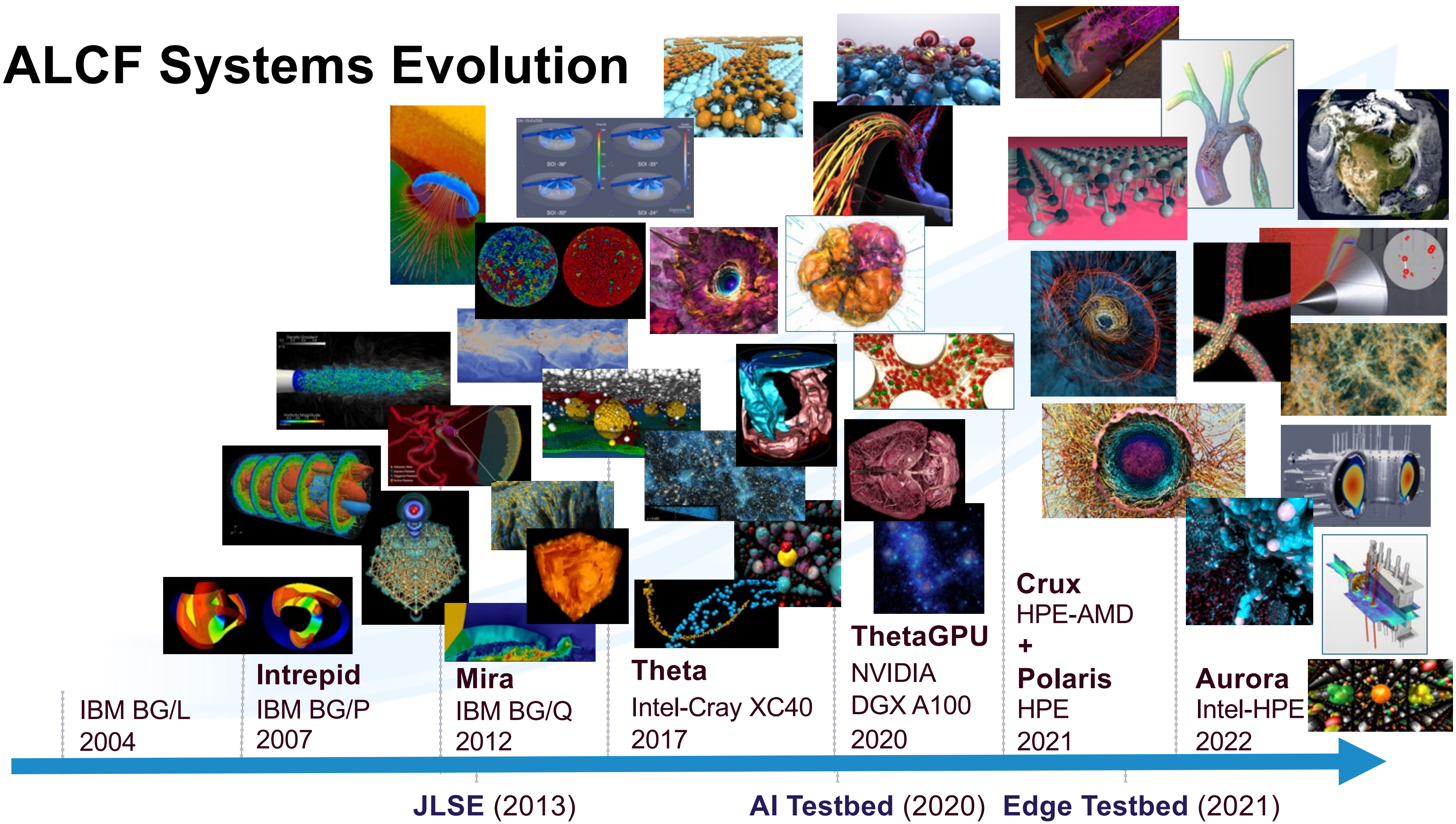
Here's the plan…

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

ALCF Systems Evolution



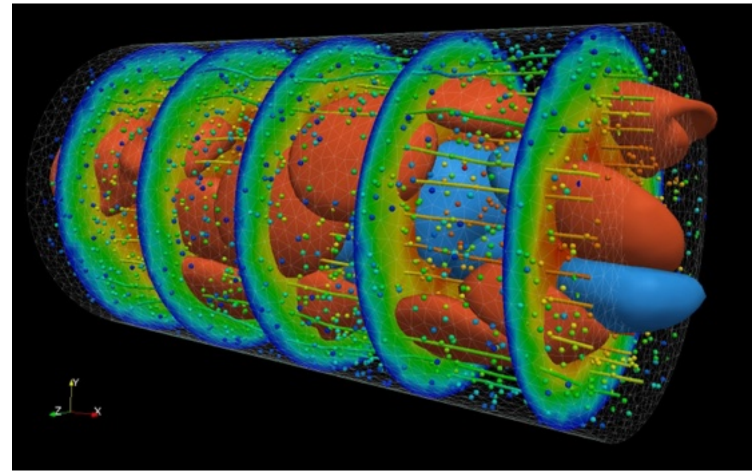
ALCF Systems Evolution



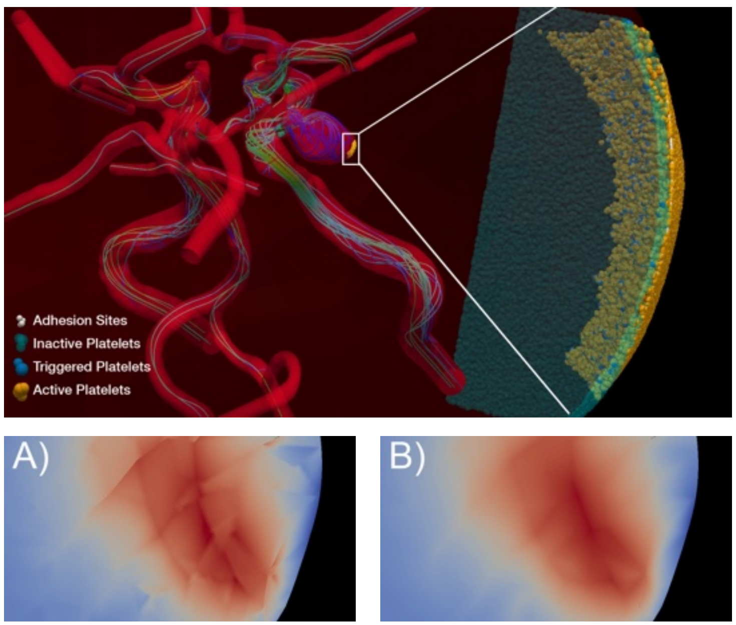
Multi-Scale Simulation / Visualization Arterial Blood Flow

PI: George Karniadakis, Brown University

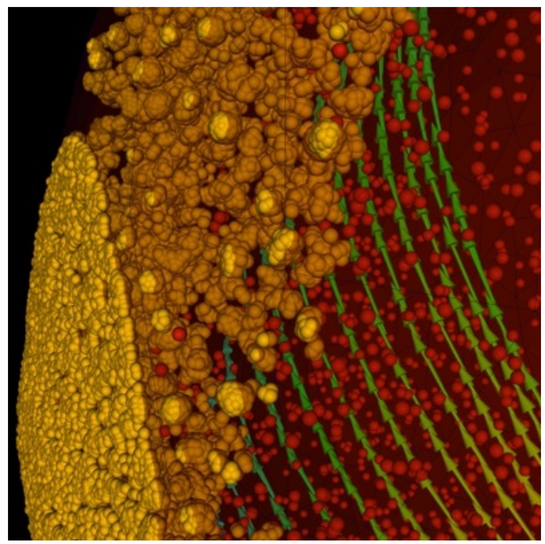
2010



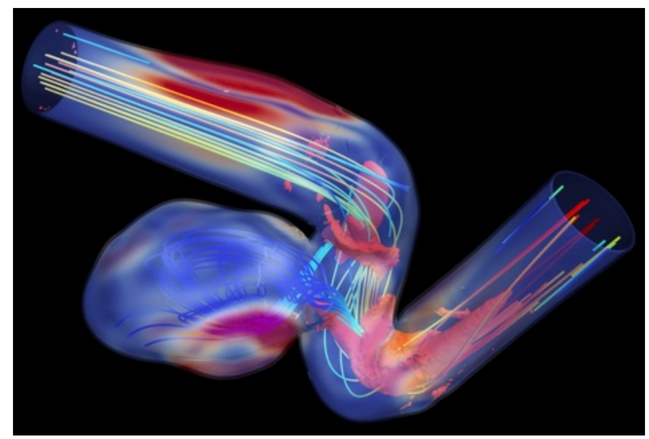
2011



2012

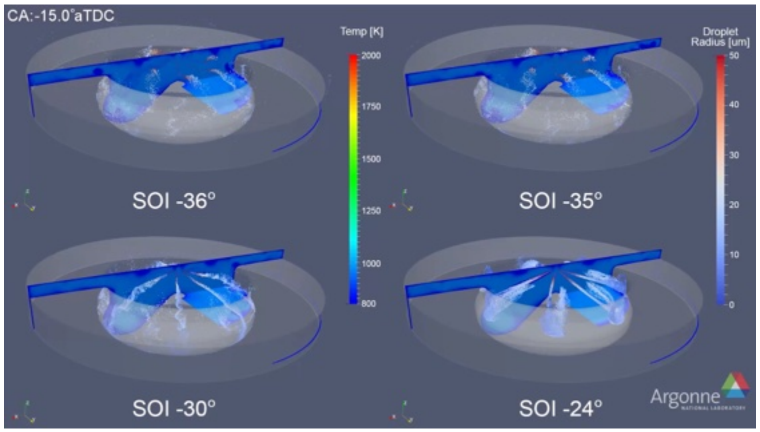


2014

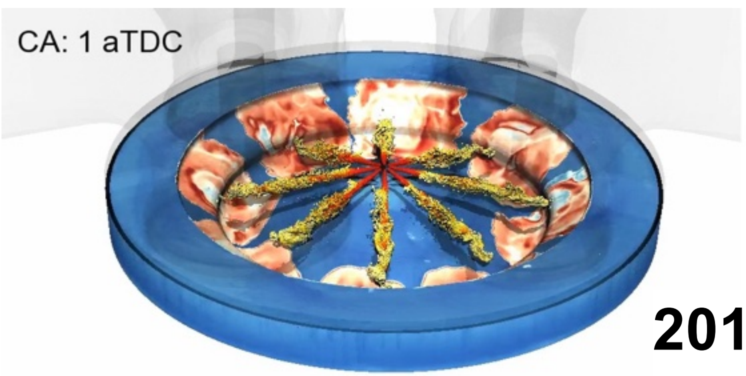


Engineering / Combustion / Biofuels

PI: Sibendu Som, Argonne National Laboratory

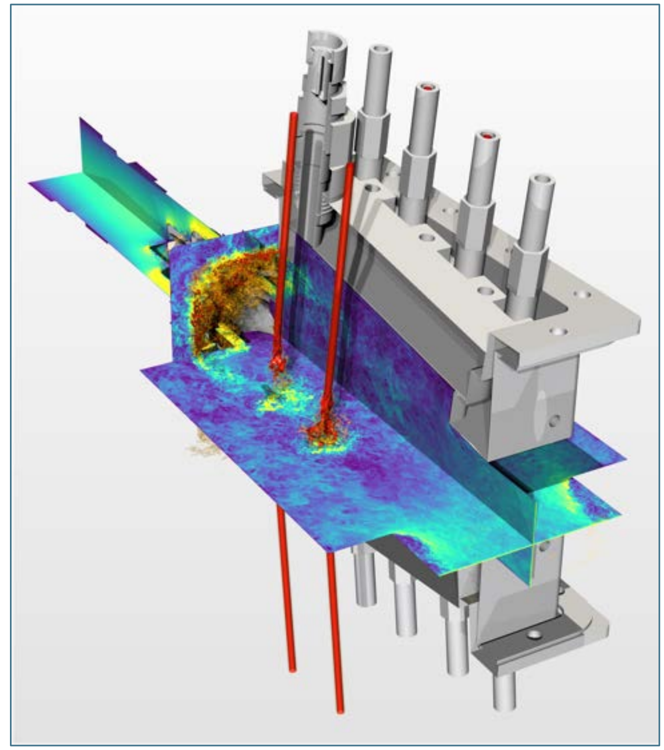


2015



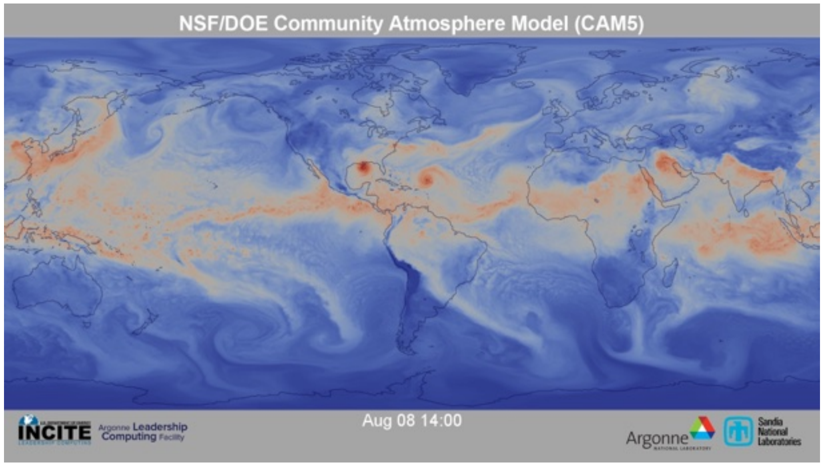
2017

2021



2023

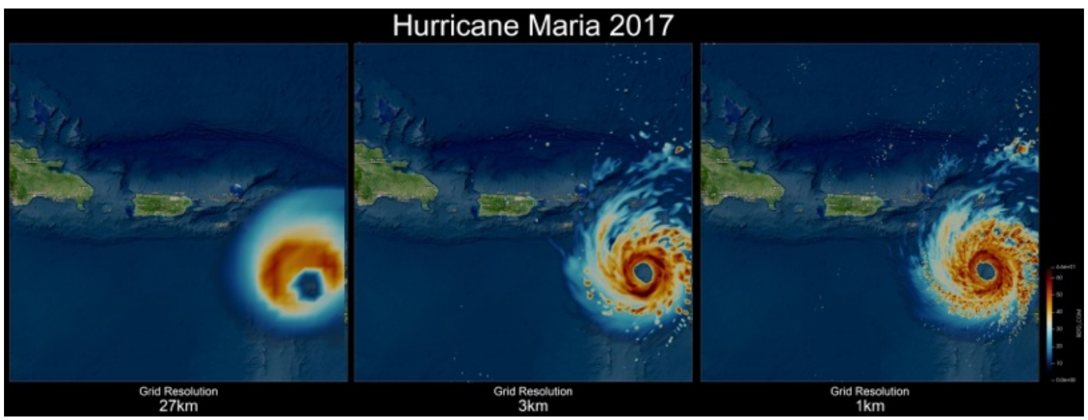
Climate

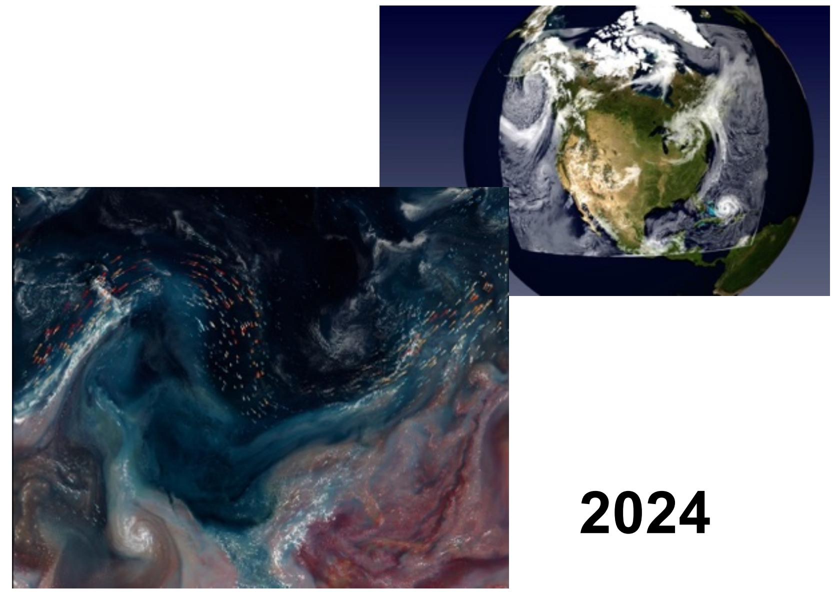


PI: Warren Washington, National Center for Atmospheric Research

2012

2022





2024

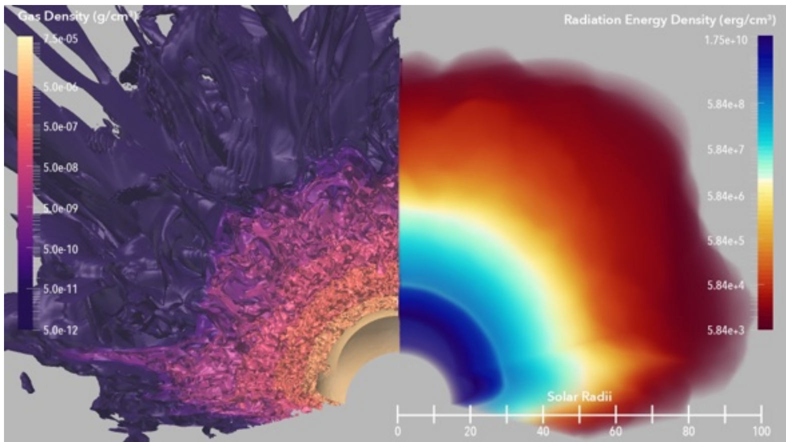
PI: Rao Kotamarthi, Argonne National Laboratory

Argonne NATIONAL LABORATORY

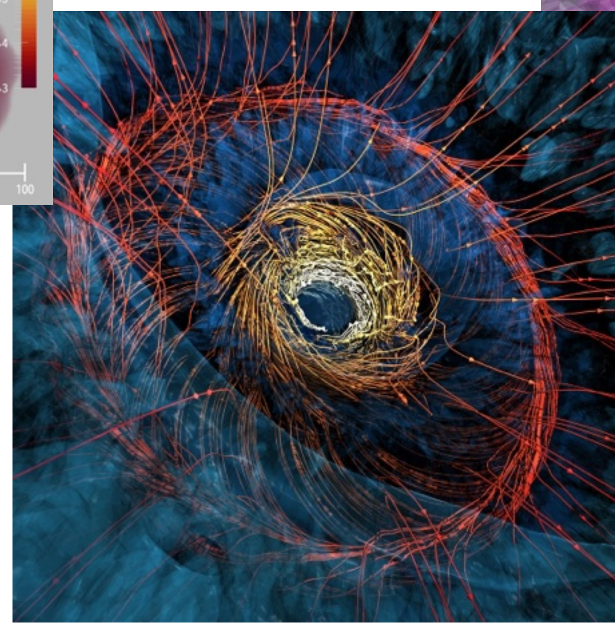
Physics: Stellar Radiation

PI: Lars Bildsten, University of California, Santa Barbara

2018



2017

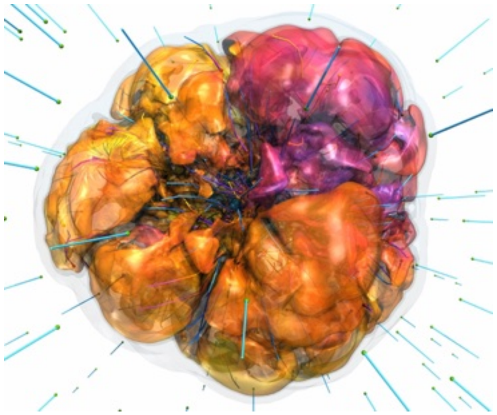


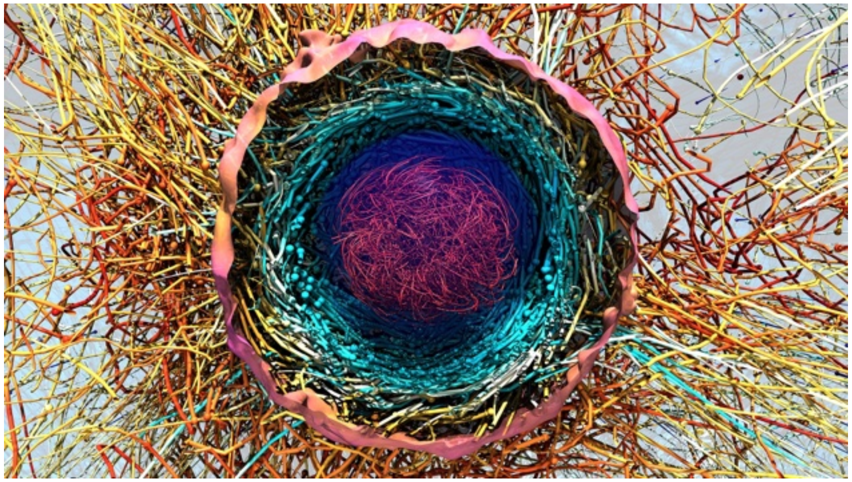
2021

Argonne NATIONAL LABORATORY

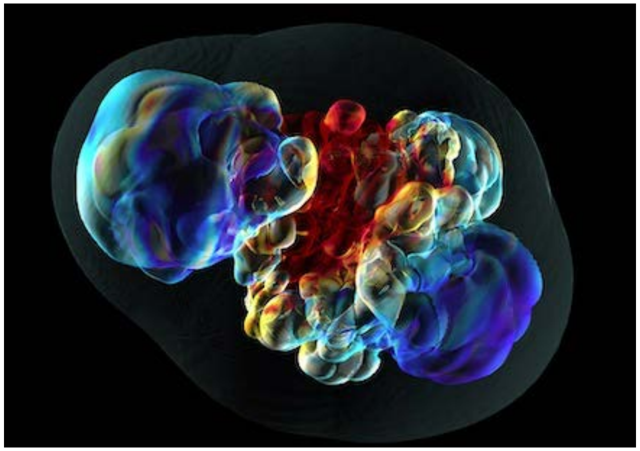
Astrophysics

PI: Adam Burrows, Princeton University

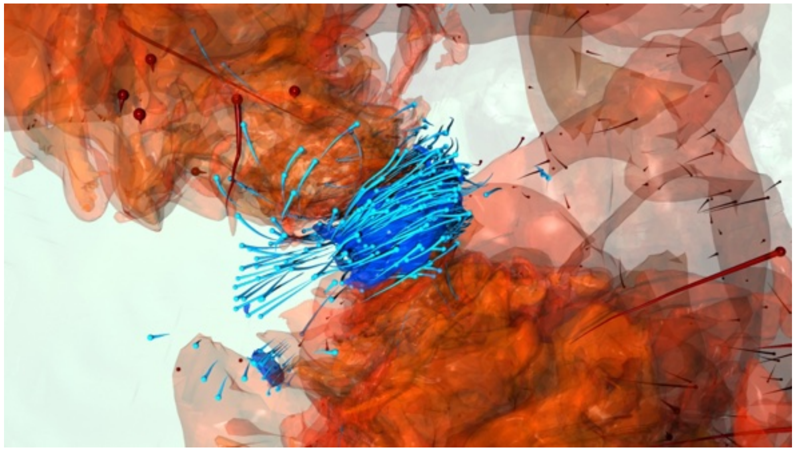

2019


2021

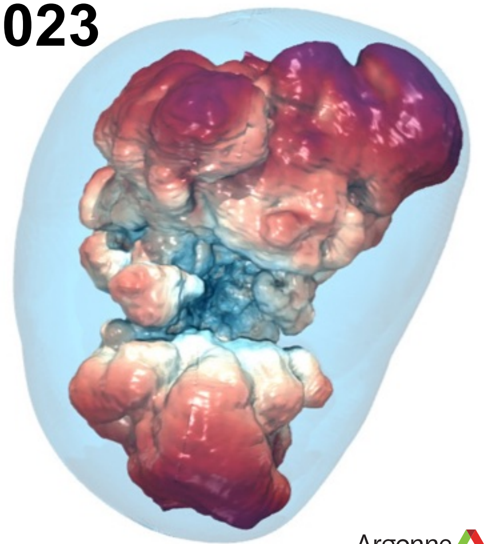
2022



2023



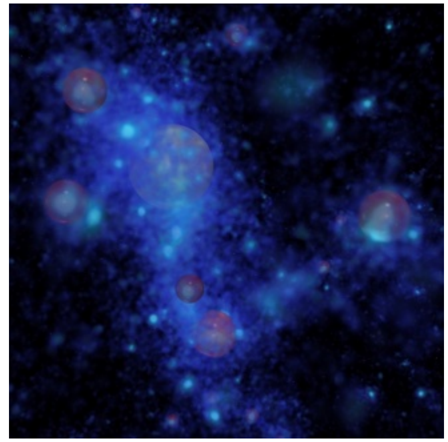
2023



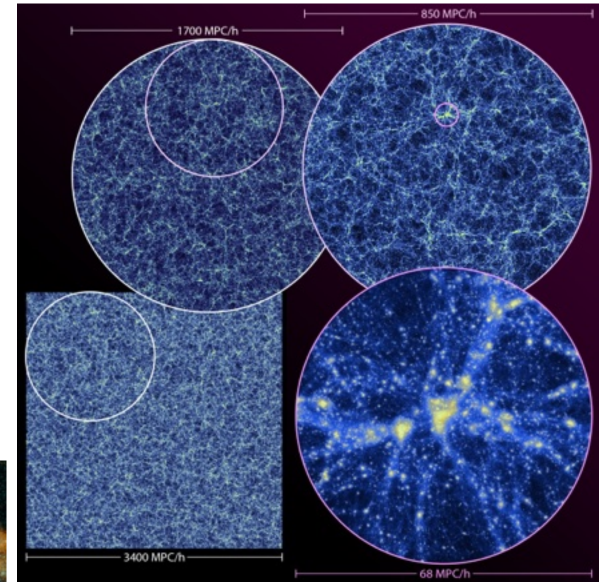
HACC: Cosmology

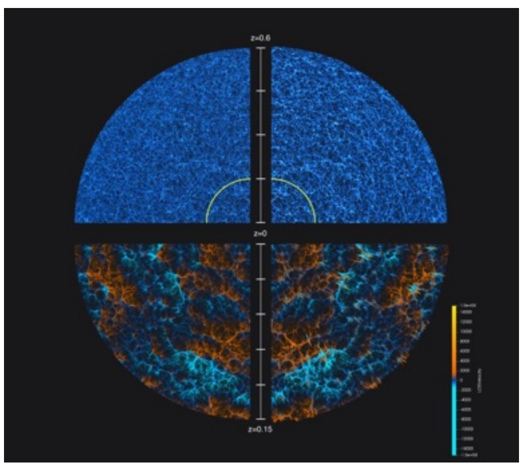
2018

2020

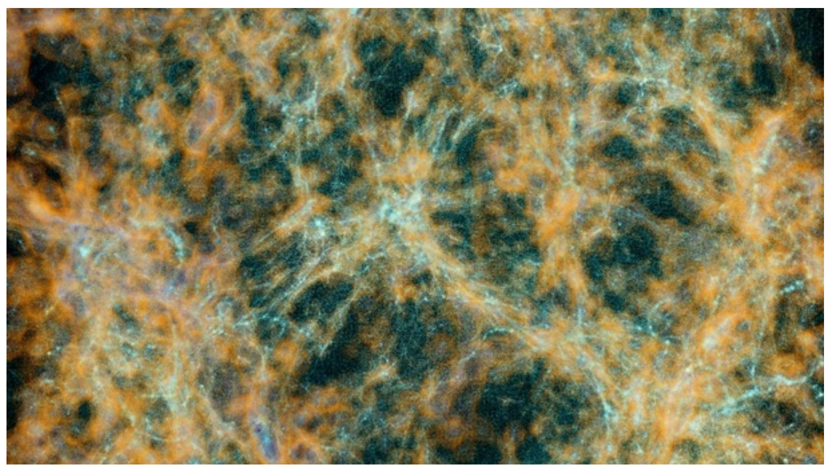


2020





2021

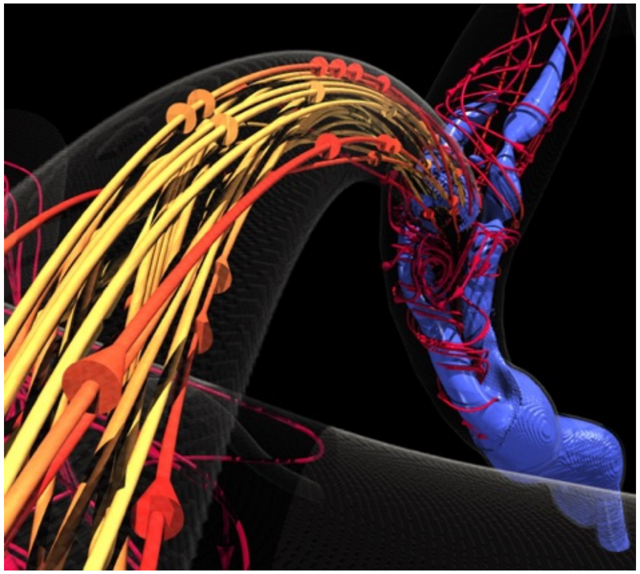


Computed and Rendered on Aurora 2023

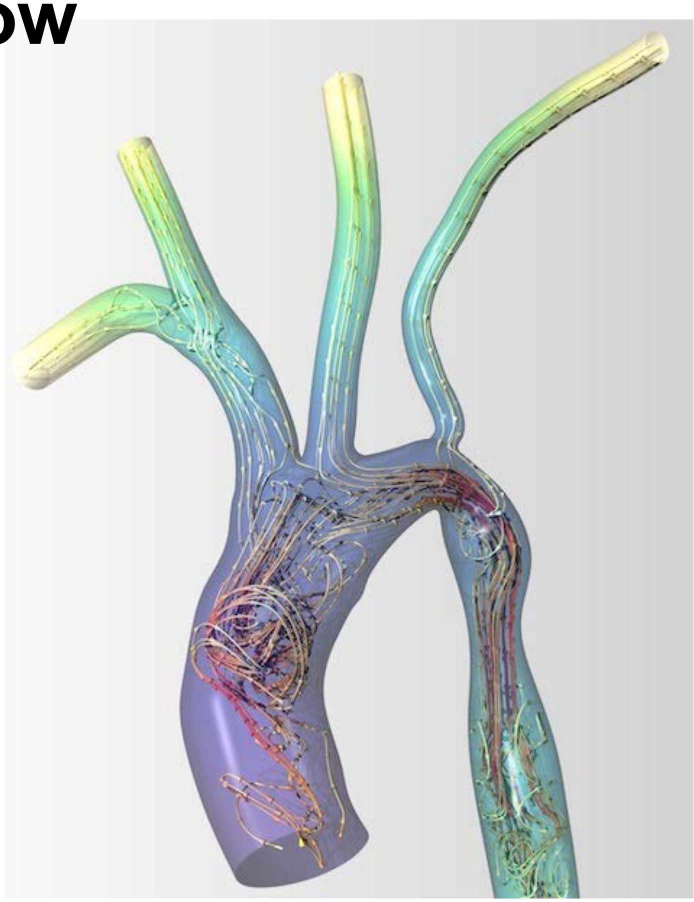
PI: Salman Habib and HACC Team, Argonne National Laboratory

Arterial Blood Flow

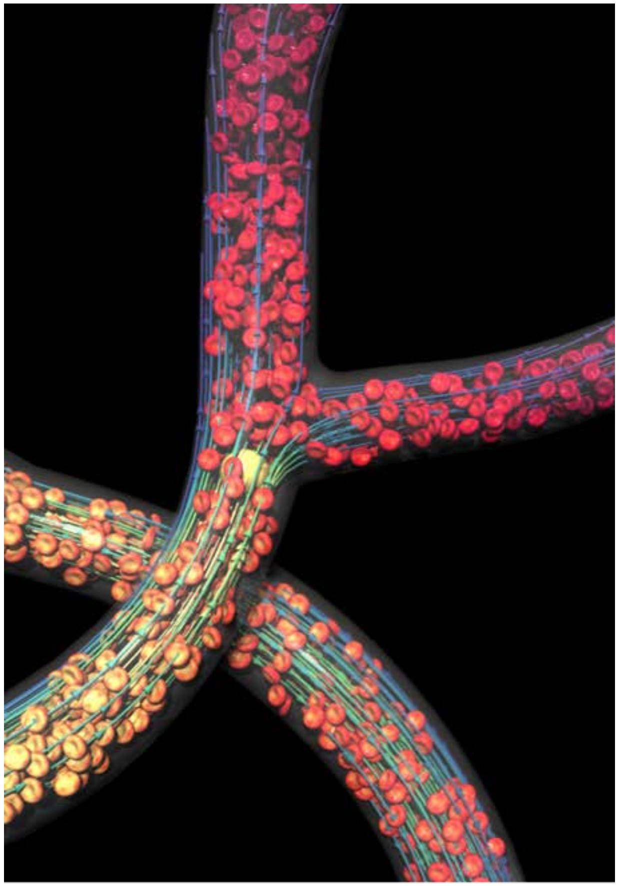
PI: Amanda Randles, Duke University



2020

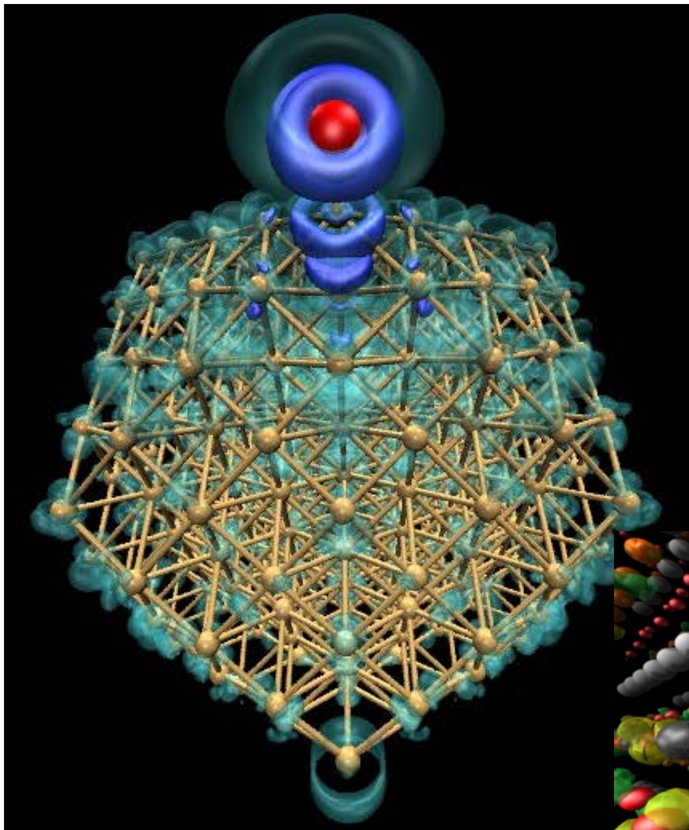


2023 Rendered on Aurora

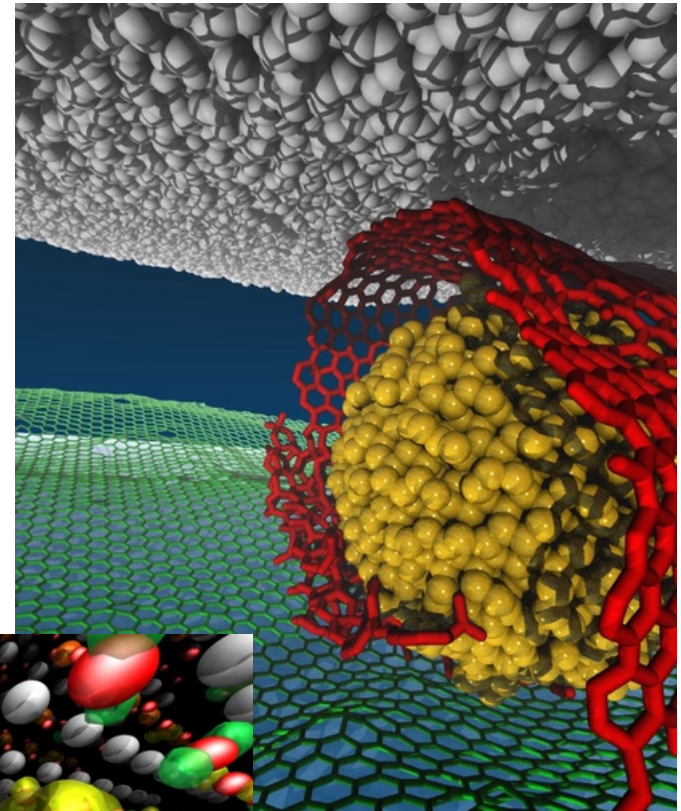


2023 Rendered on Aurora

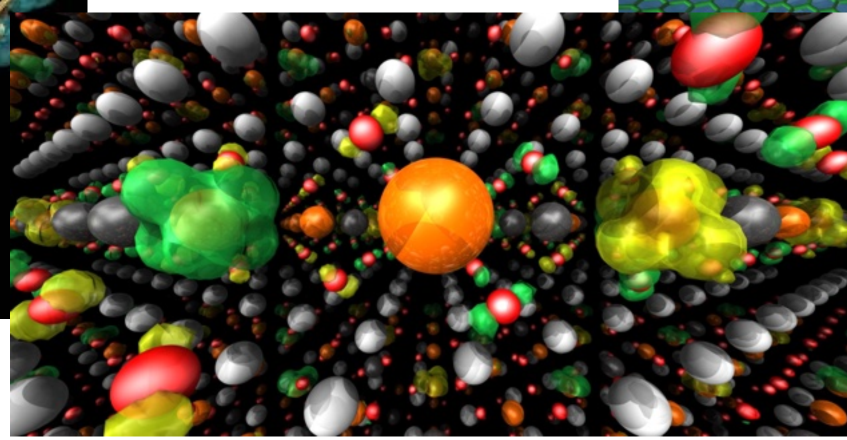
Materials Science / Molecular



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

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

Analysis Environments
– Matlab
– Parallel R

Utilities
– GnuPlot
– ImageMagick
– ffmpeg

Available on Polaris and Aurora

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)

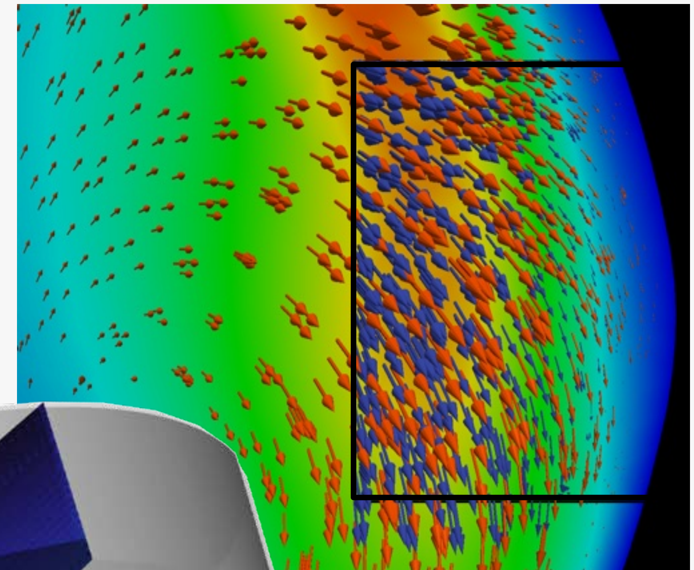
Data File Formats (ParaView & VisIt)

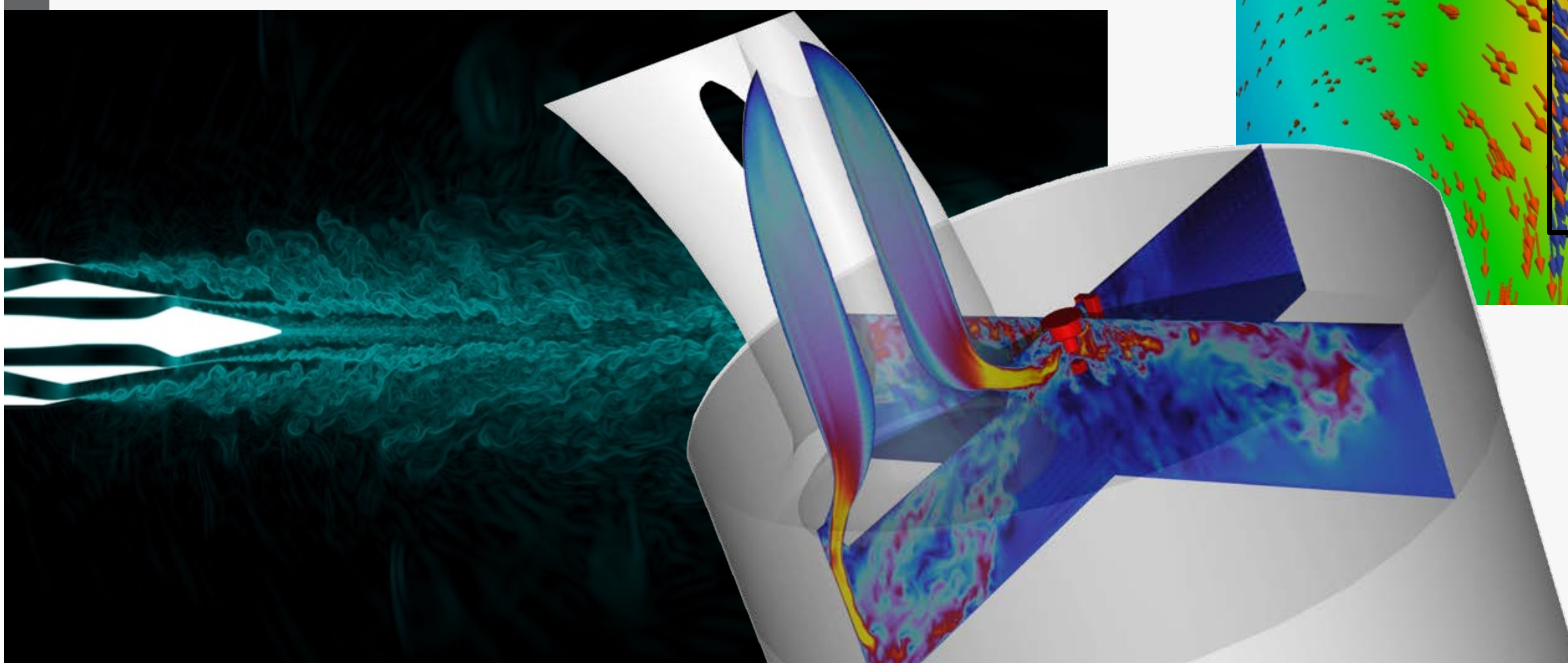
- VTK
- Parallel (partitioned) VTK
- VTK MultiBlock (MultiGroup, Hierarchical, Hierarchical Box)
- Legacy VTK
- Parallel (partitioned) legacy VTK
- EnSight files
- EnSight Master Server
- Exodus
- BYU
- XDMF
- PLOT2D
- PLOT3D
- SpyPlot CTH
- HDF5 raw image data
- DEM
- VRML
- PLY
- Polygonal Protein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

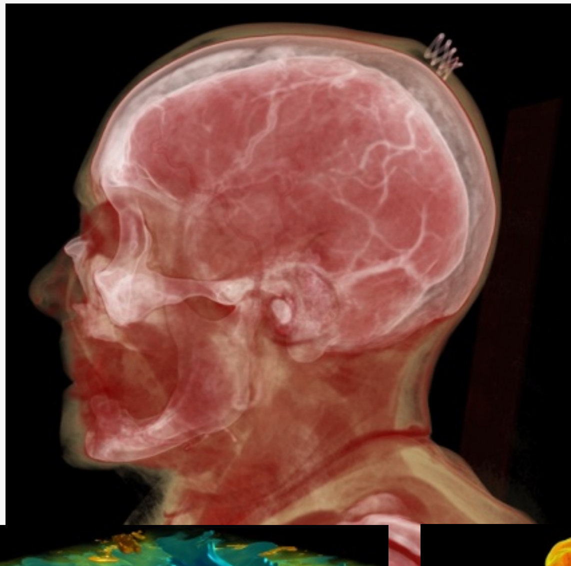
Slice a plane through the data
– Can apply additional visualization methods to resulting plane
VisIt & ParaView & vtk good at this
VMD has similar capabilities for some data formats

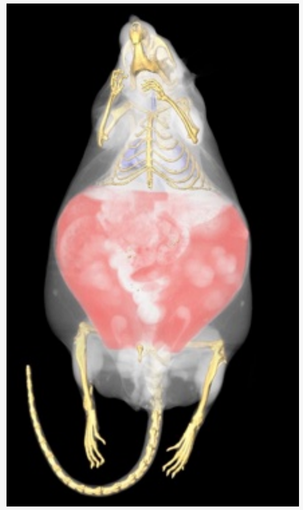


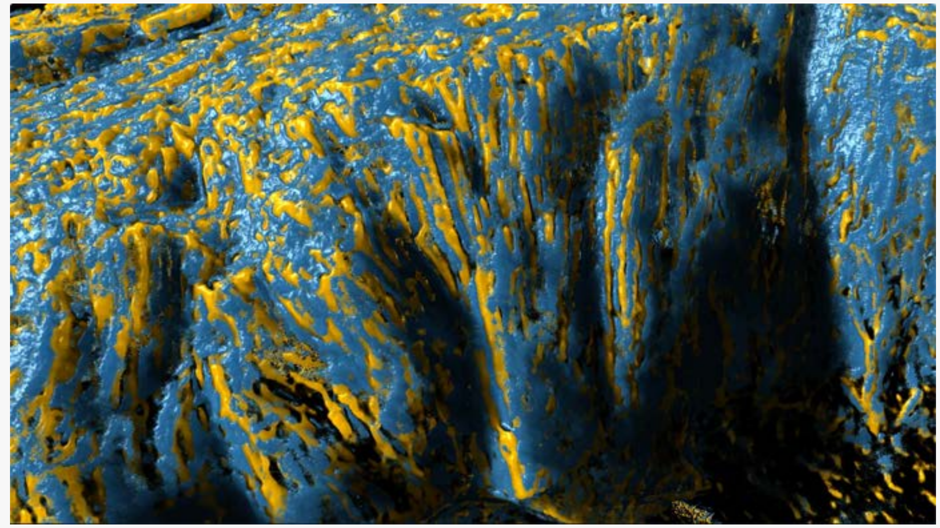


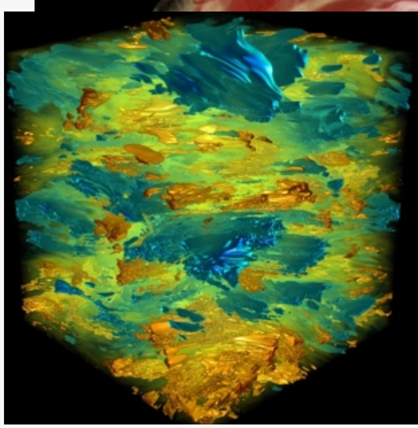
Argonne NATIONAL LABORATORY

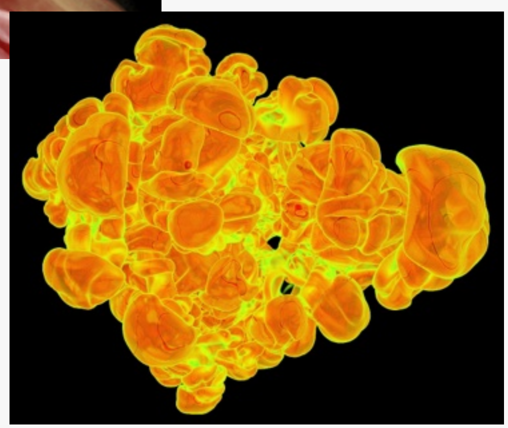
Data Representations: Volume Rendering

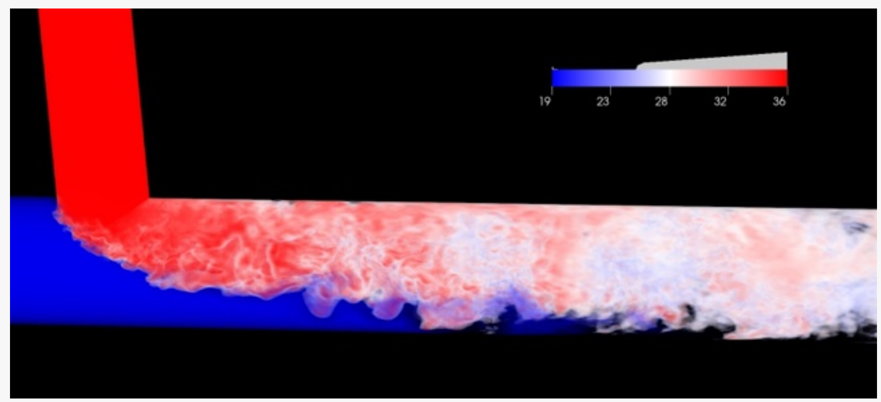












Data Representations: Contours (Isosurfaces)

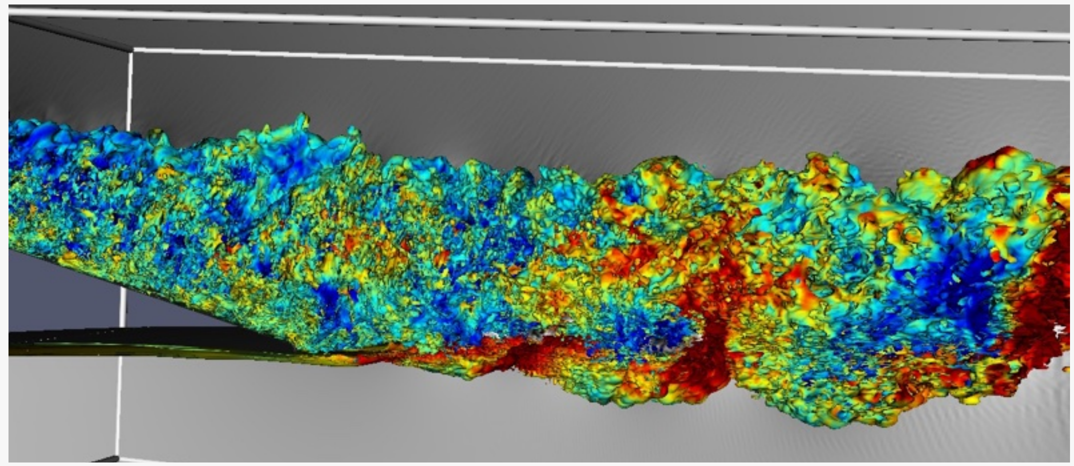
A Line (2D) or Surface (3D), representing a constant value

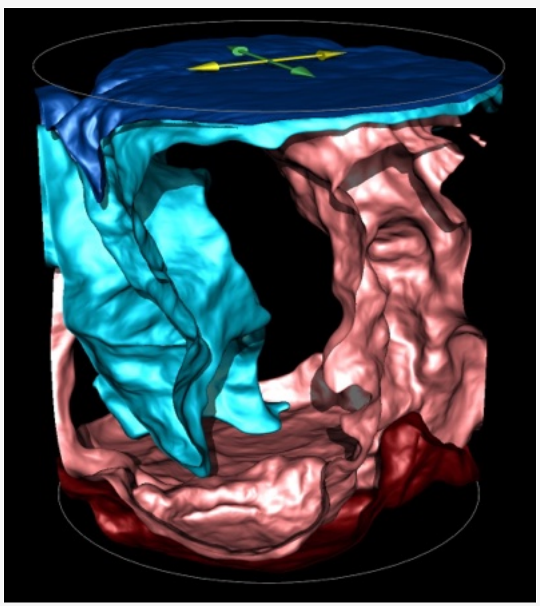
VisIt & ParaView:

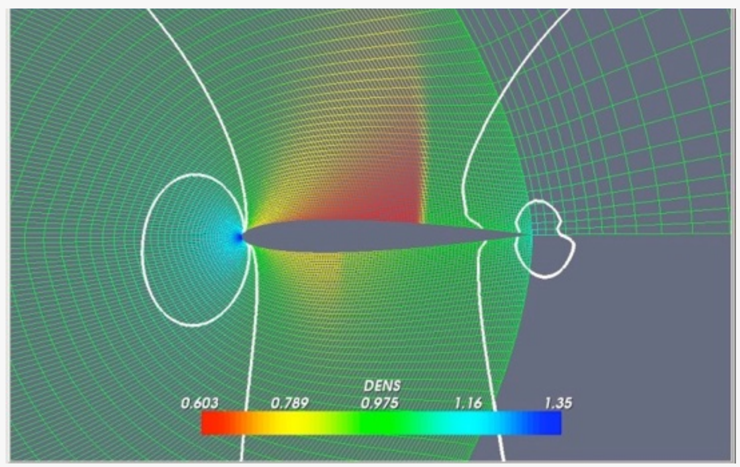
– good at this

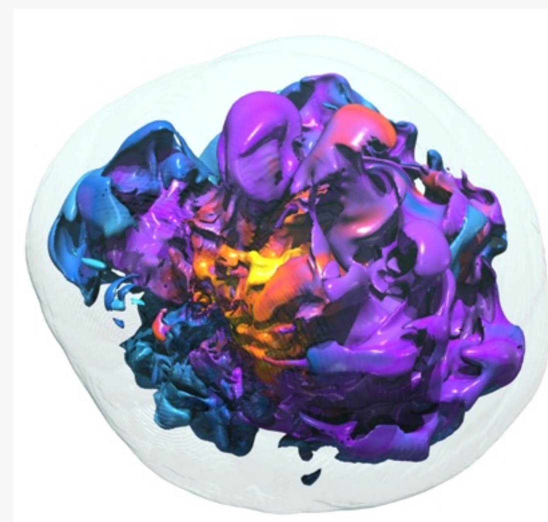
vtk:

– same, but again requires more effort



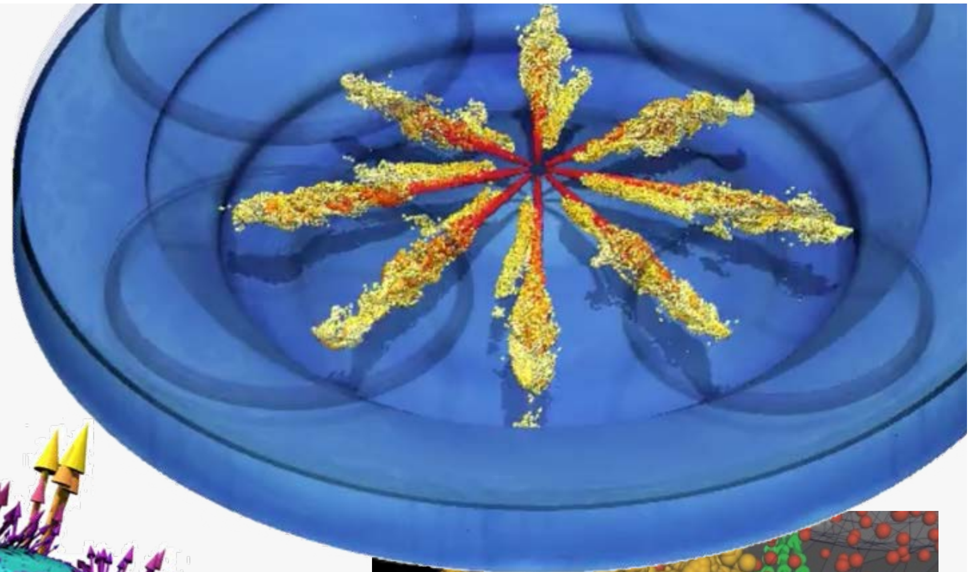


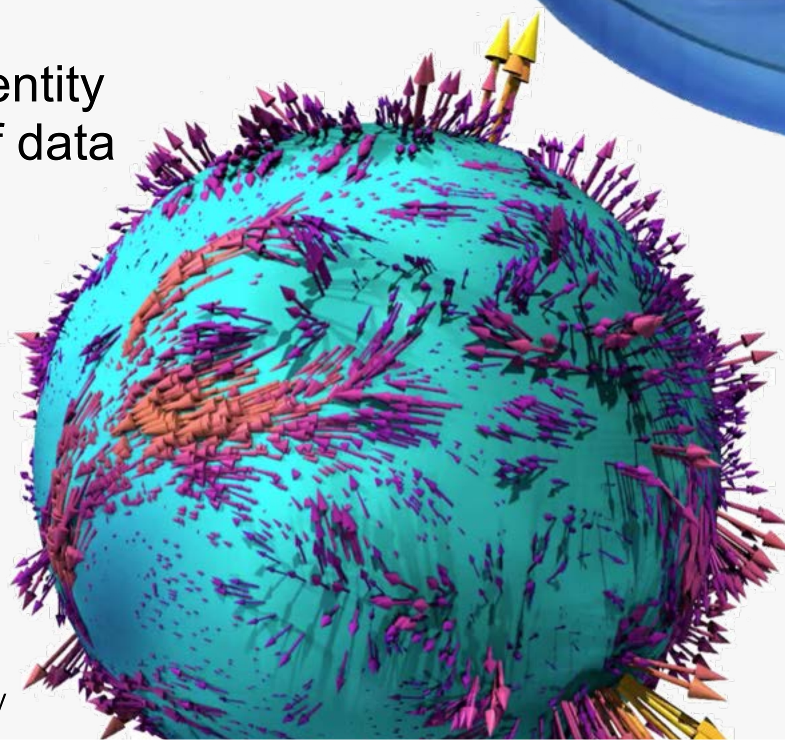


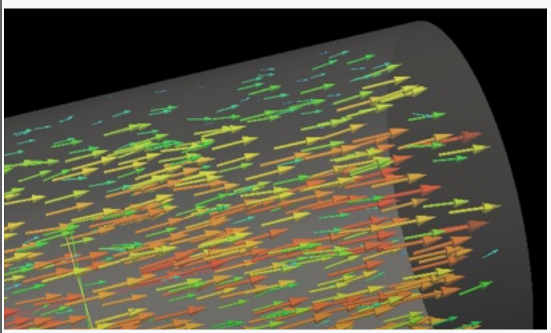


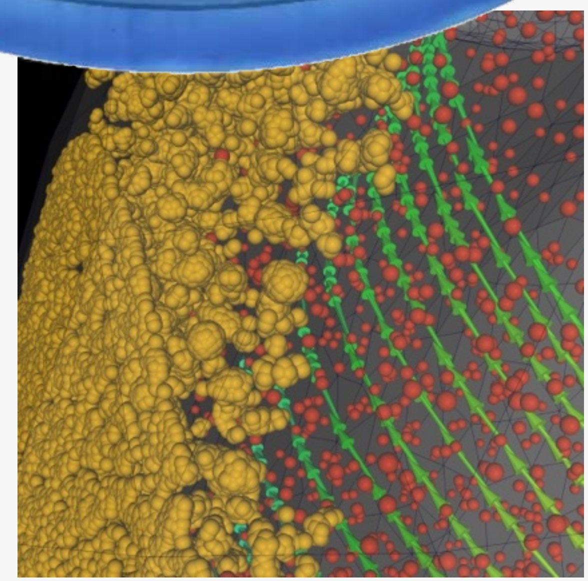
Data Representations: Glyphs

- 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







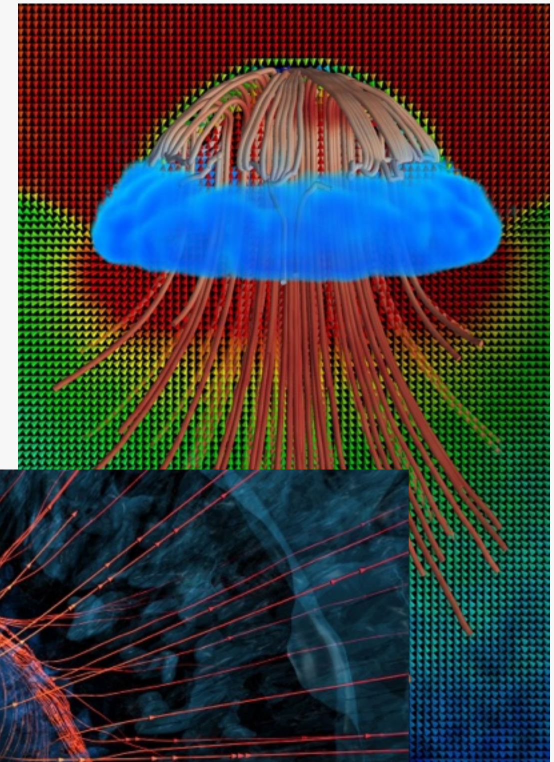


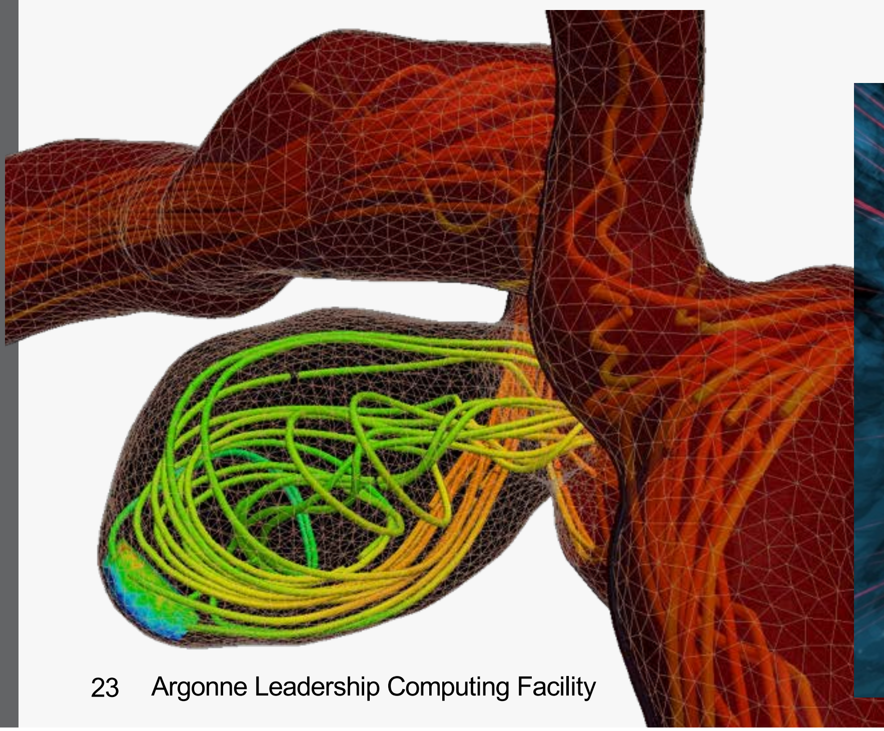
Data Representations: Streamlines

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/Ovito:

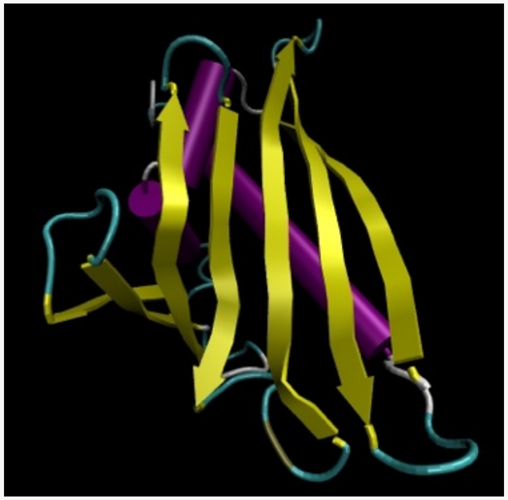
– Lots of domain-specific representations
– Many different file formats
– Animation
– Scriptable

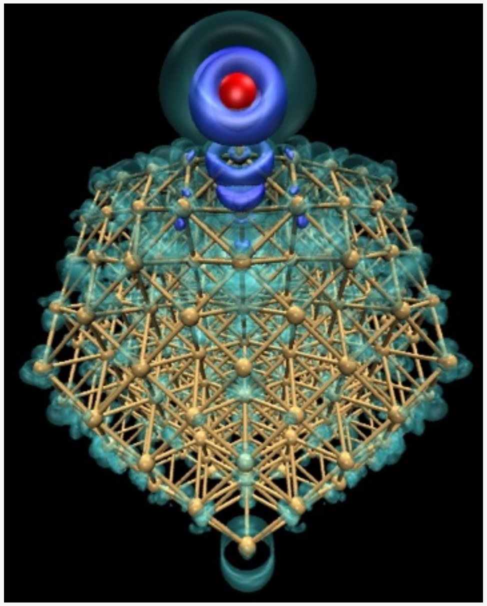
VisIt & ParaView:

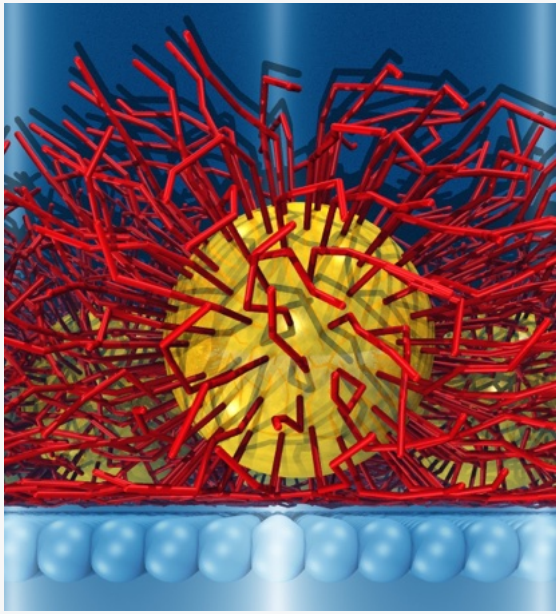
– Limited support for these types of representations, but improving

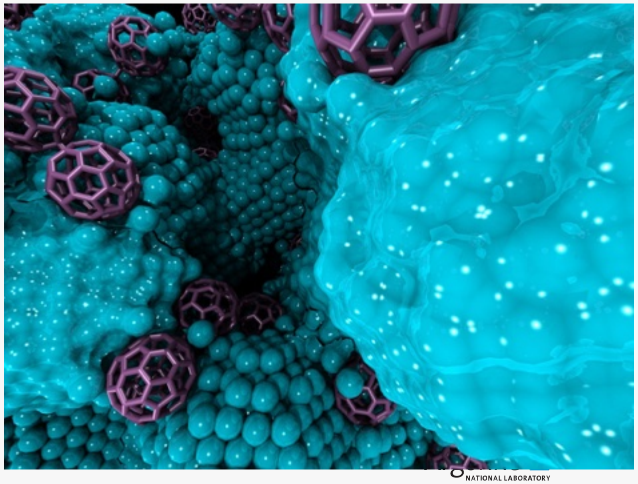
VTK:

– Anything's possible if you try hard enough



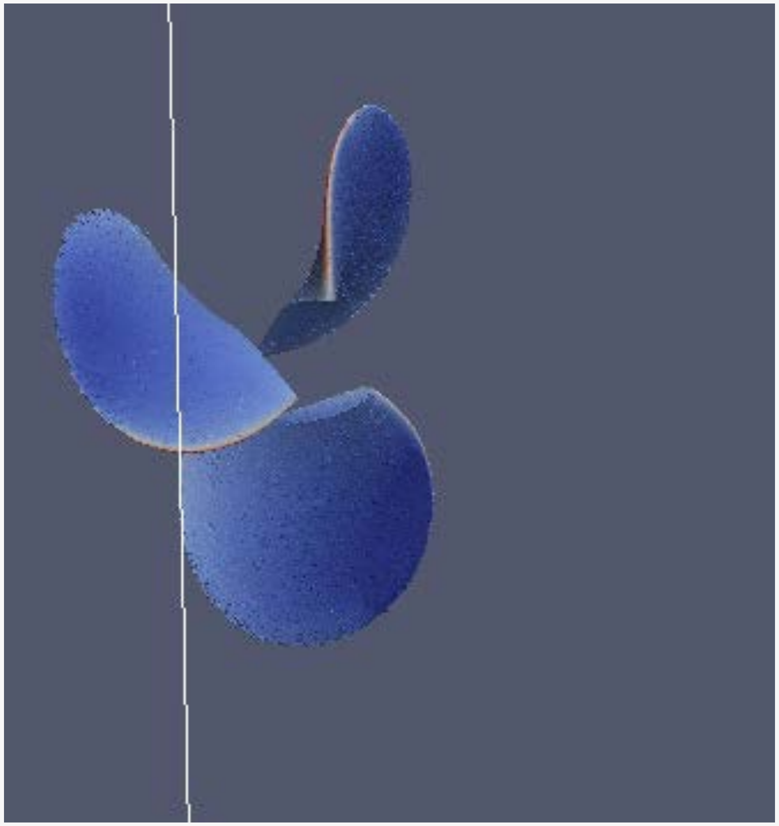




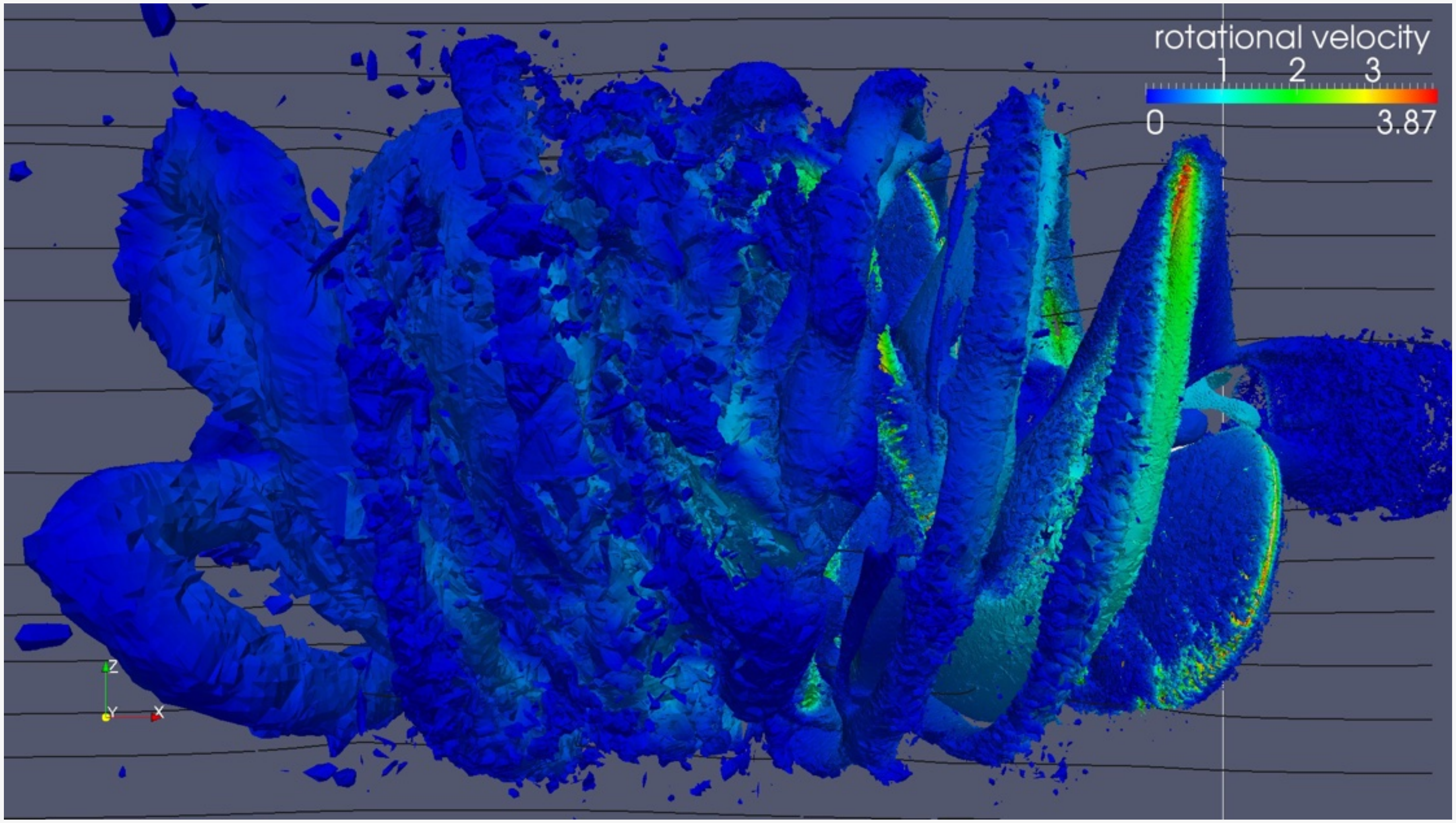


Visualization for Debugging

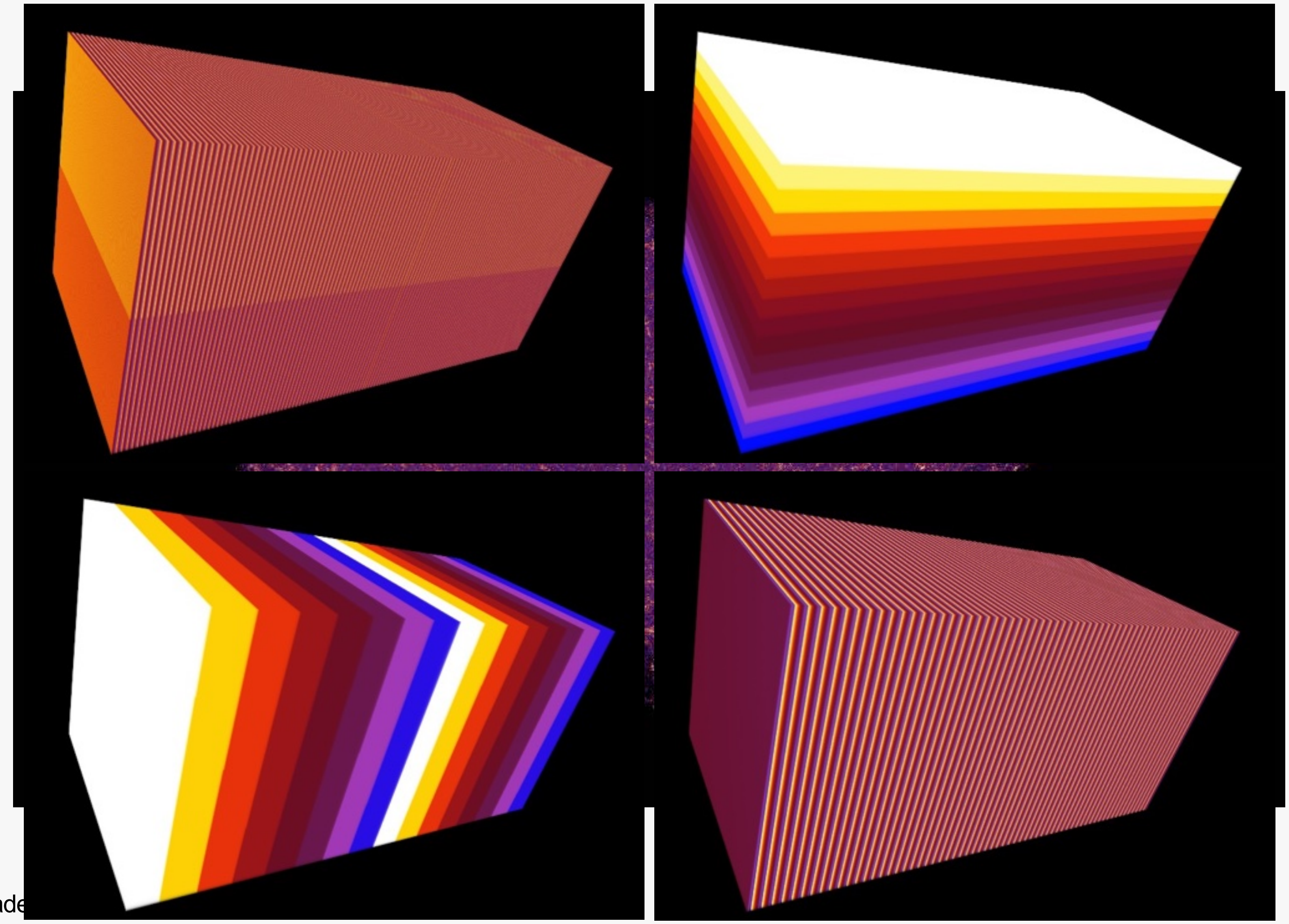
Visualization for Debugging



Visualization for Debugging

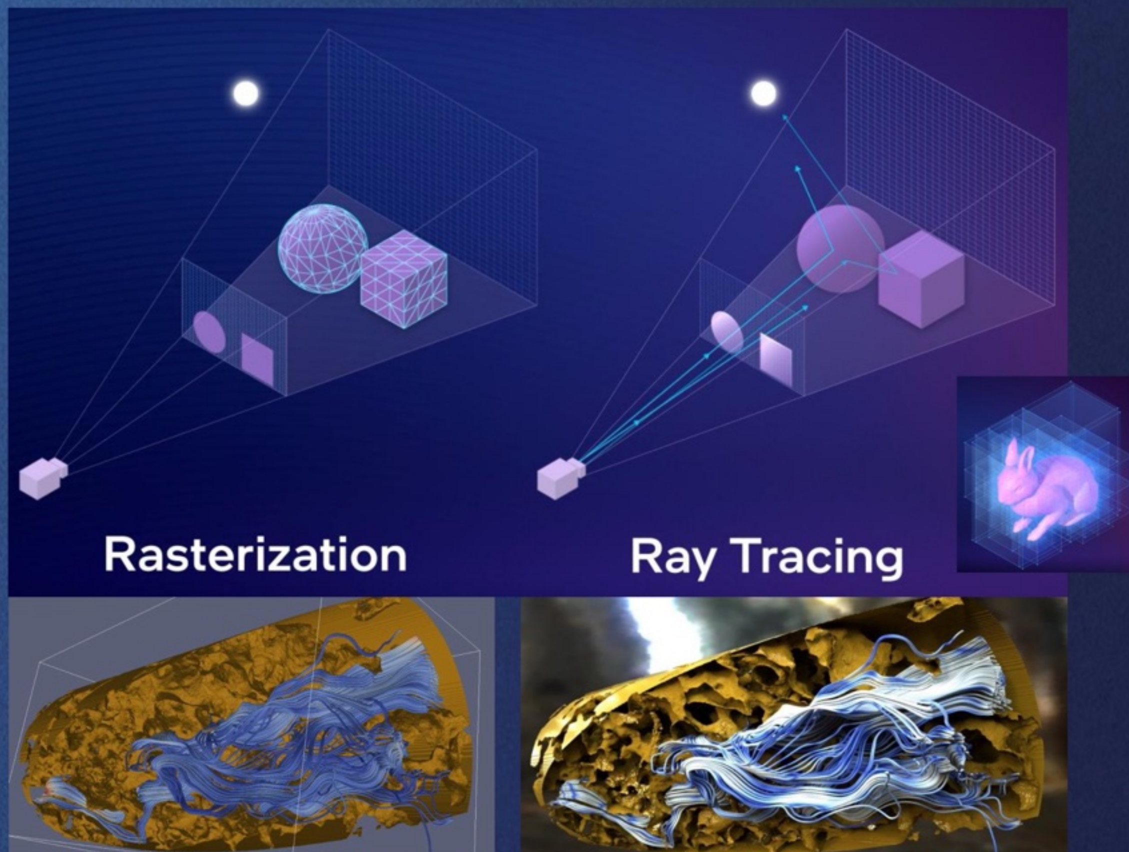


Visualization as Diagnostics: Color by Thread ID



Advanced Rendering

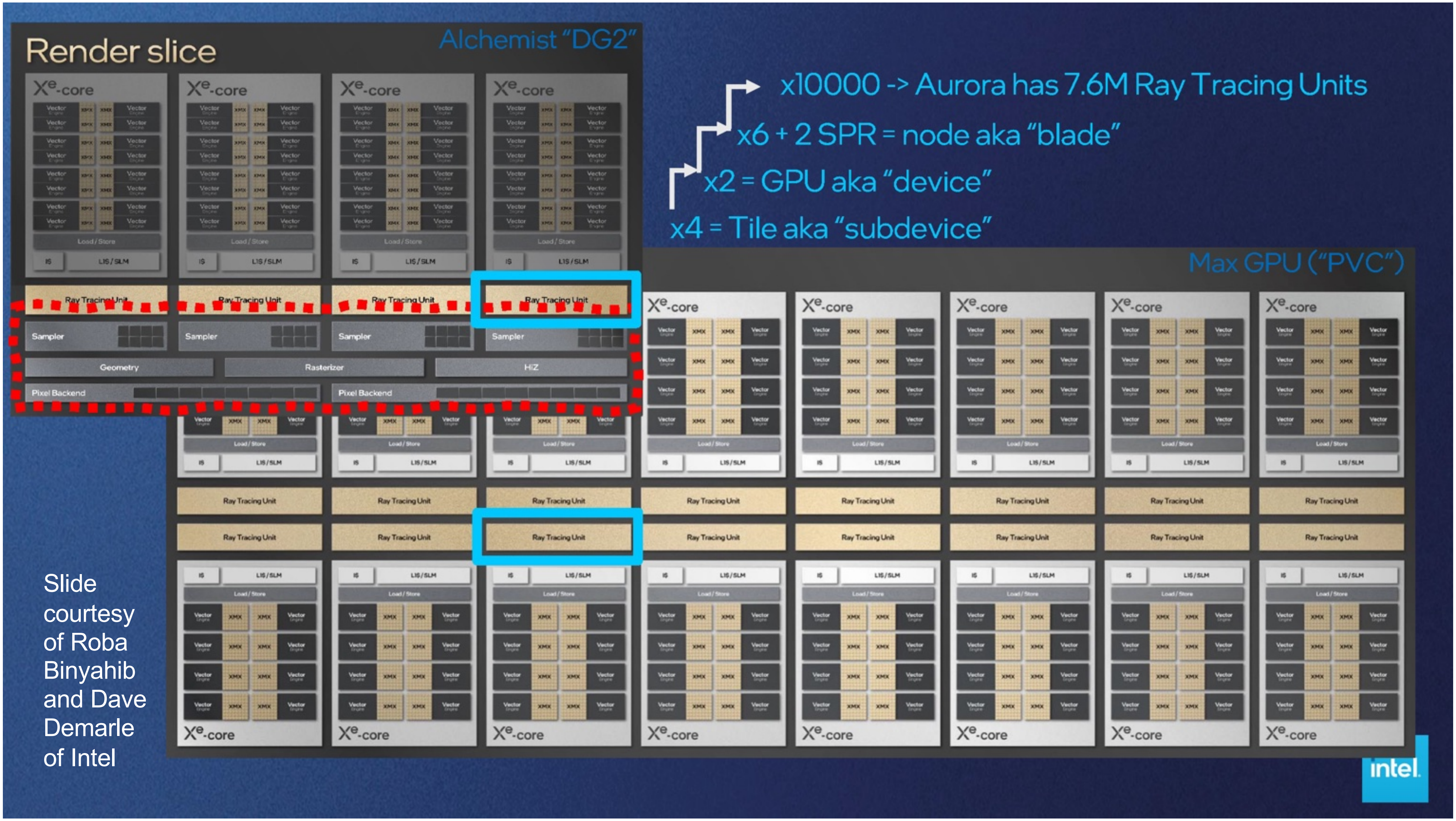
Rendering

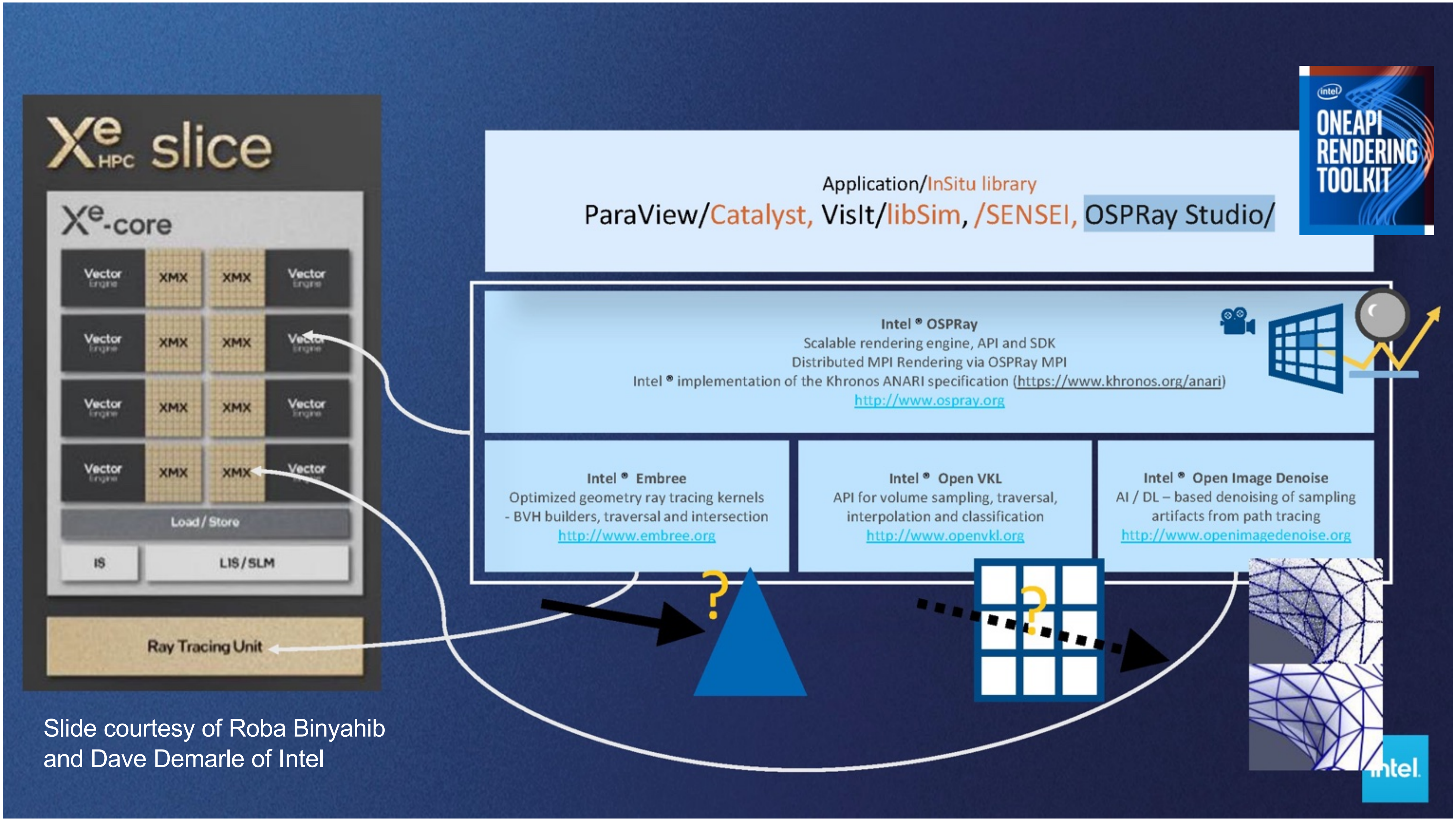


BVH sorts space to accelerate Ray Tracing

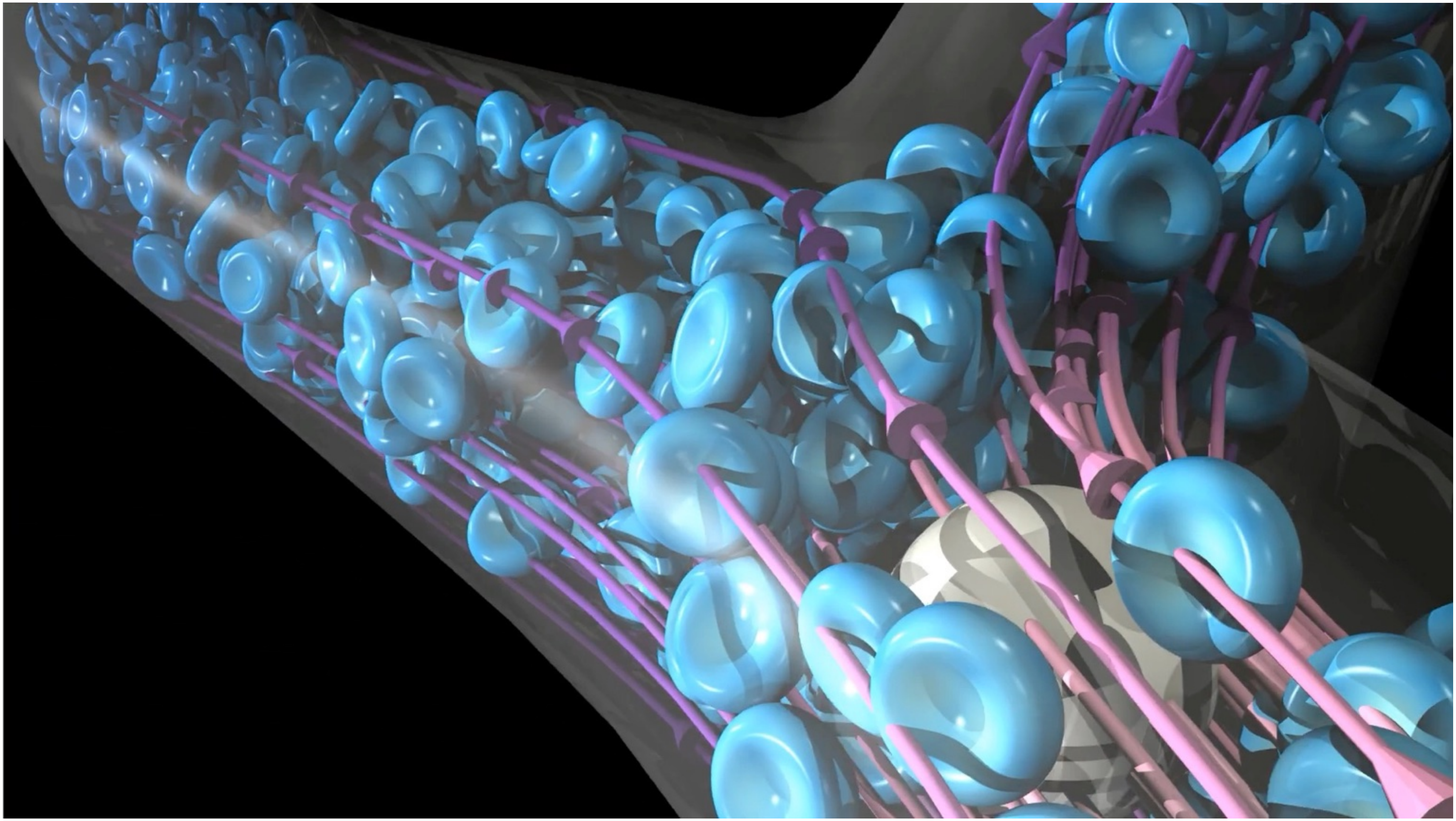
Slide courtesy of Roba Binyahib and Dave Demarle of Intel

Visualizing water flowing through a limestone karst from a South Florida ground core sample. Credit: Data courtesy of Michael Sukop, Sade Garcia, Florida International University and Kevin Cunningham, United States Geological Survey. Visualization: Carson Brownlee, Aaron Knoll, Paul Navratil, Texas Advanced Computing


Render slice
Alchemist "DG2"
Xe-core
Ray Tracing Unit
Sampler
Geometry
Rasterizer
HiZ
Pixel Backend
x10000 -> Aurora has 7.6M Ray Tracing Units
x6 + 2 SPR = node aka "blade"
x2 = GPU aka "device"
x4 = Tile aka "subdevice"
Max GPU ("PVC")
Slide courtesy of Roba Binyahib and Dave Demarle of Intel
intel



Slide courtesy of Roba Binyahib and Dave Demarle of Intel



In Situ Visualization and Analysis

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

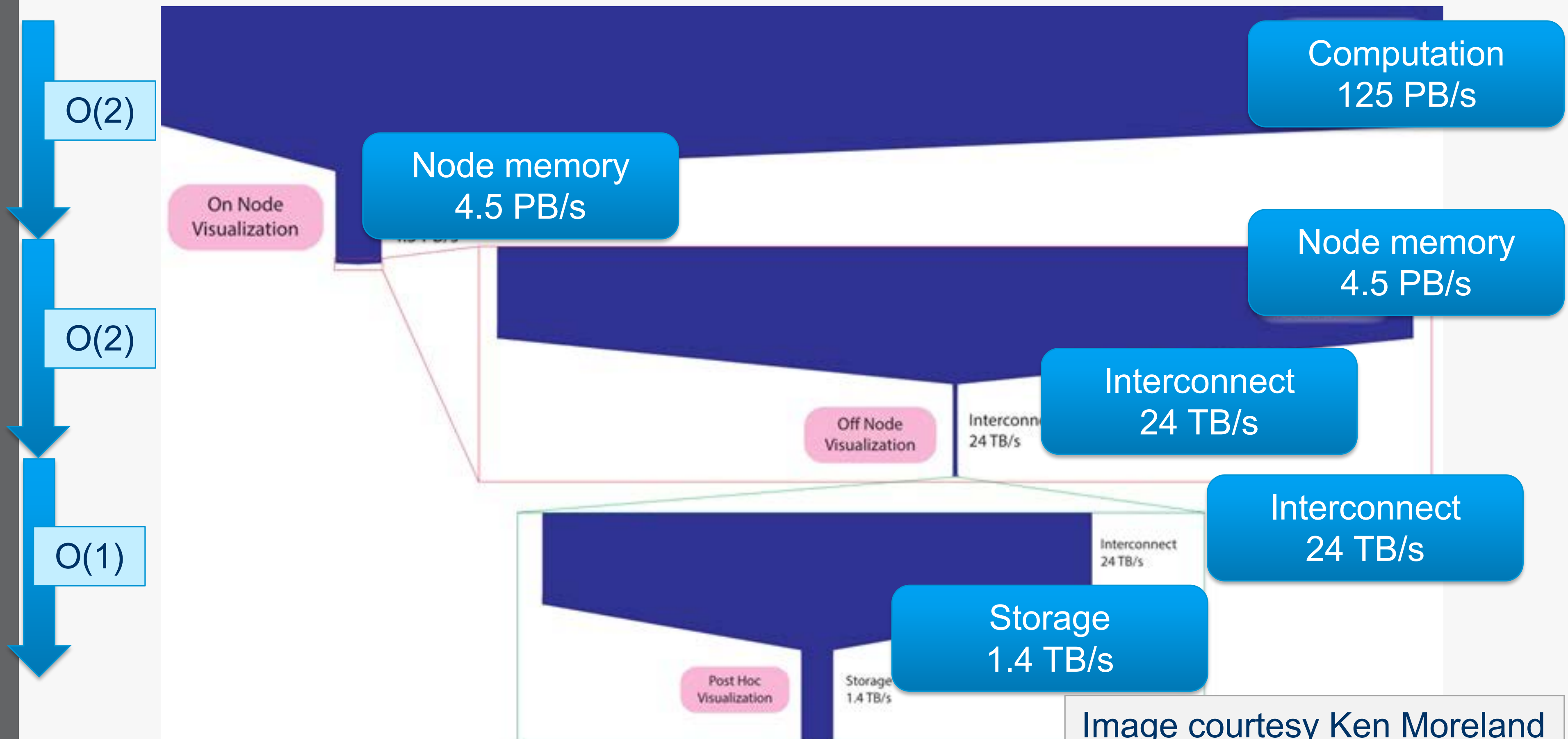


Image courtesy Ken Moreland

In Situ vis and Analysis
Problem:

FLOPS to I/O Bottleneck

– Frontier
  - Peak Performance: 1.6 EF
  - Storage: 2-4x Summit's I/O 2.5TB/s. At best 10TB/s
  - 5 orders of magnitude difference

– Aurora
  - Peak Performance: 1.012 EF
  - Storage: 31TB/s
  - 5 orders of magnitude difference

Problem

I/O is too expensive

Scientists cannot save every timestep, and/or resolution

Lost cycles: simulation waits while I/O is happening

Lost discoveries: scientists might miss discoveries

Solution: *In situ* visualization and analysis

What is *IN SITU* vis and analysis

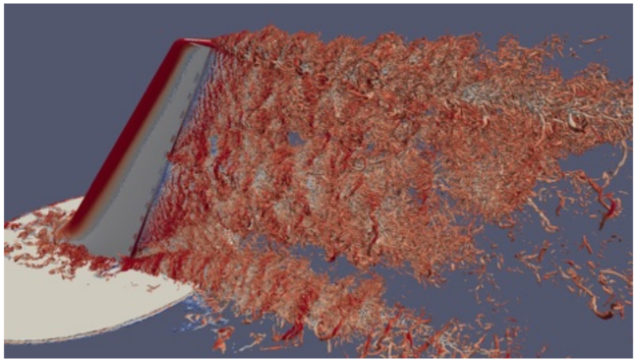
Traditionally visualization and analysis happens post hoc

- aka: Data gets saved to the disk, scientist opens it after the simulation has ended

In situ

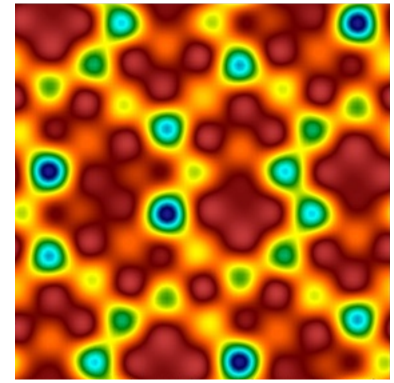
- Data gets visualized/analyzed **while** in memory.
- If zero-copy used, there is no data movement
- Ideally the data is on the GPU and stays on the GPU

In Situ

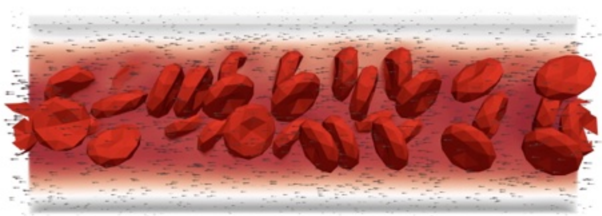


~2014
PHASTA, Catalyst,
Ken Jansen

2018
Nek5000,
SENSEI

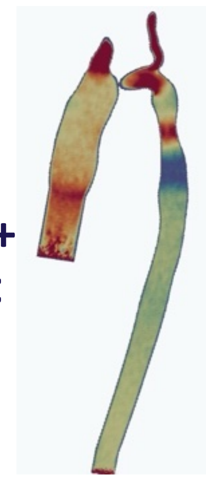


2021 - 2024

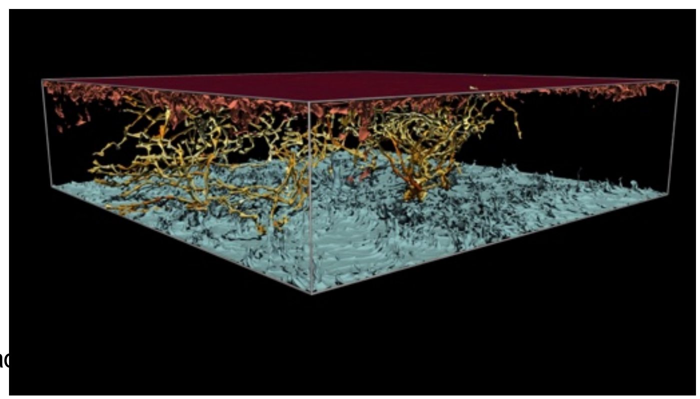


Palabos+LAMMPS,
SENSEI + Catalyst,
bi-directional

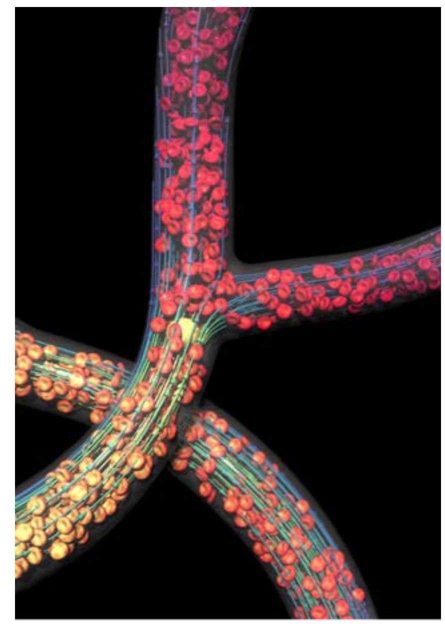
2019
SENSEI +
Catalyst



2024
nekRS,
Ascent +
Catalyst



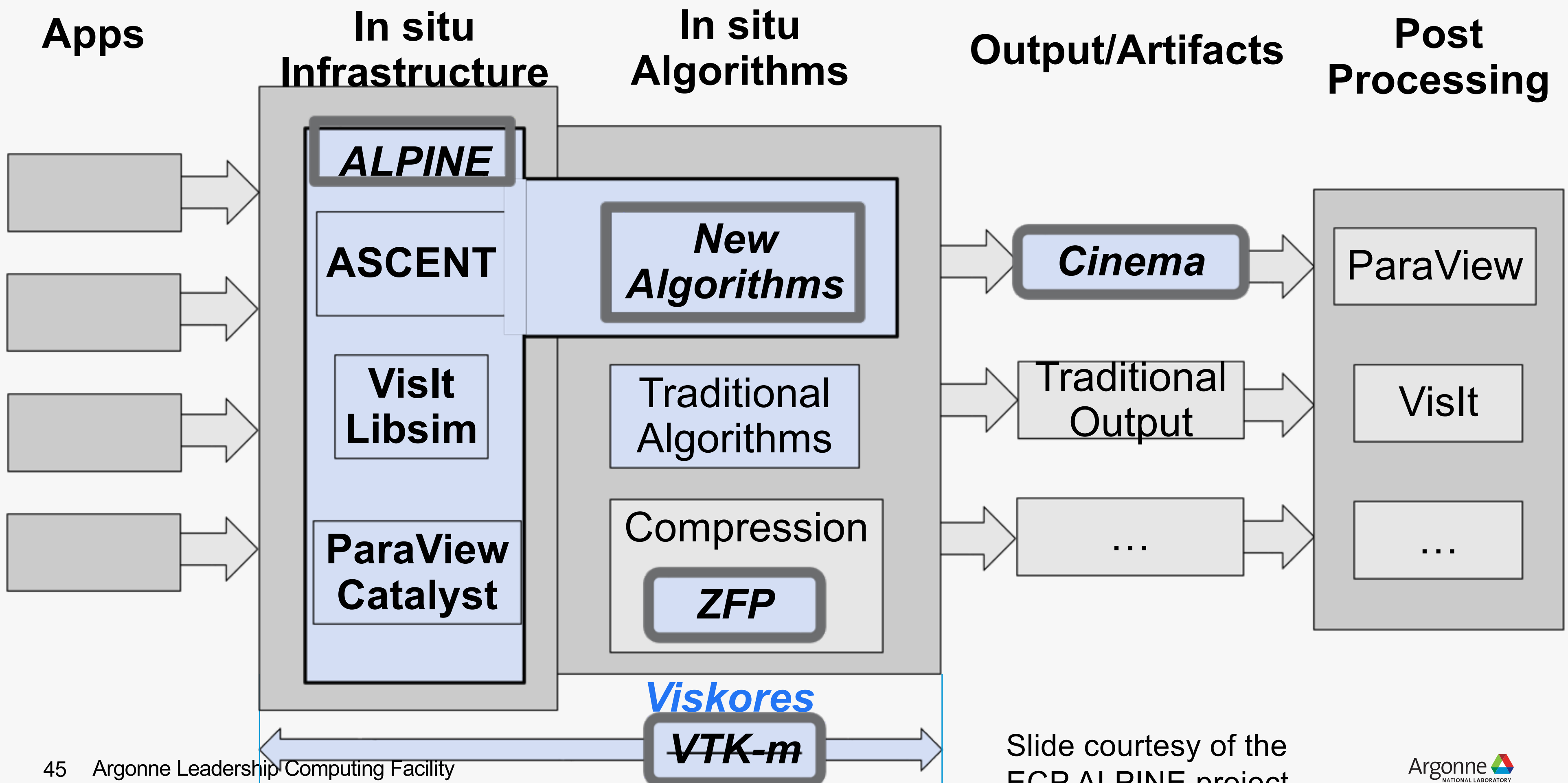
HARVEY
Ascent +
Catalyst
2024



In Situ Frameworks and Infrastructures at ALCF

Name	Description	Contact person at ATPESC
Ascent	The Ascent in situ infrastructure is designed for leading-edge supercomputers, and has support for both distributed-memory and shared-memory parallelism.	Cyrus Harrison, Justin Privitera
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
Cinema	Cinema is an innovative way of capturing, storing, and exploring both extreme scale scientific data and experimental data. It is a highly interactive image-based approach to data analysis and visualization that promotes investigation of large scientific datasets.	Joe Insley, Silvio Rizzi
Viskores	Viskores is a toolkit of scientific visualization algorithms for emerging processor architectures. Viskores supports the fine-grained concurrency for data analysis and visualization algorithms required to drive extreme scale computing.	Joe Insley, 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

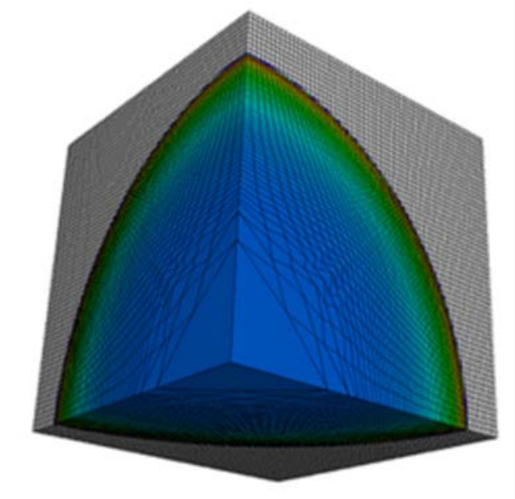
```
//
// Run Ascent
//

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

Slide courtesy of the Ascent team

Catalyst 2.0 : Rethinking the ParaView in situ data analysis and visualization API

- **inspired by MPICH ABI compatibility initiative**
- **leverages Conduit**

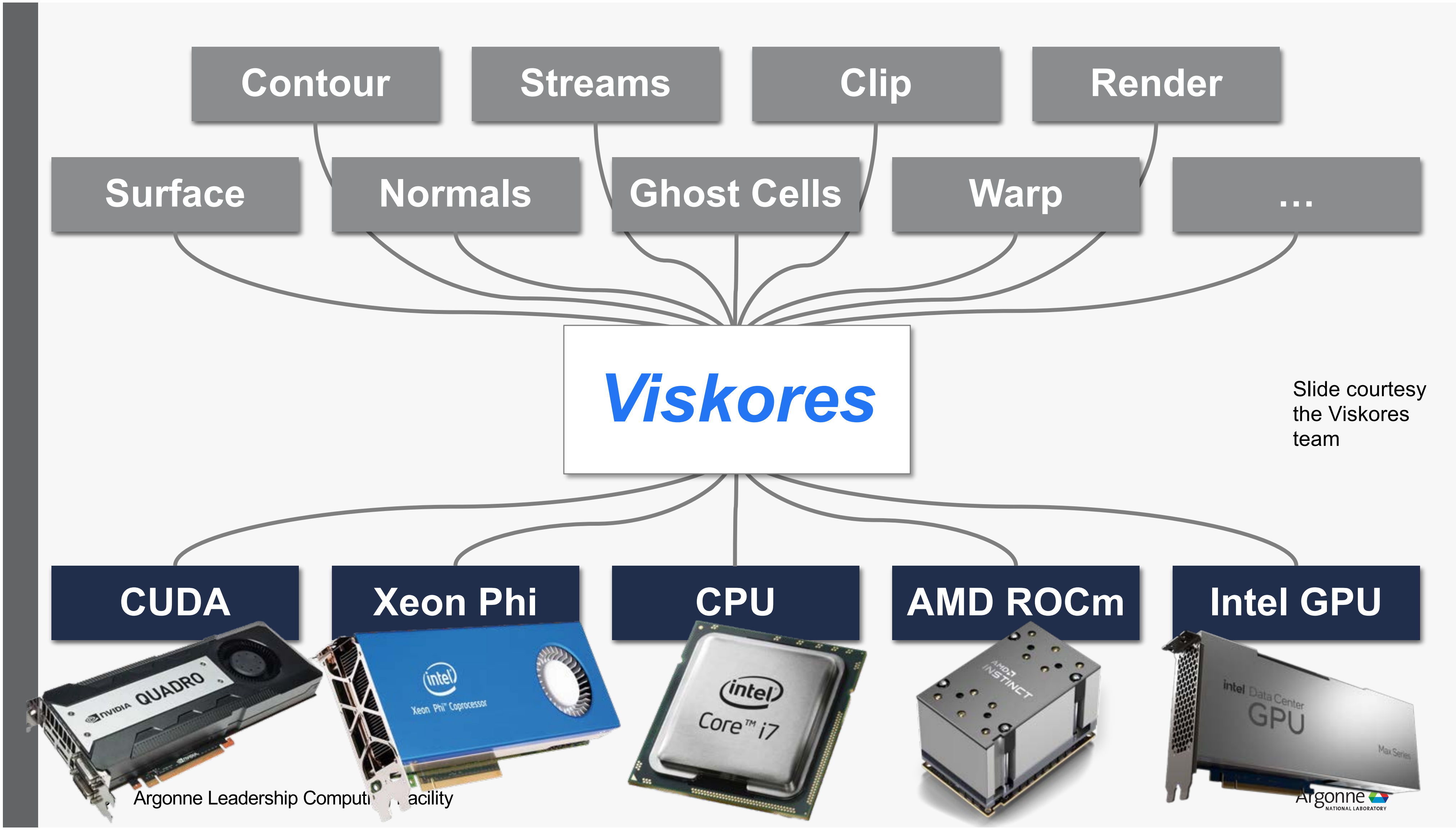
- ***Catalyst***: a new project with ***no*** external dependencies: https://gitlab.kitware.com/paraview/catalyst
- Gitlab-CI enabled, ReadTheDocs-enabled
- Provides API ***specification***, ***stub*** implementation, ***SDK*** for developing adaptors and custom implementations



Lulesh, a miniapp representing a typical hydrodynamics simulation code, instrumented with Catalyst

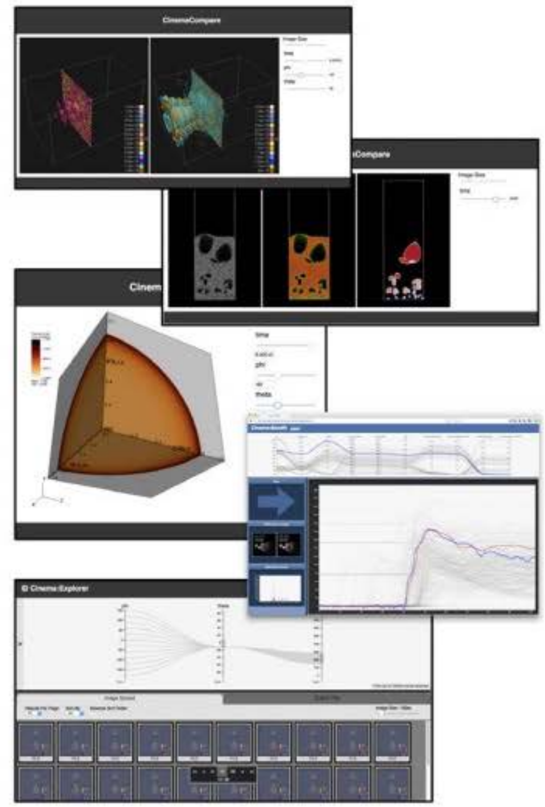
Source: https://gitlab.kitware.com/paraview/catalyst
Docs: https://catalyst-in-situ.readthedocs.io/en/latest/

Contents of this slide courtesy of Utkarsh Ayachit (Nvidia Corp., formerly Kitware, Inc.)


Contour
Streams
Clip
Render
Surface
Normals
Ghost Cells
Warp
...
Viskores
Slide courtesy the Viskores team
CUDA
Xeon Phi
CPU
AMD ROCm
Intel GPU
Argonne Leadership Computi
acility
Argonne

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

QUESTIONS?

Joe Insley
insley@anl.gov

Silvio Rizzi
srizzi@anl.gov