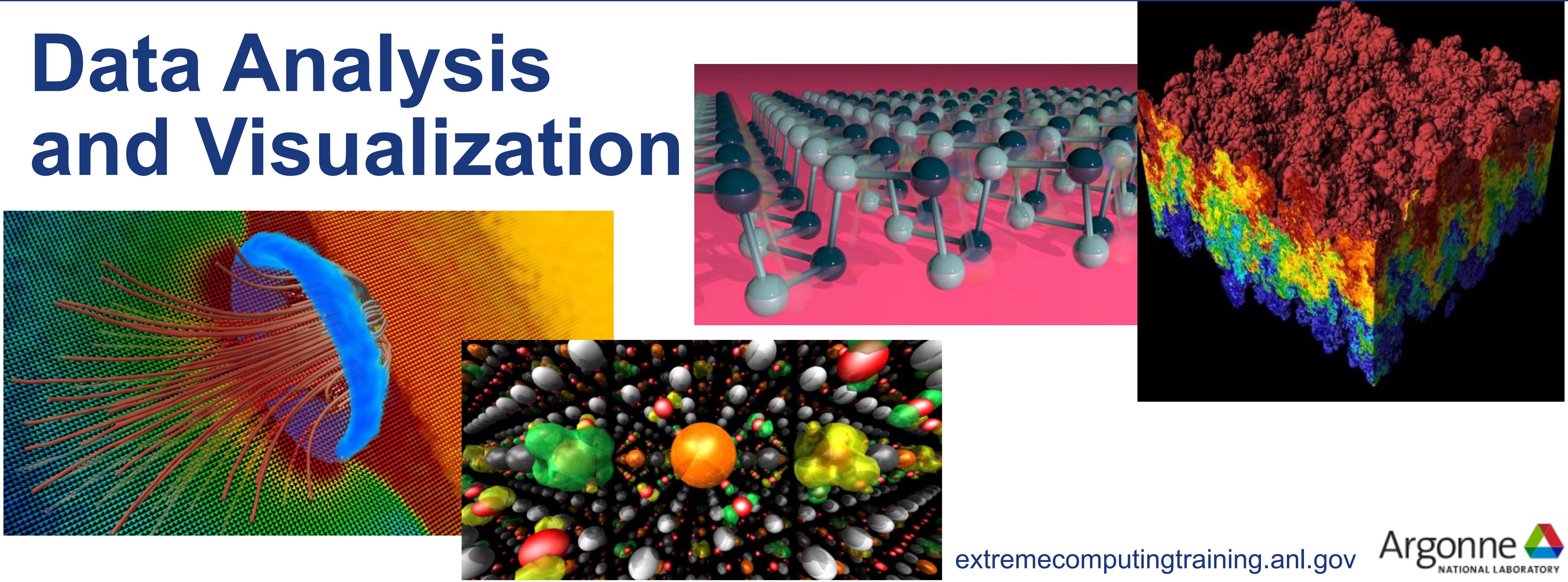
ARGONNE

ATPESC2024

EXTREME-SCALE COMPUTING

Data Analysis and Visualization



extremecomputingtraining.anl.gov

Argonne NATIONAL LABORATORY

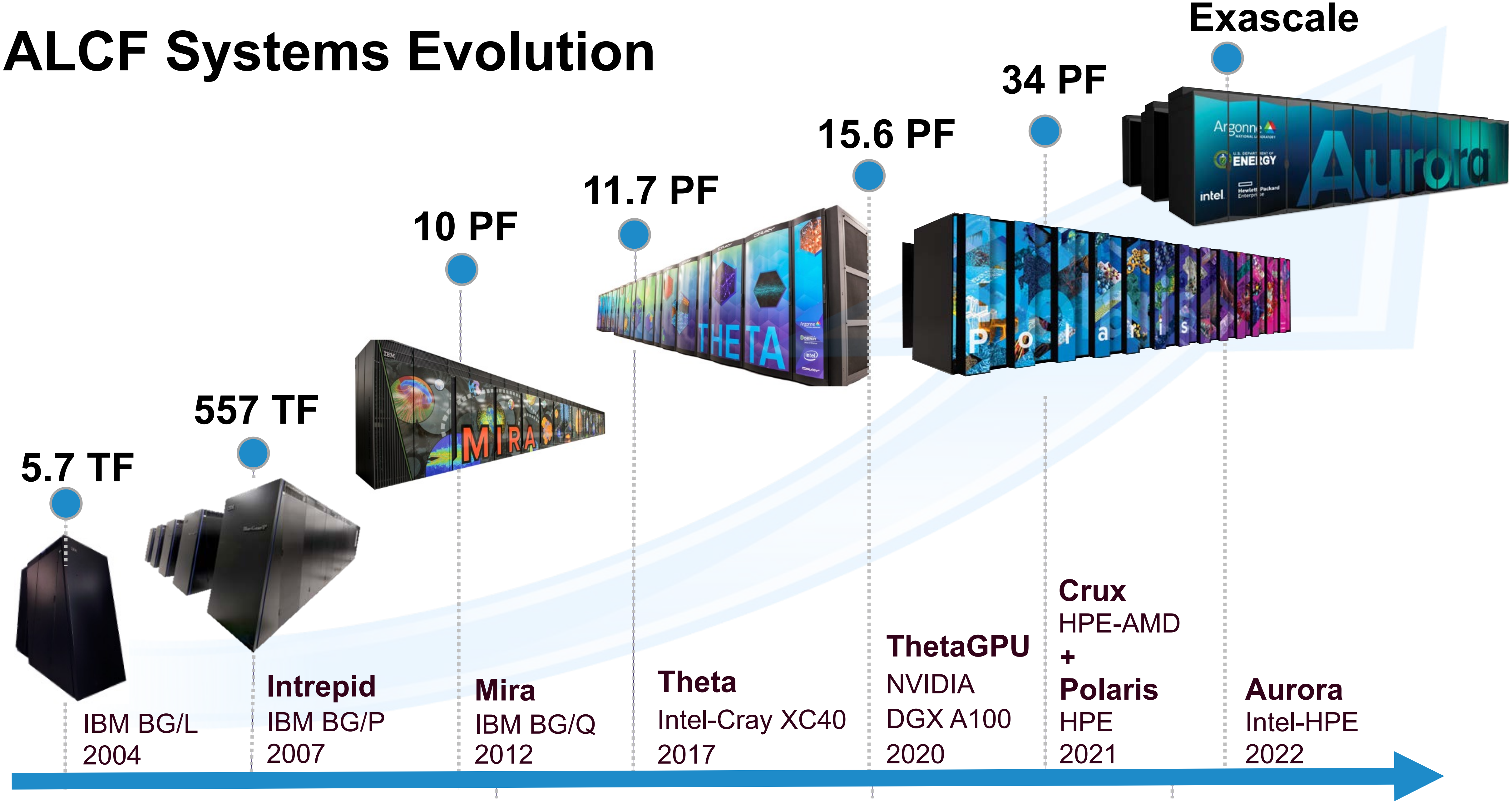
Visualization & Data Analysis

Time	Title of presentation	Lecturer
8:30 am	**Data Analysis and Visualization Introduction**	Joe Insley *ANL/NIU*, Silvio Rizzi *ANL,* Victor Mateevitsi, *ANL*
9:30 am	**Large Scale Visualization with ParaView**	Dan Lipsa *Kitware*
10:00 am	*Break*	
10:30 am	**Large Scale Visualization with ParaView (Cont.)**	Dan Lipsa *Kitware*
11:30 am	**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*
2:00 pm	**Exploring Visualization with Jupyter Notebooks**	David Koop *NIU*
2:45 pm	**Ascent**	Cyrus Harrison *LLNL* Andres Sewell *Utah State*
3:45 pm	*Break*	
4:15 pm	**Trame**	Patrick Avery *Kitware*
4:45 pm	**NOODLES**	Nicholas Brunhart-Lupo *NREL*
5:30 pm	*Hands-on*	All
6:30 pm	*Dinner*	
7:30 pm	*After-dinner talk*	Brad Carvey *Sandia*

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



5.7 TF
IBM BG/L
2004

557 TF
Intrepid
IBM BG/P
2007

10 PF
Mira
IBM BG/Q
2012

11.7 PF
Theta
Intel-Cray XC40
2017

15.6 PF
ThetaGPU
NVIDIA
DGX A100
2020

34 PF
Crux
HPE-AMD
+
Polaris
HPE
2021

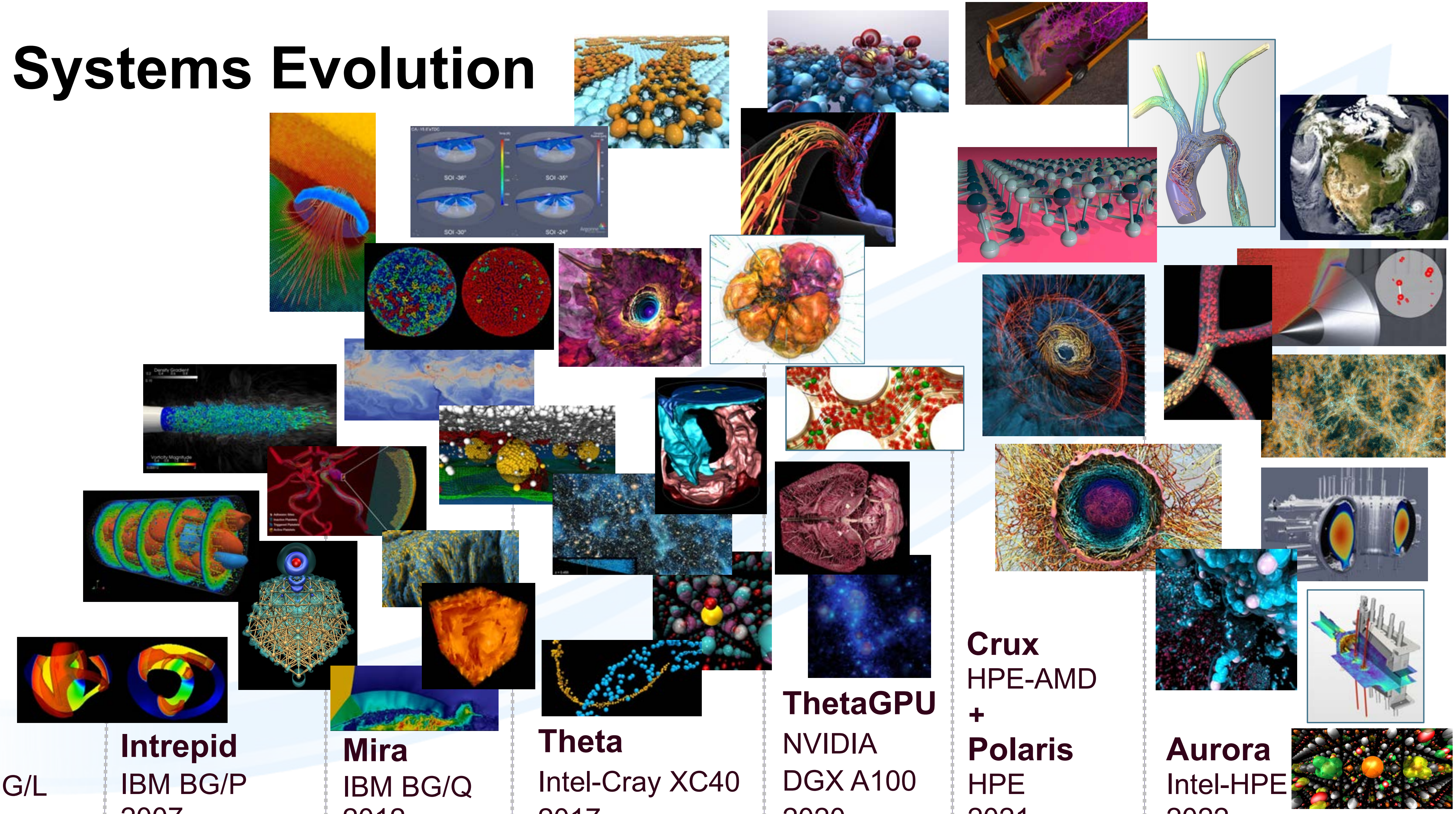
Exascale
Aurora
Intel-HPE
2022

JLSE (2013)

AI Testbed (2020)

Edge Testbed (2021)

ALCF Systems Evolution



IBM BG/L
2004

Intrepid
IBM BG/P
2007

Mira
IBM BG/Q
2012

Theta
Intel-Cray XC40
2017

ThetaGPU
NVIDIA
DGX A100
2020

Crux
HPE-AMD
+
Polaris
HPE
2021

Aurora
Intel-HPE
2022

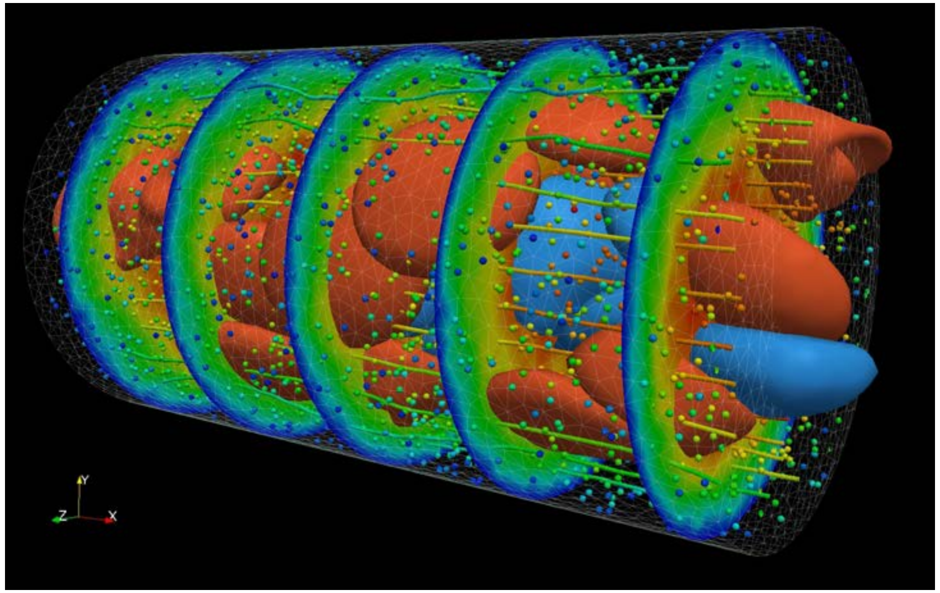
JLSE (2013)

AI Testbed (2020)

Edge Testbed (2021)

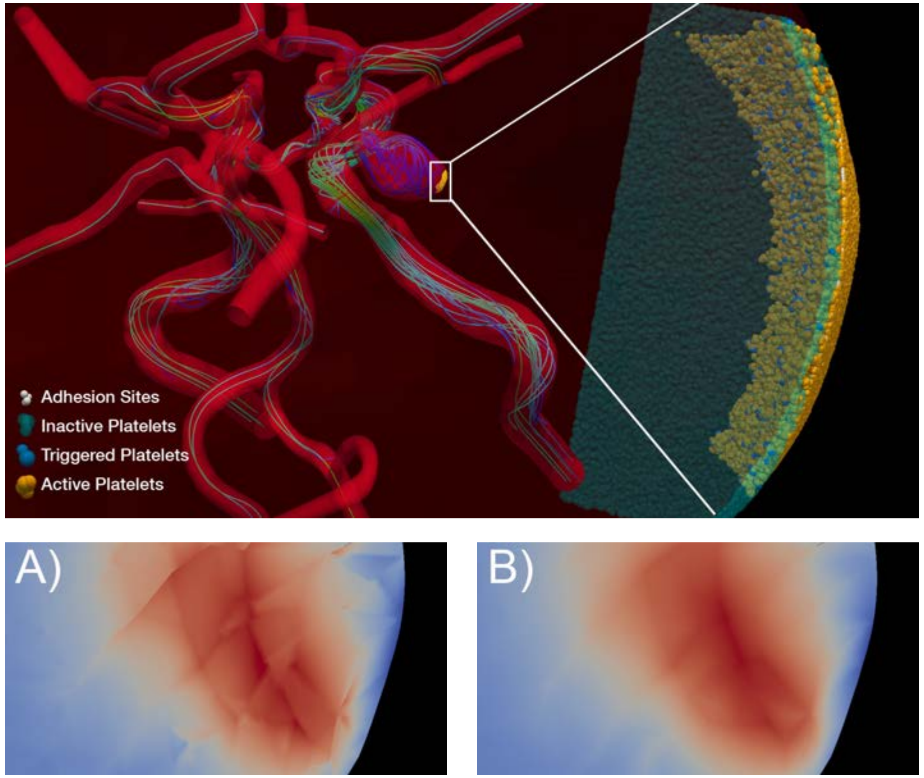
Multi-Scale Simulation / Visualization Arterial Blood Flow

PI: George Karniadakis, Brown University

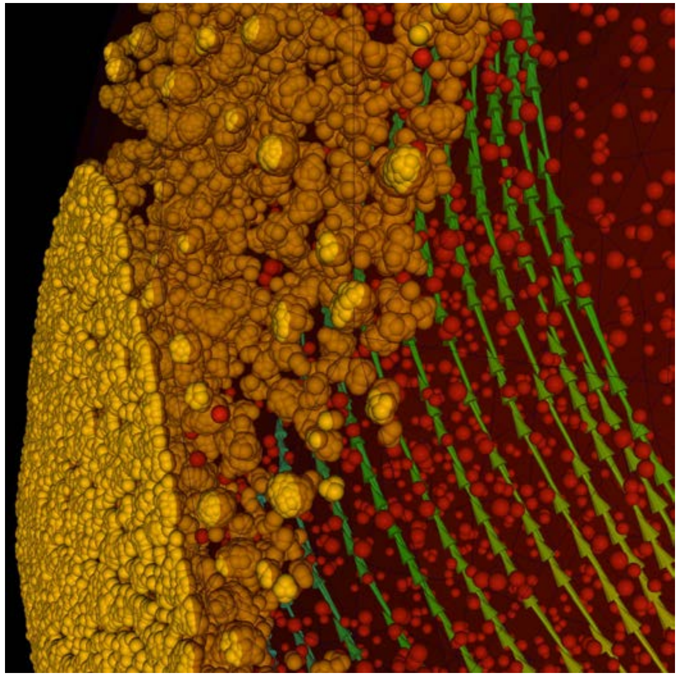


2010

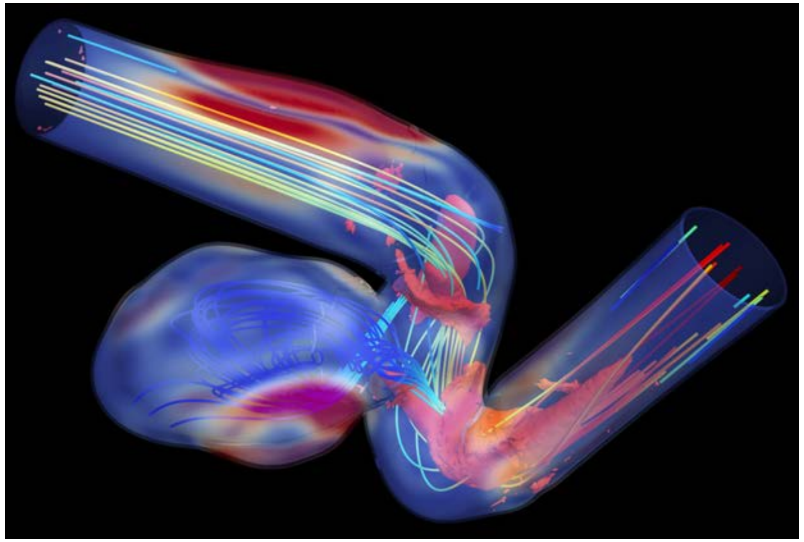
2011



2012

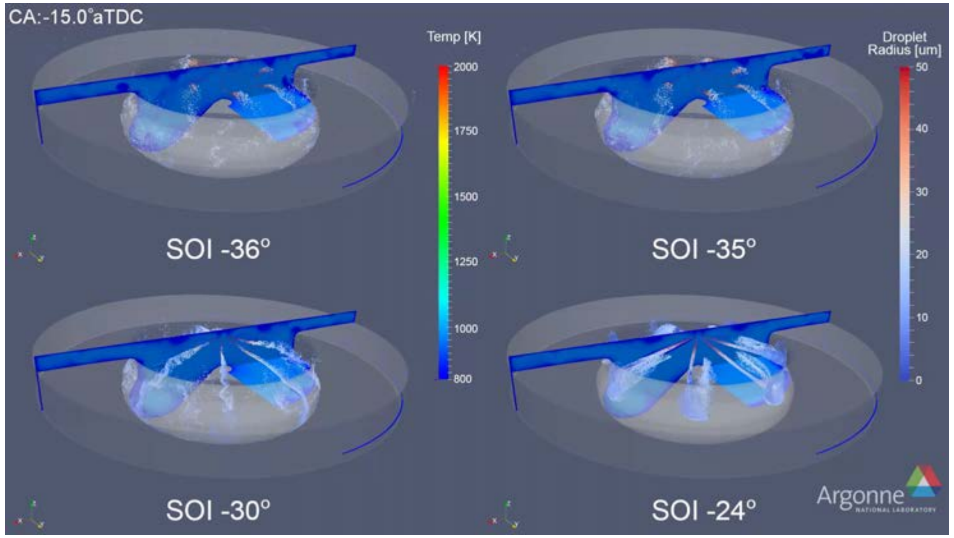


2014

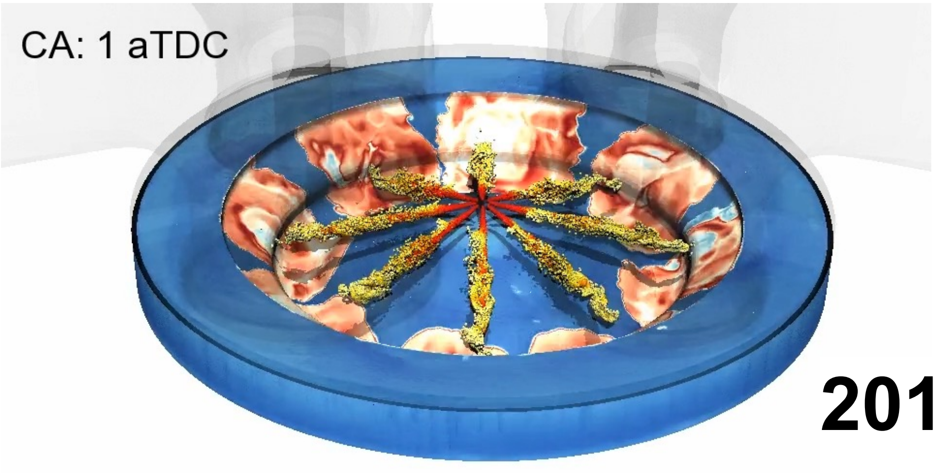


Engineering / Combustion / Biofuels

PI: Sibendu Som, Argonne National Laboratory

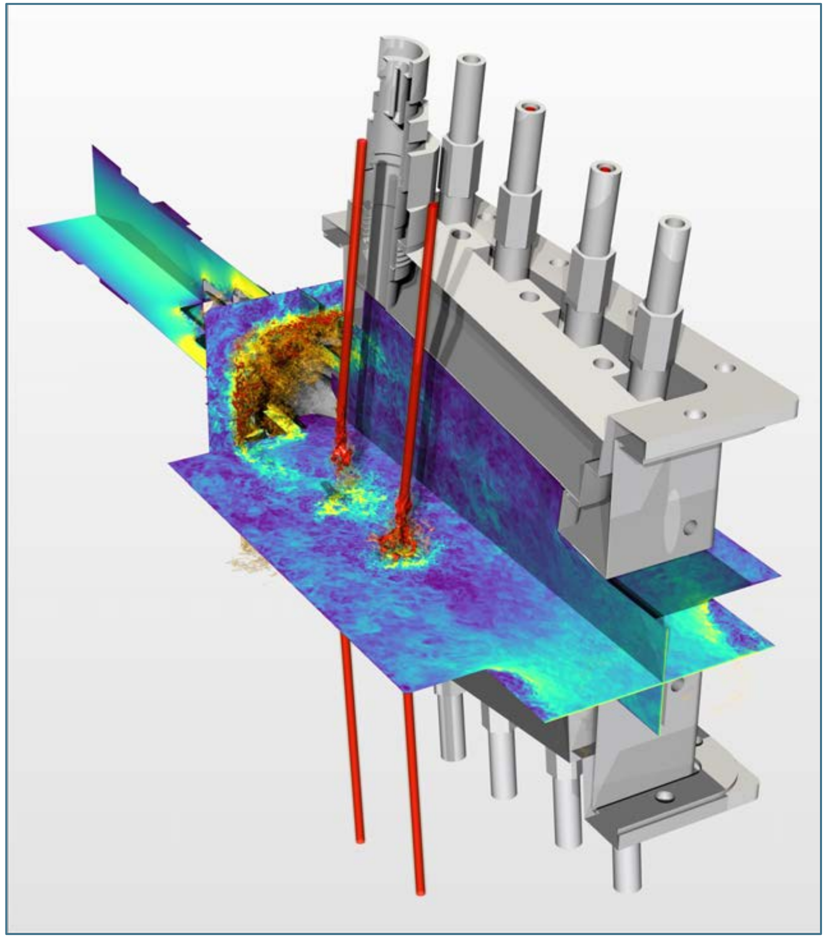


2015



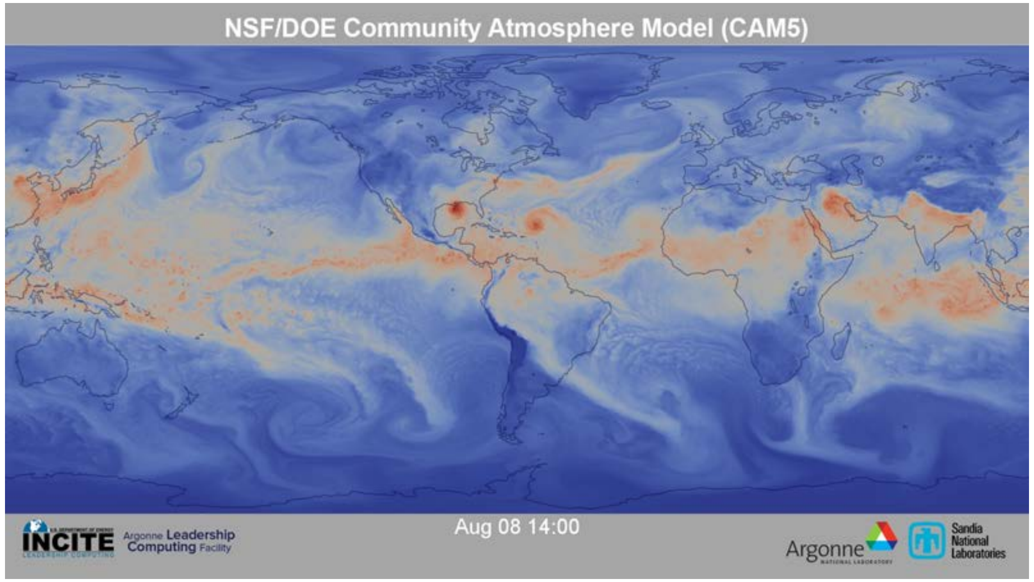
2017

2021



2023

Climate



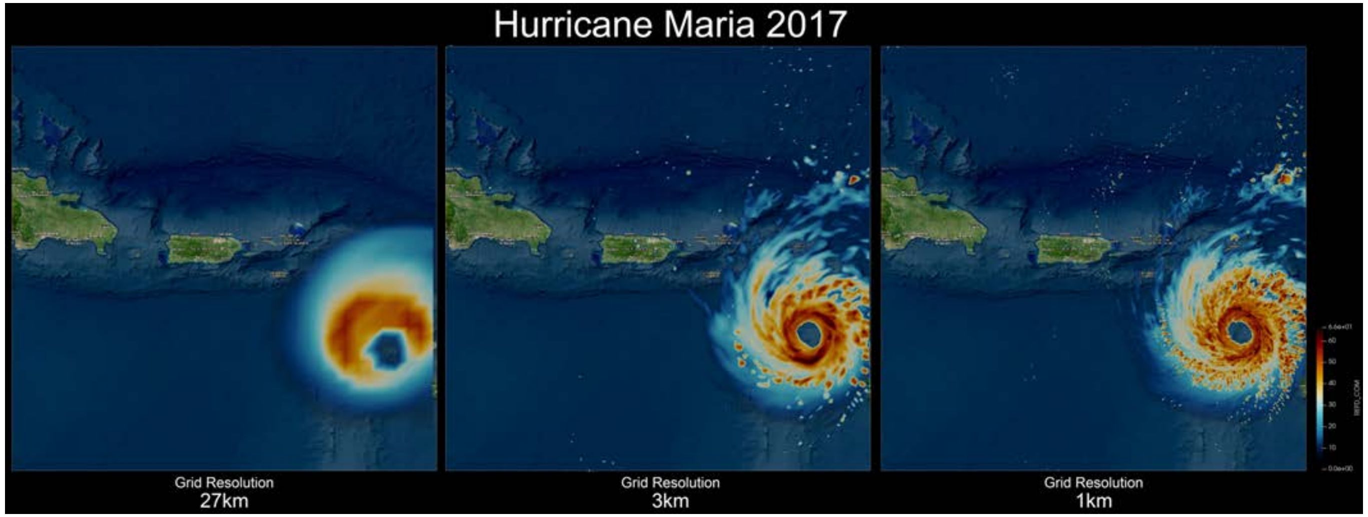
PI: Warren Washington, National Center for Atmospheric Research

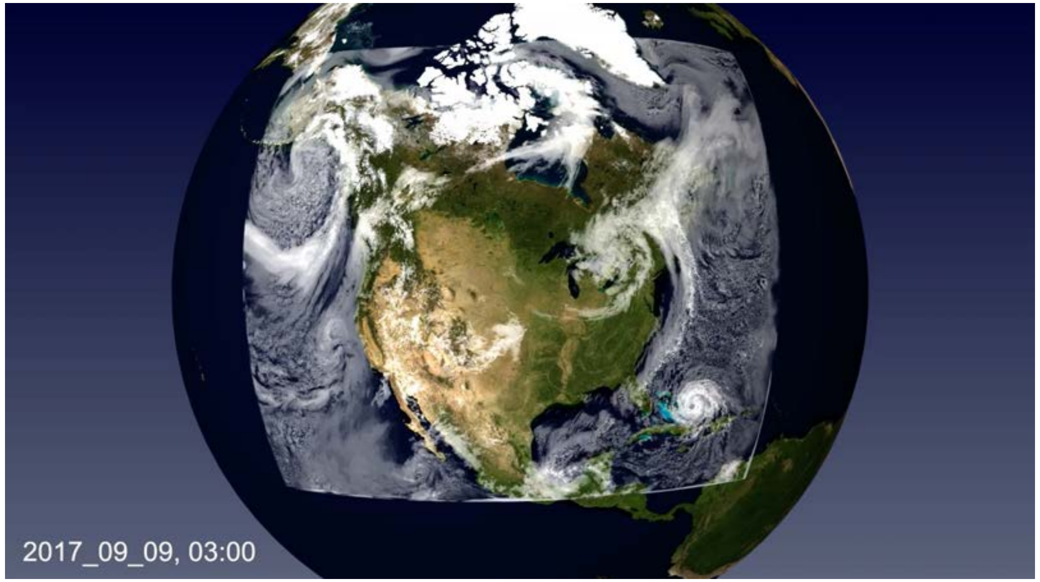
2012

PI: Rao Kotamarthi,
Argonne National Laboratory

2024

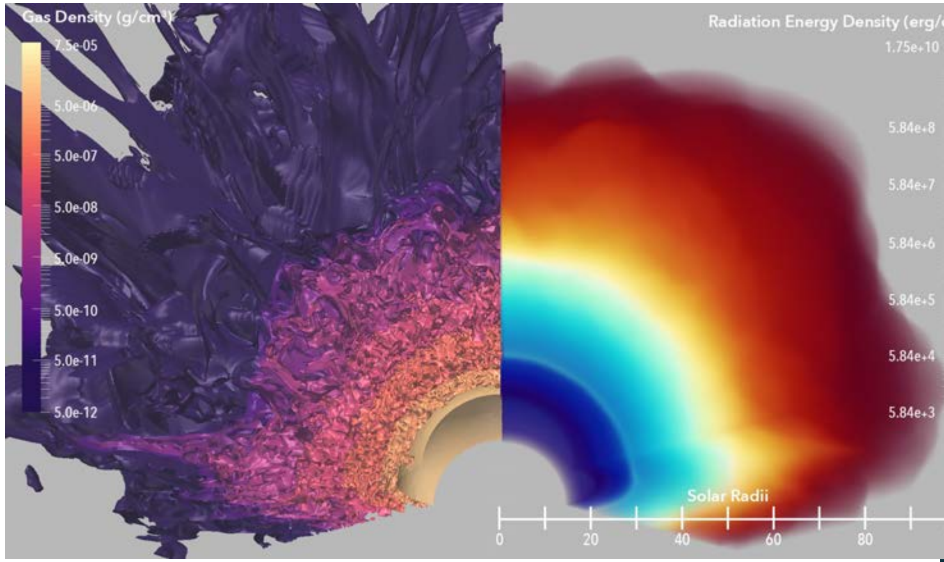
2022





Physics: Stellar Radiation

PI: Lars Bildsten, University of California, Santa Barbara



2017

2018

OUTLOOK
Brain cancer

nature

THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE

STELLAR OUTBURST

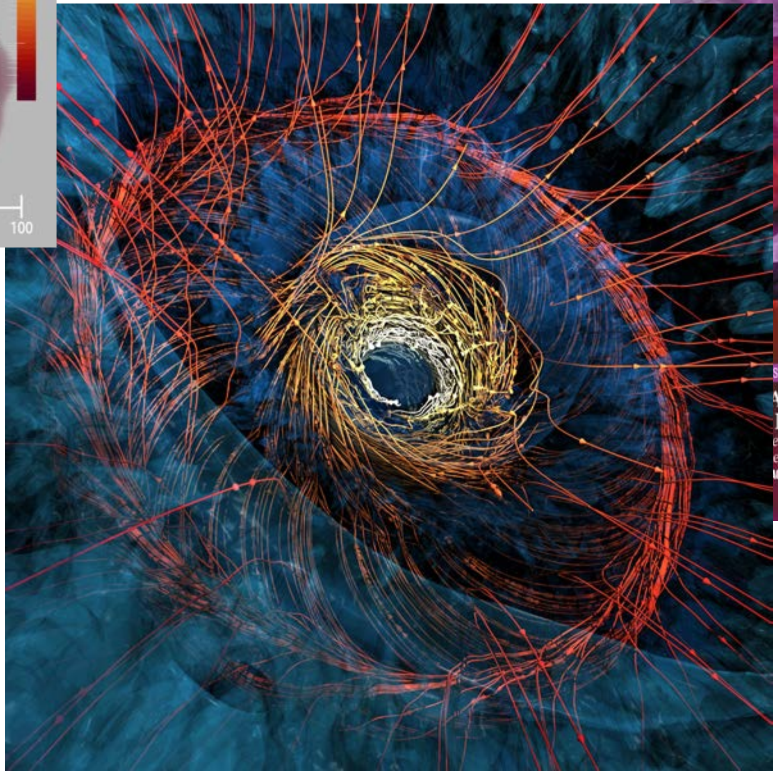
3D simulations reveal source of variability in massive blue stars
PAGE 498

SCIENCE
AL OR
DEAL
es on Brexit six
efore the split
AGE 452

MEDICAL RESEARCH
WHAT'S IN A NAME?
Time to halt the scourge of mesenchymal stem cells
PAGE 455

BIOCHEMISTRY
ROLL OUT THE β-BARREL
Targeted protein binding by computational design
PAGES 471 & 485

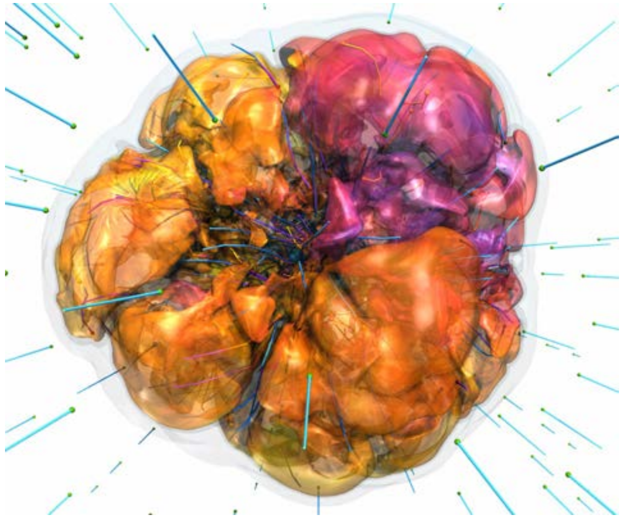
NATURE.COM
27 September 2018 £10
Vol. 561, No. 7724



2021

Astrophysics

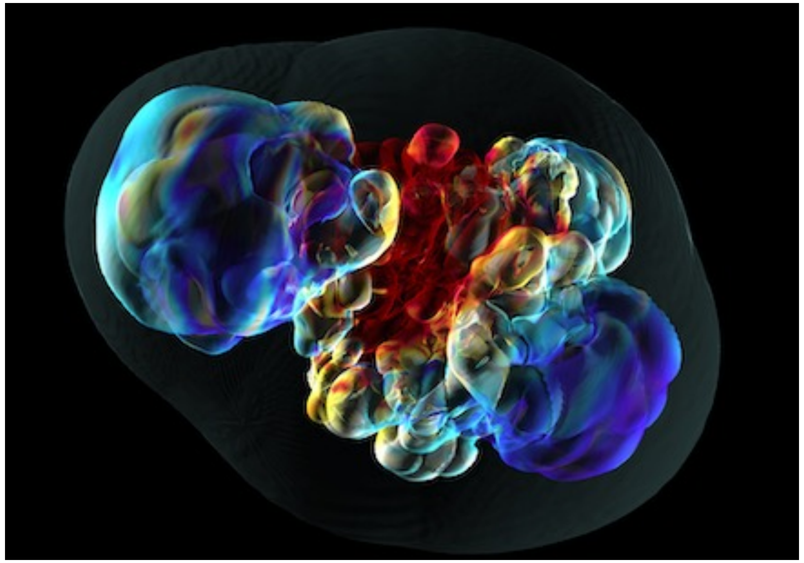
PI: Adam Burrows, Princeton University



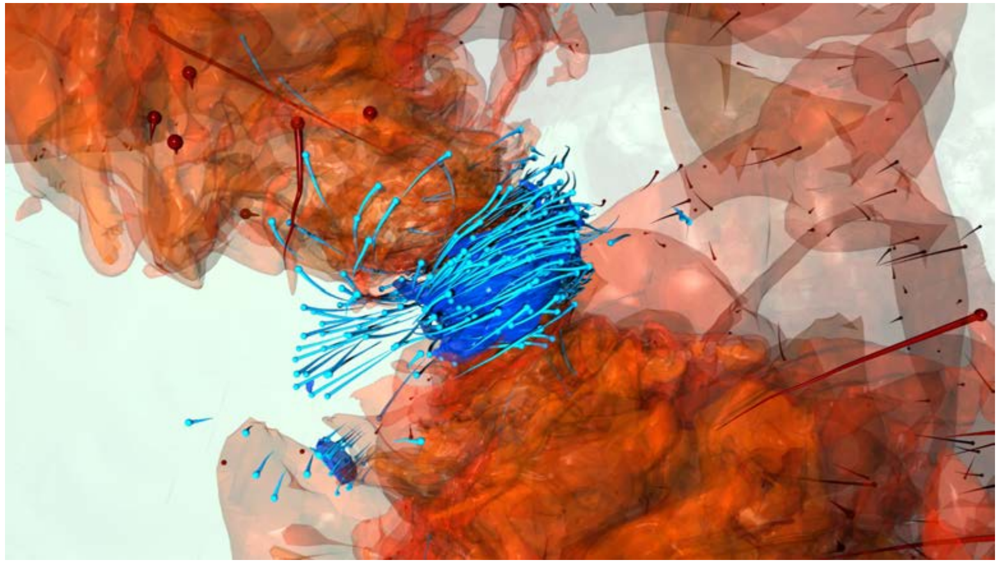
2019

2021

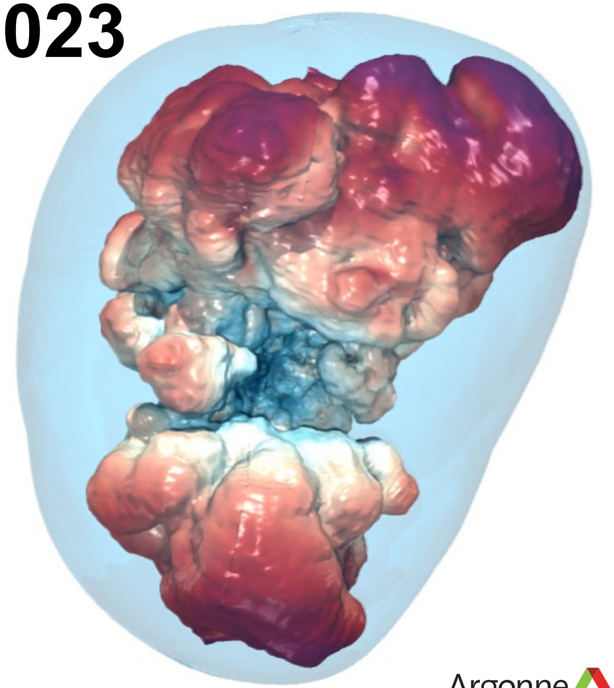
2022



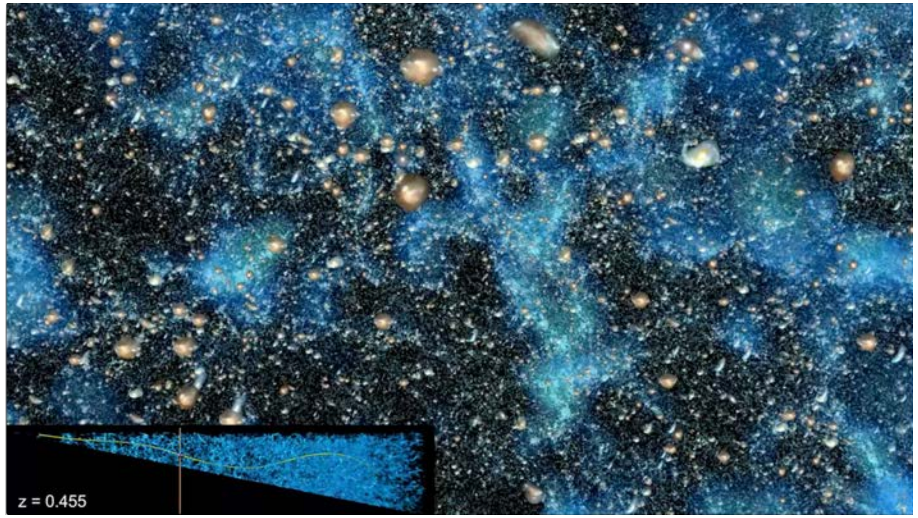
2023



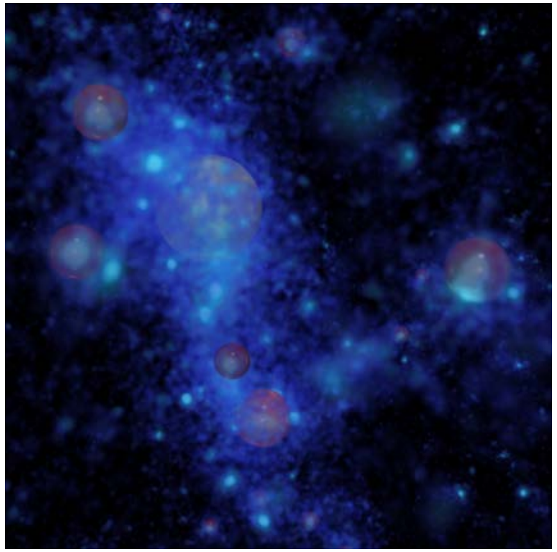
2023



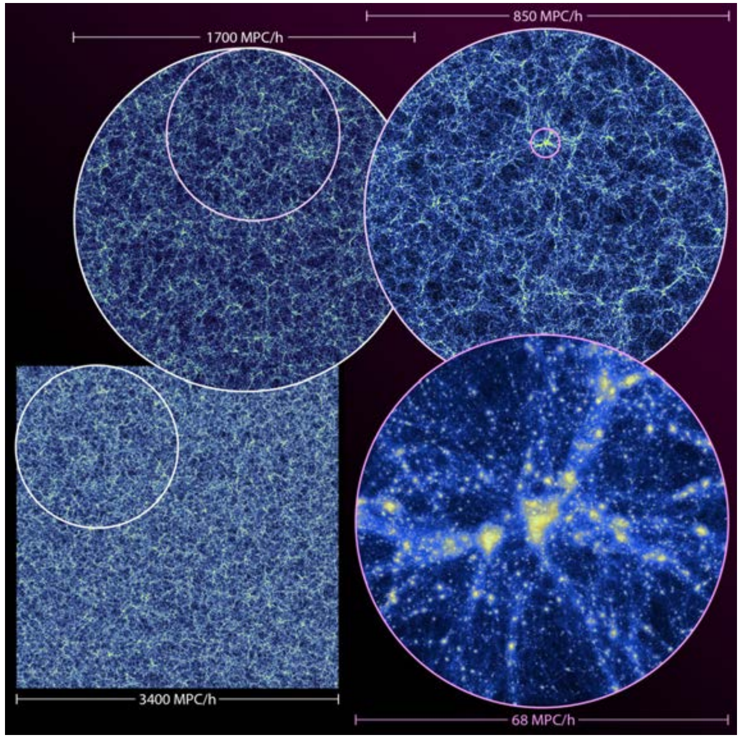
HACC: Cosmology



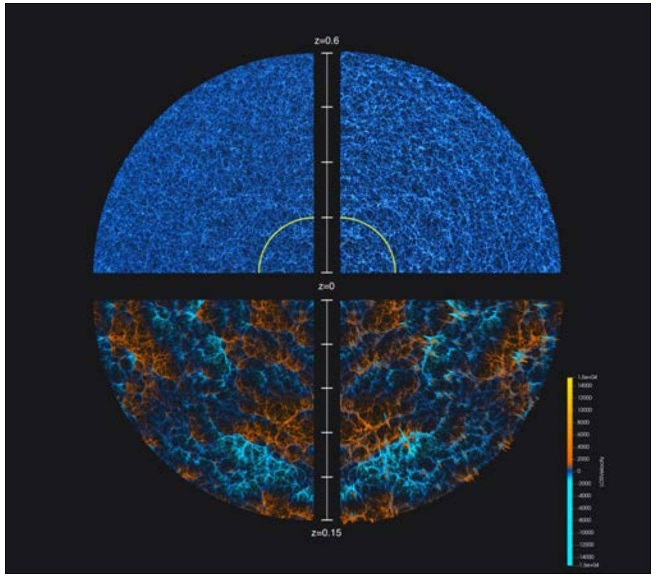
2018



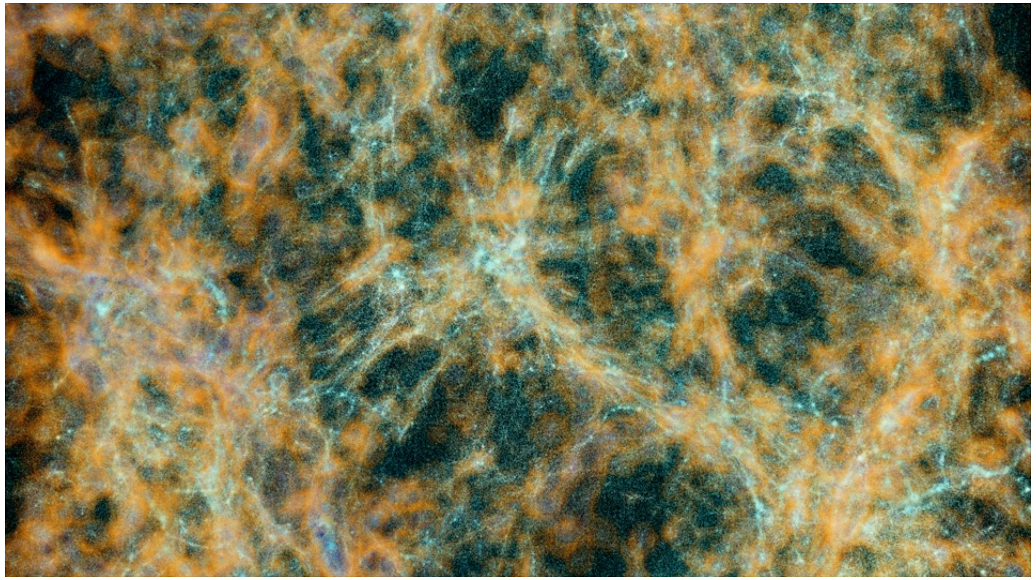
2020



2020



2021



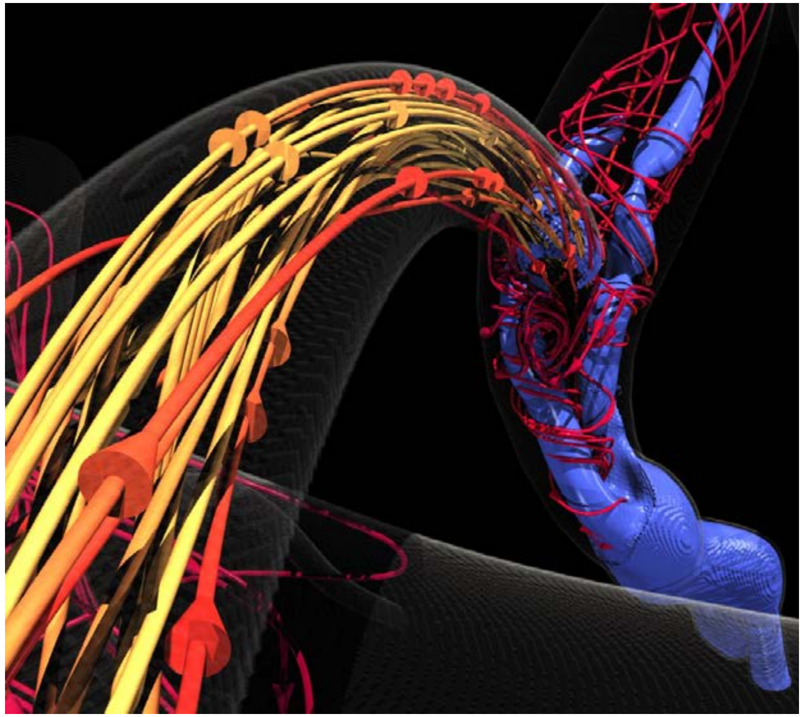
Computed and Rendered on Aurora

2023

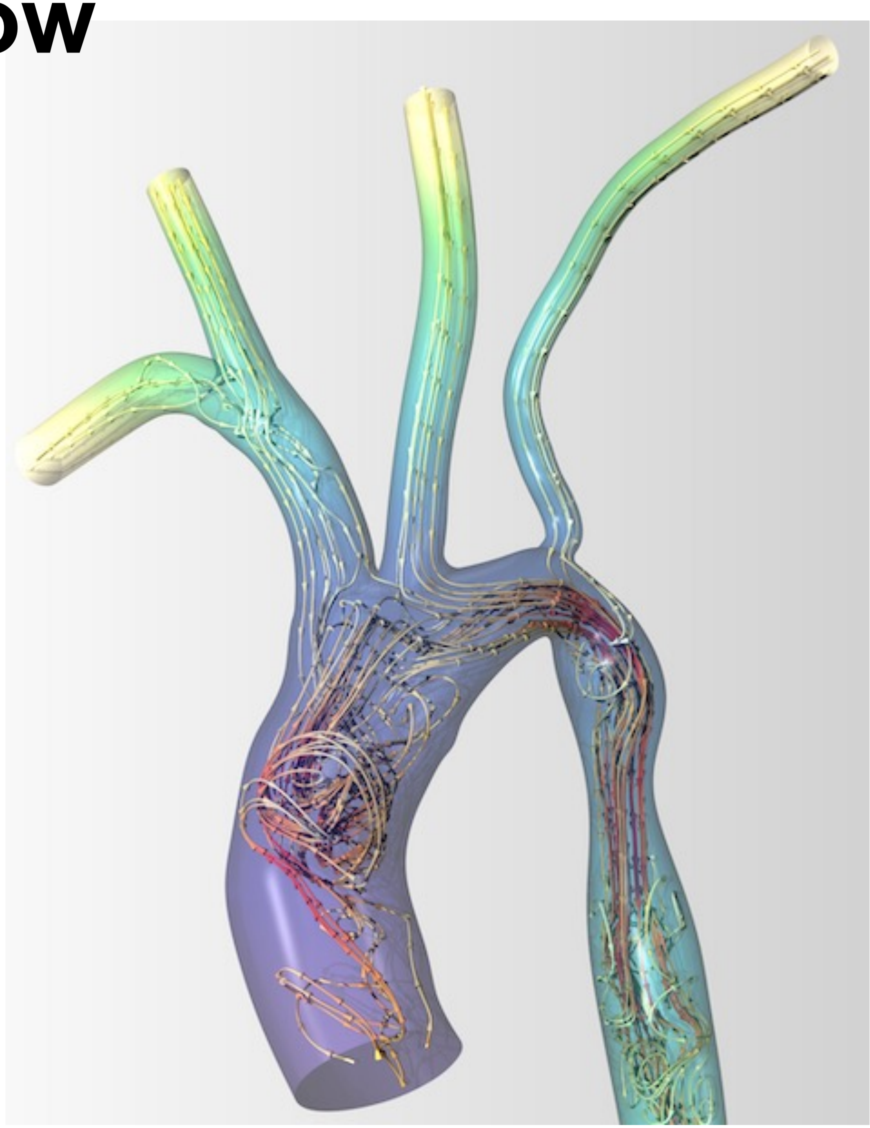
PI: Salman Habib and HACC Team, Argonne National Laboratory

Arterial Blood Flow

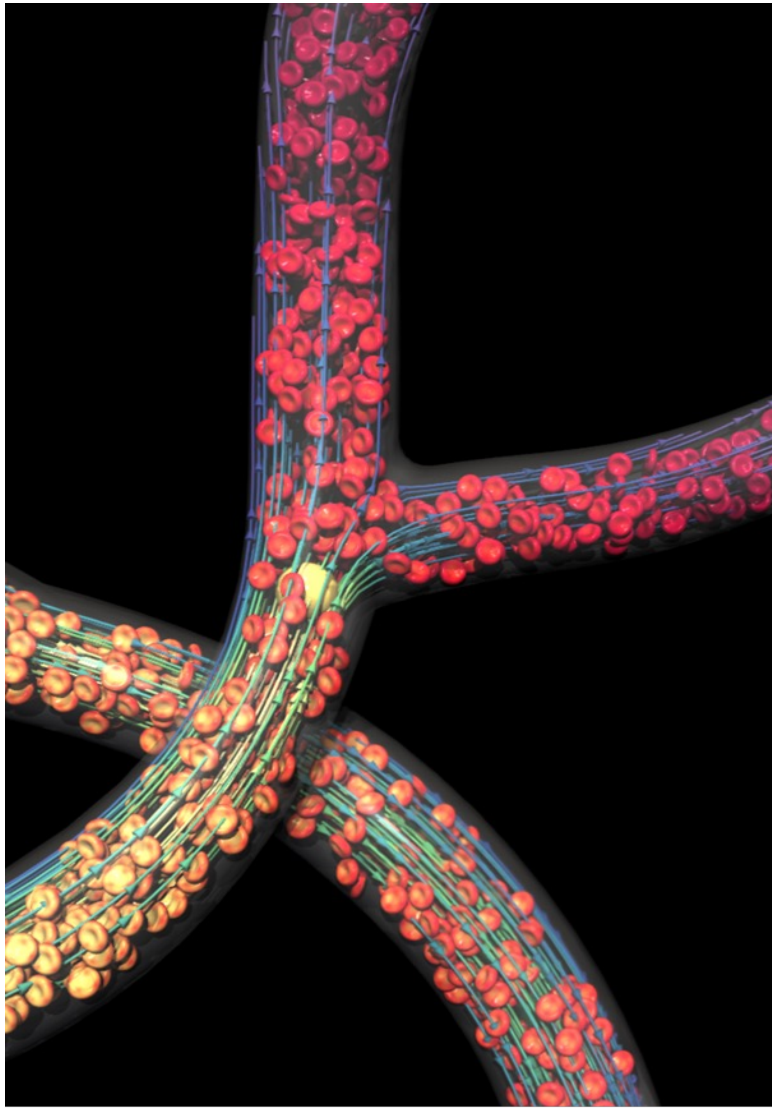
PI: Amanda Randles, Duke University



2020

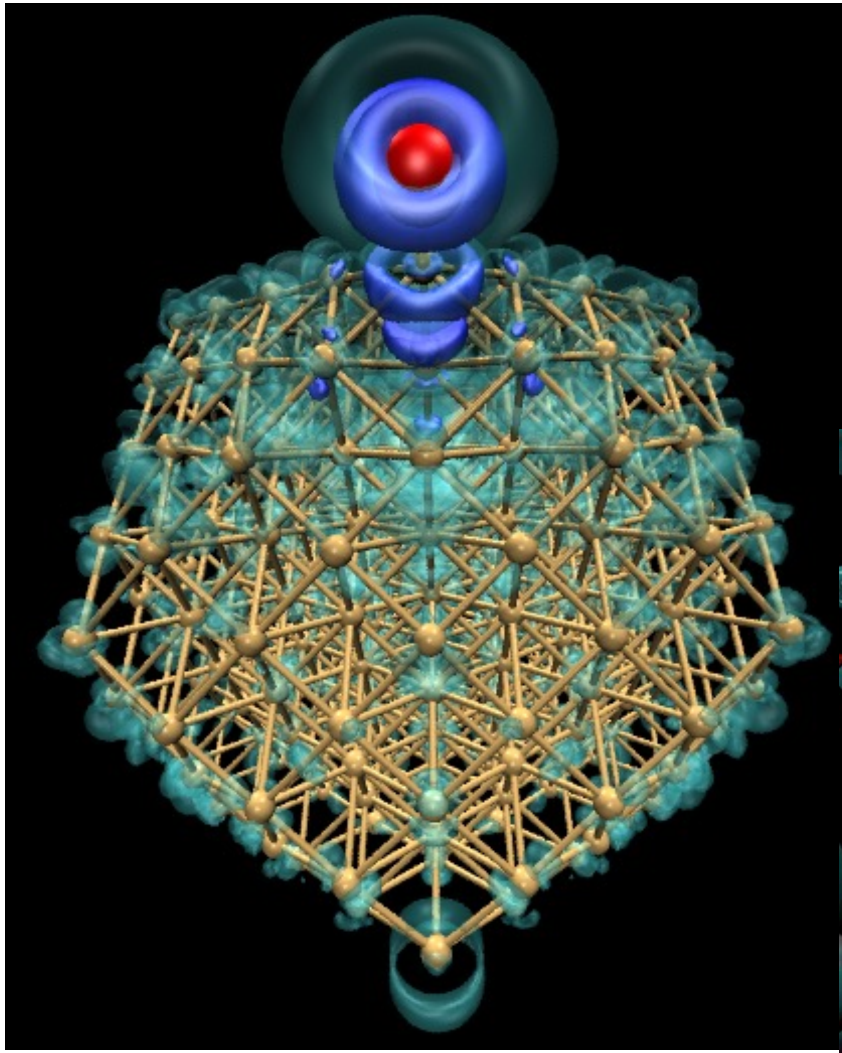


2023 **Rendered on Aurora**



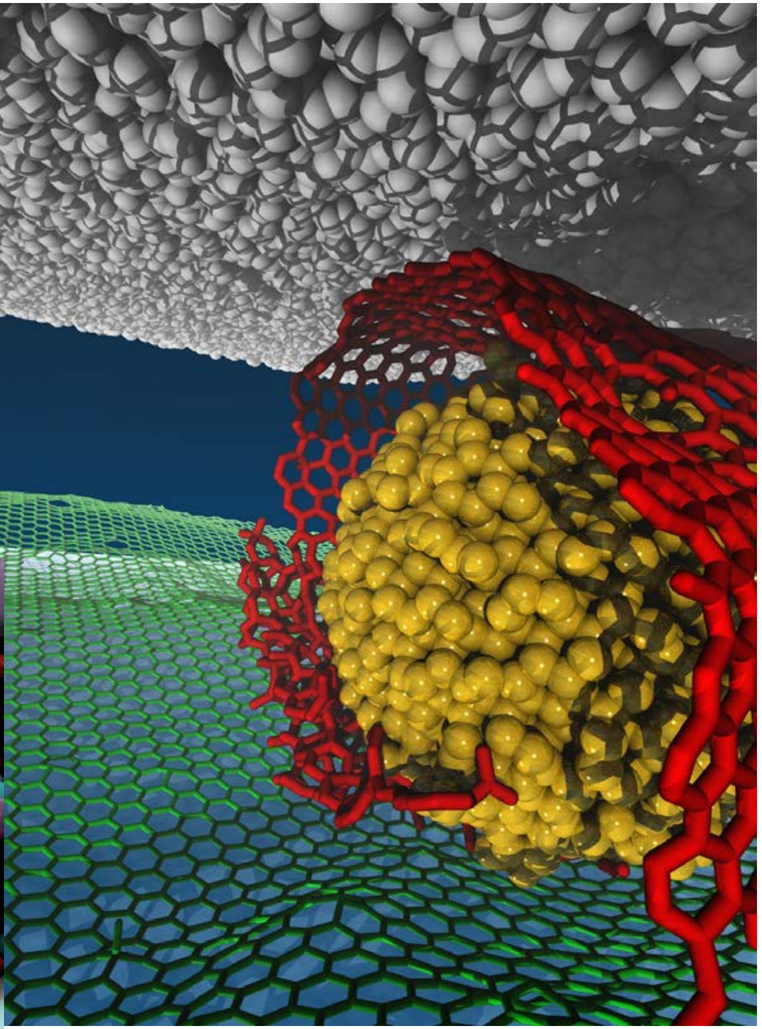
2023 **Rendered on Aurora**

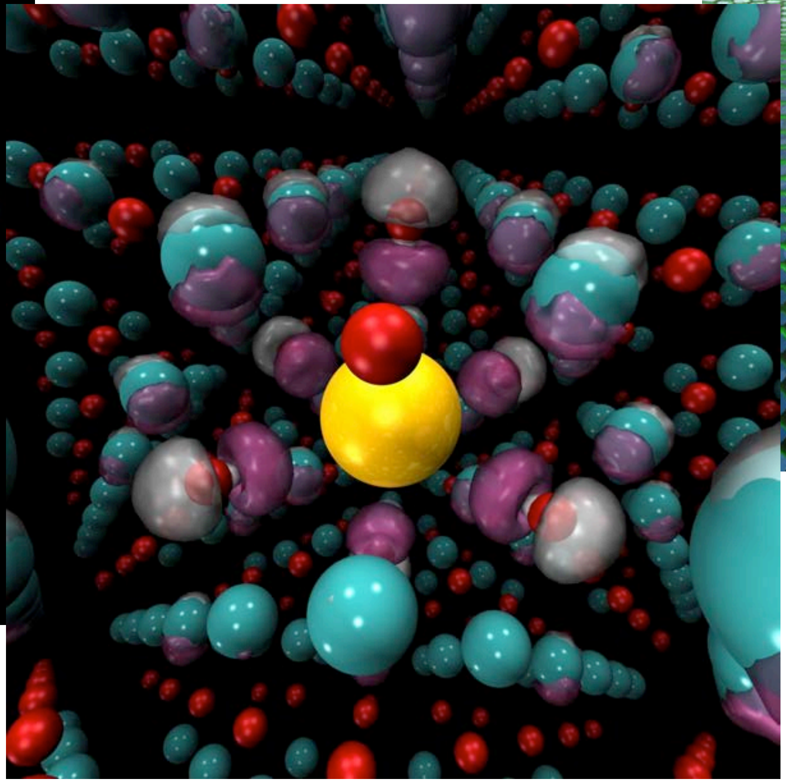
Materials Science / Molecular



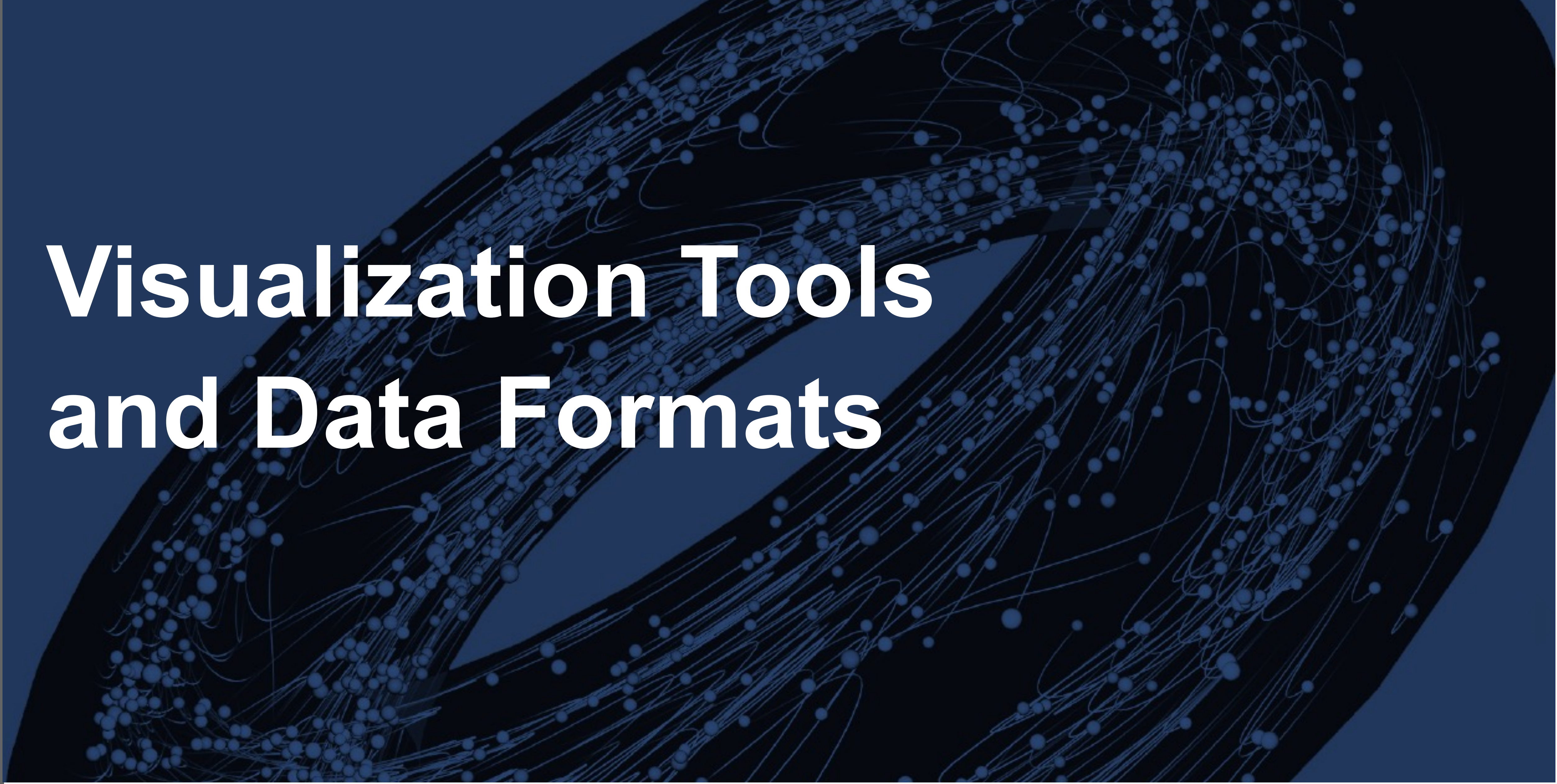
Data courtesy of: Jeff Greeley, Nichols Romero, Argonne National Laboratory

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

🟦 Available on Cooley

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)

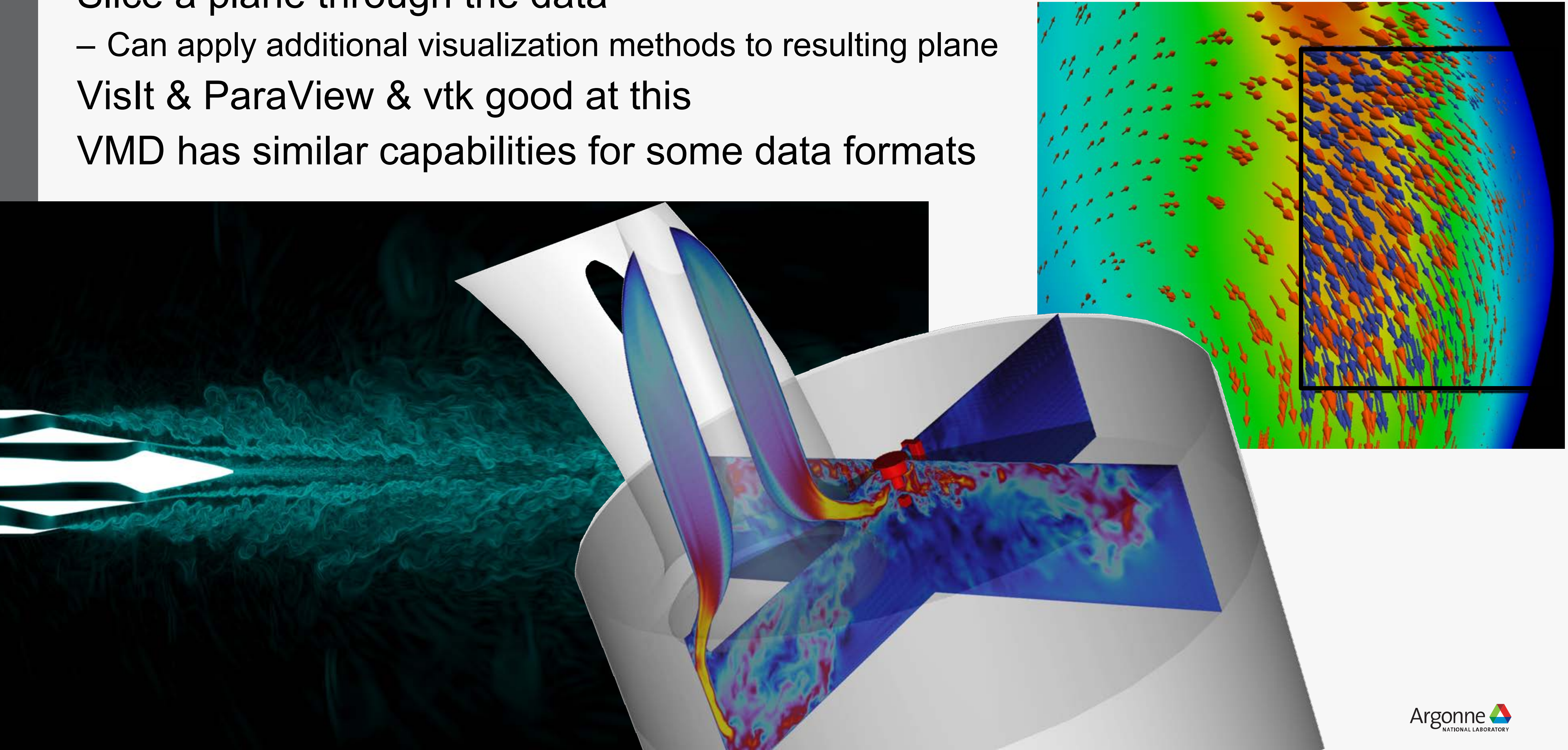
Data File Formats (ParaView & VisIt)

- 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

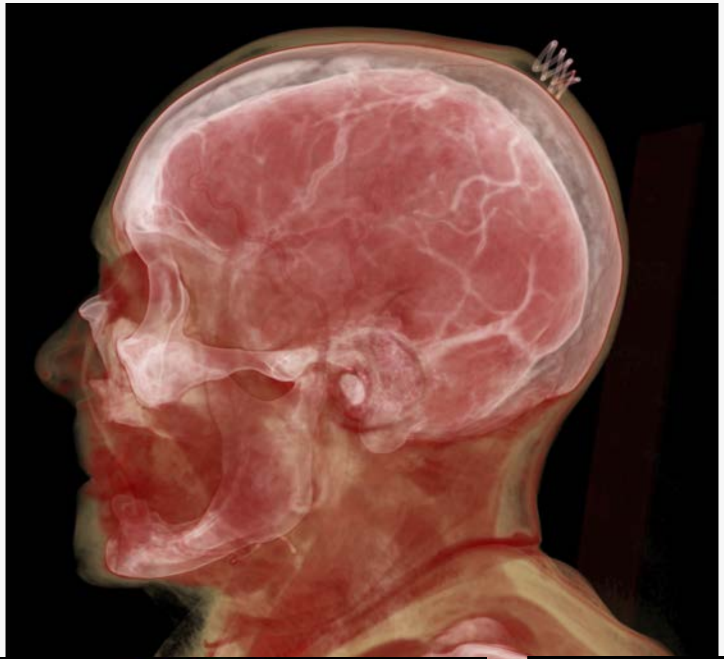
Data Representations

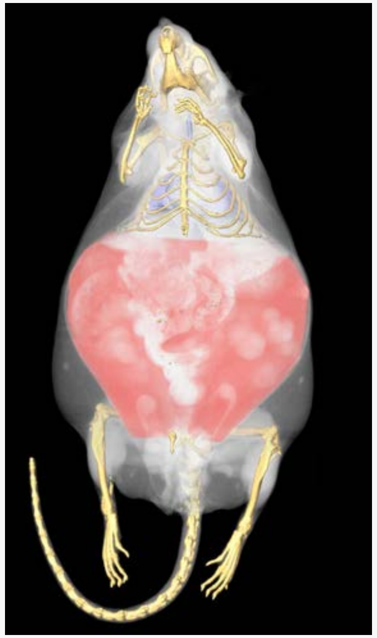
Data Representations: Cutting Planes

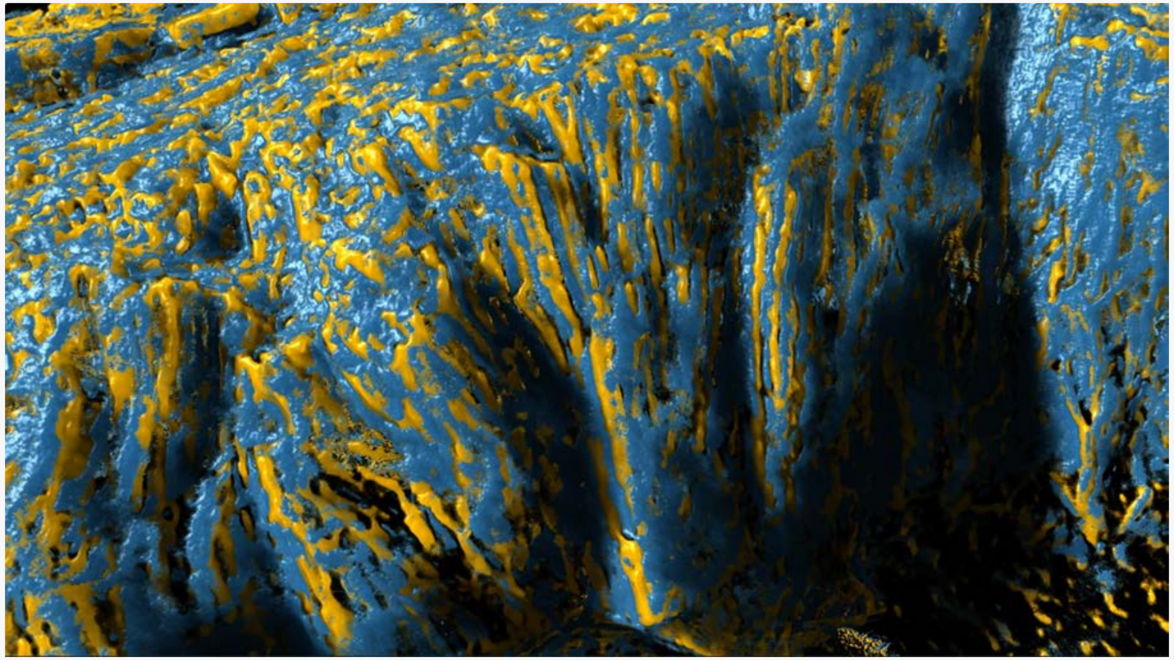
- Slice a plane through the data
  - Can apply additional visualization methods to resulting plane
- VisIt & ParaView & vtk good at this
- VMD has similar capabilities for some data formats

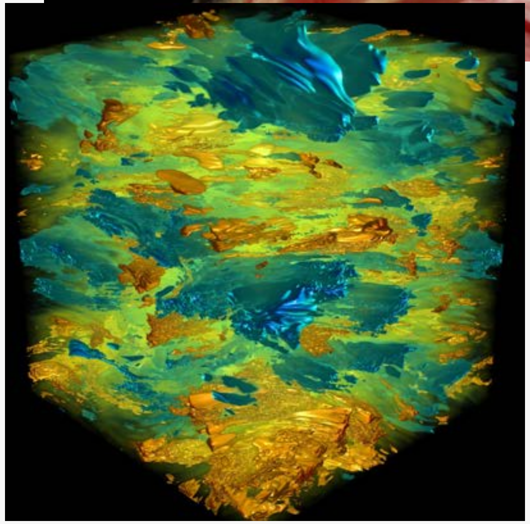


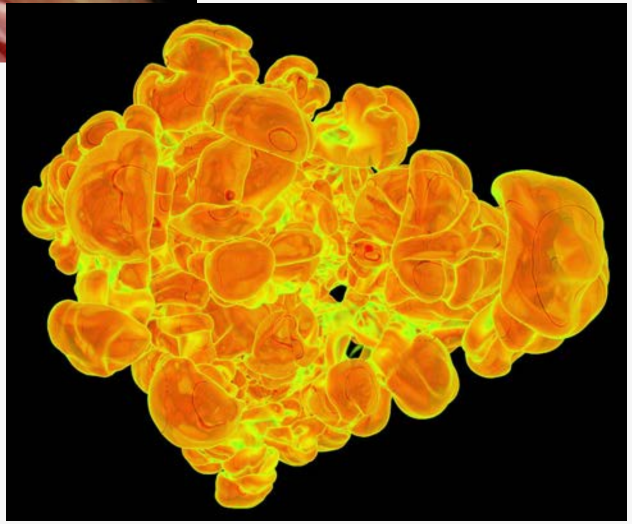
Data Representations: Volume Rendering

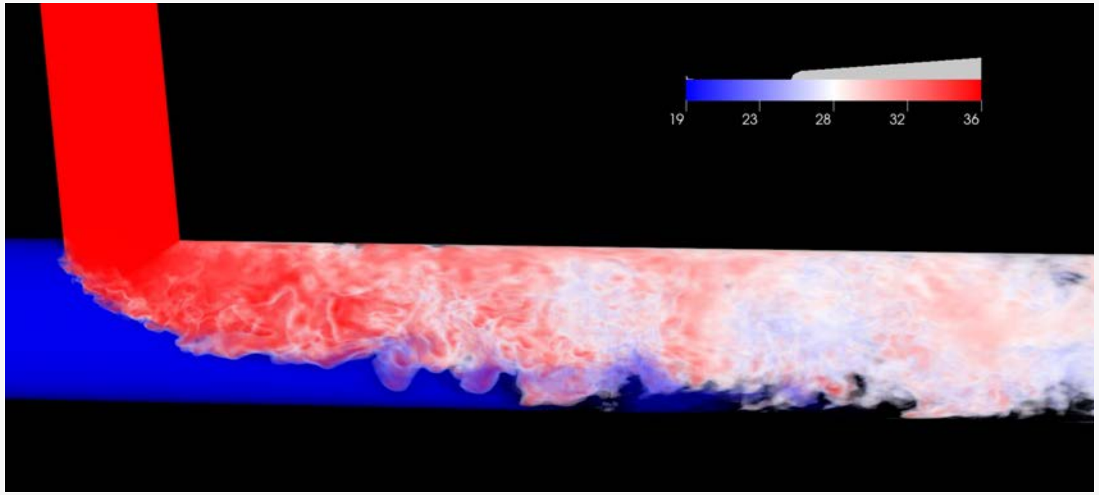












Data Representations: Contours (Isosurfaces)

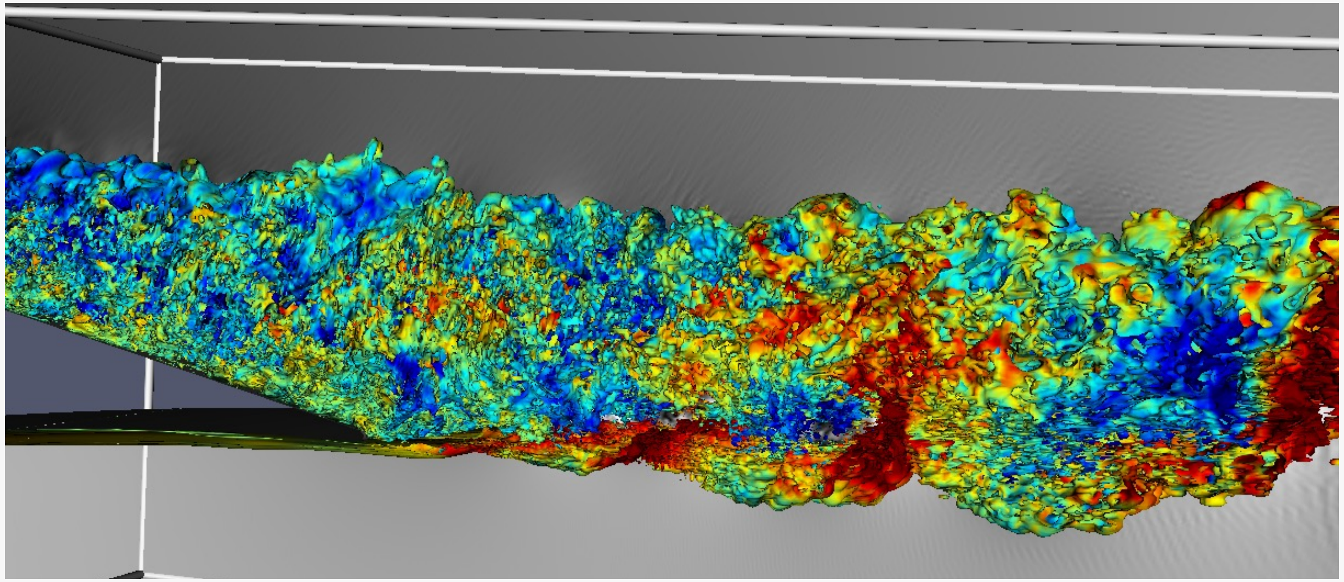
A Line (2D) or Surface (3D), representing a constant value

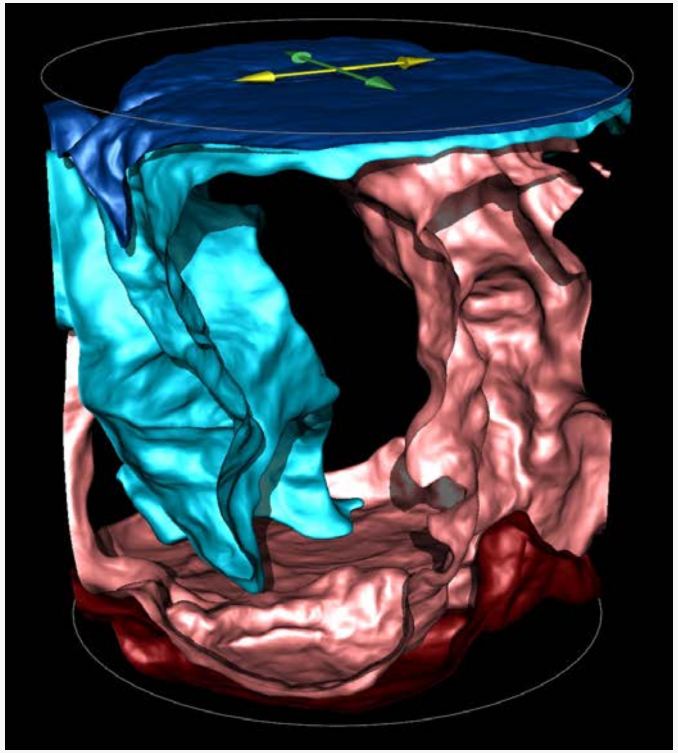
VisIt & ParaView:

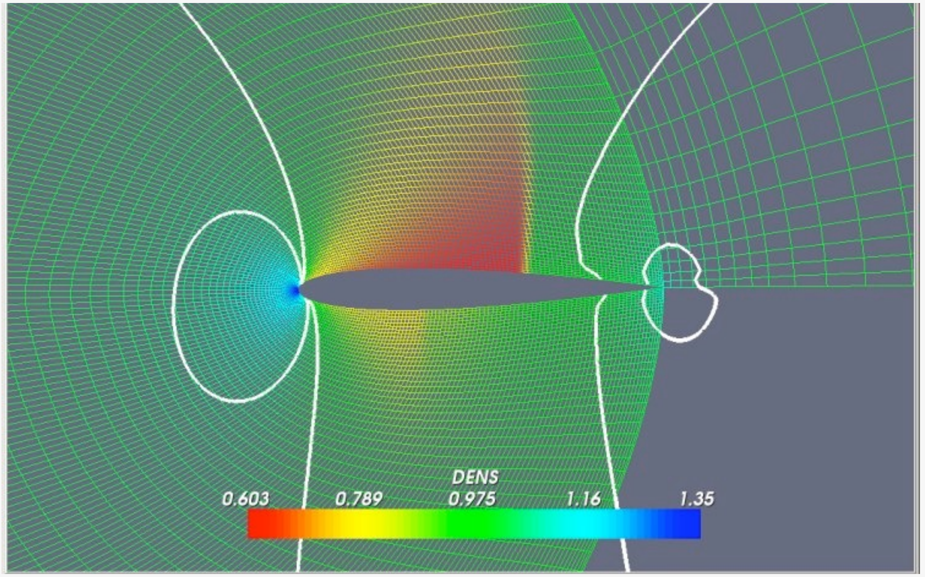
– good at this

vtk:

– same, but again requires more effort

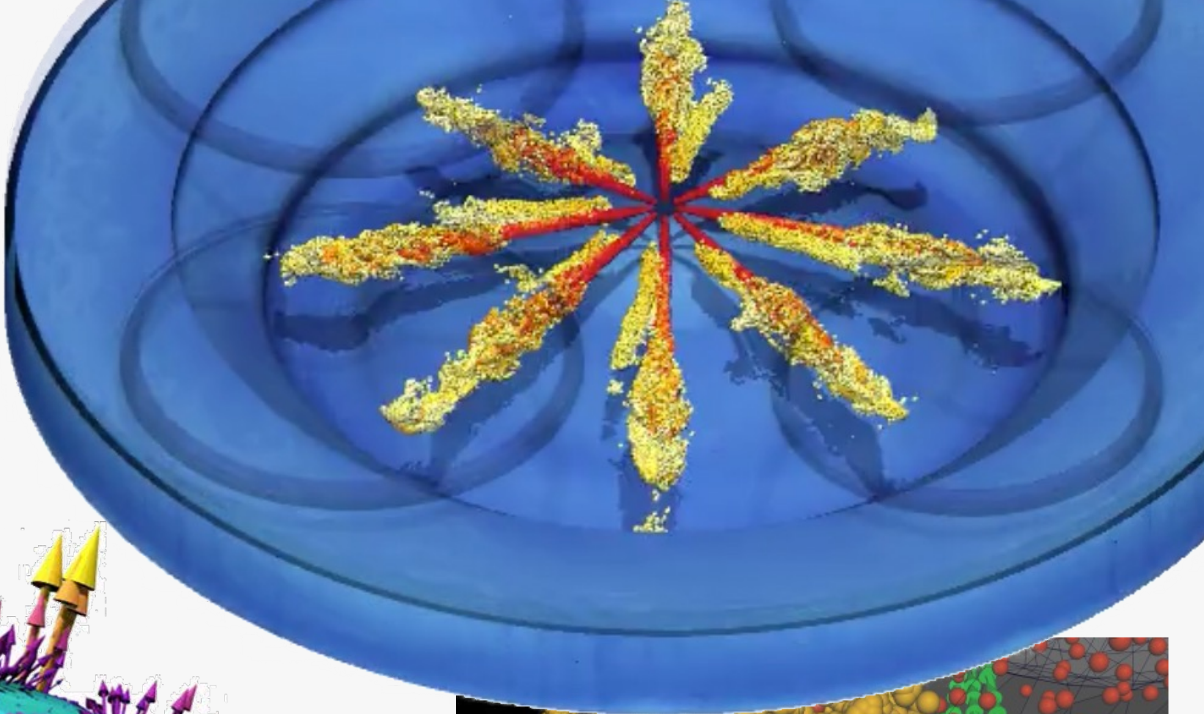


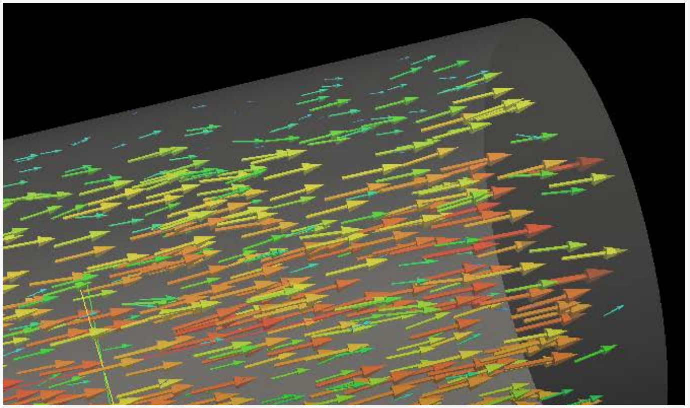


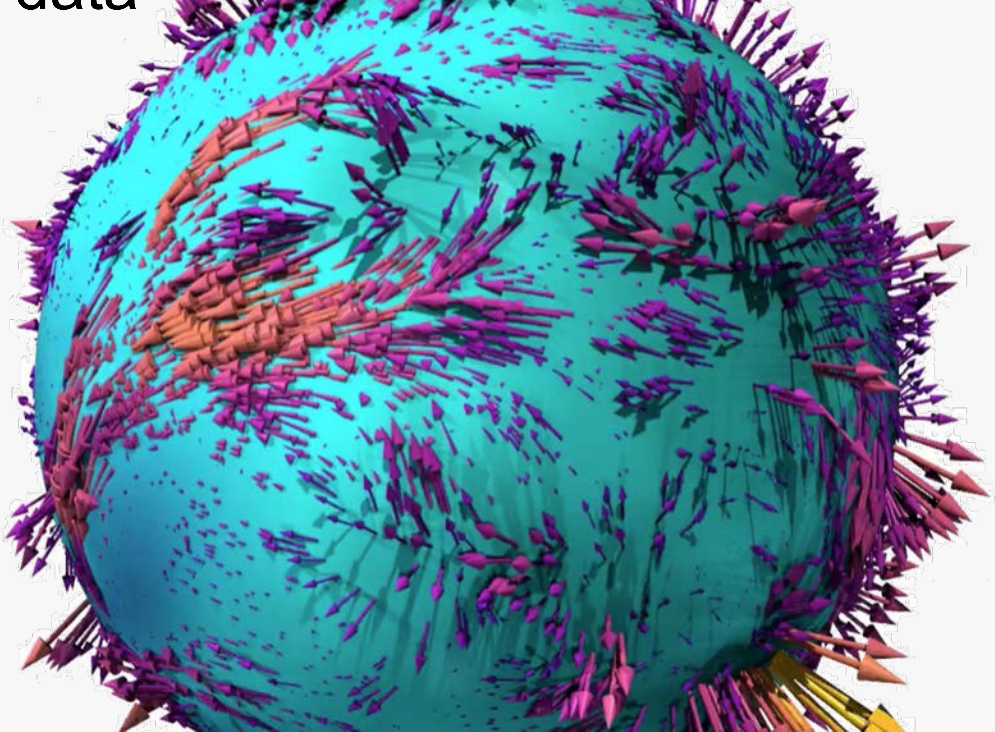


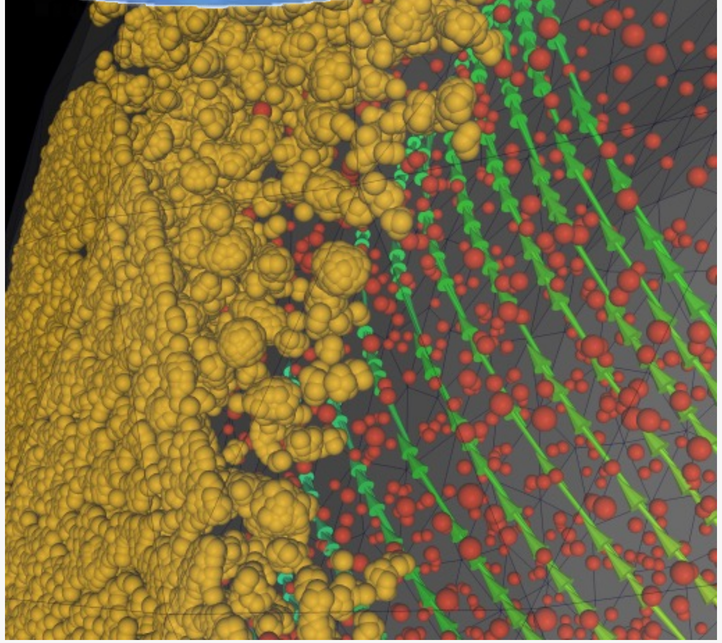
Data Representations: Glyphs

- 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



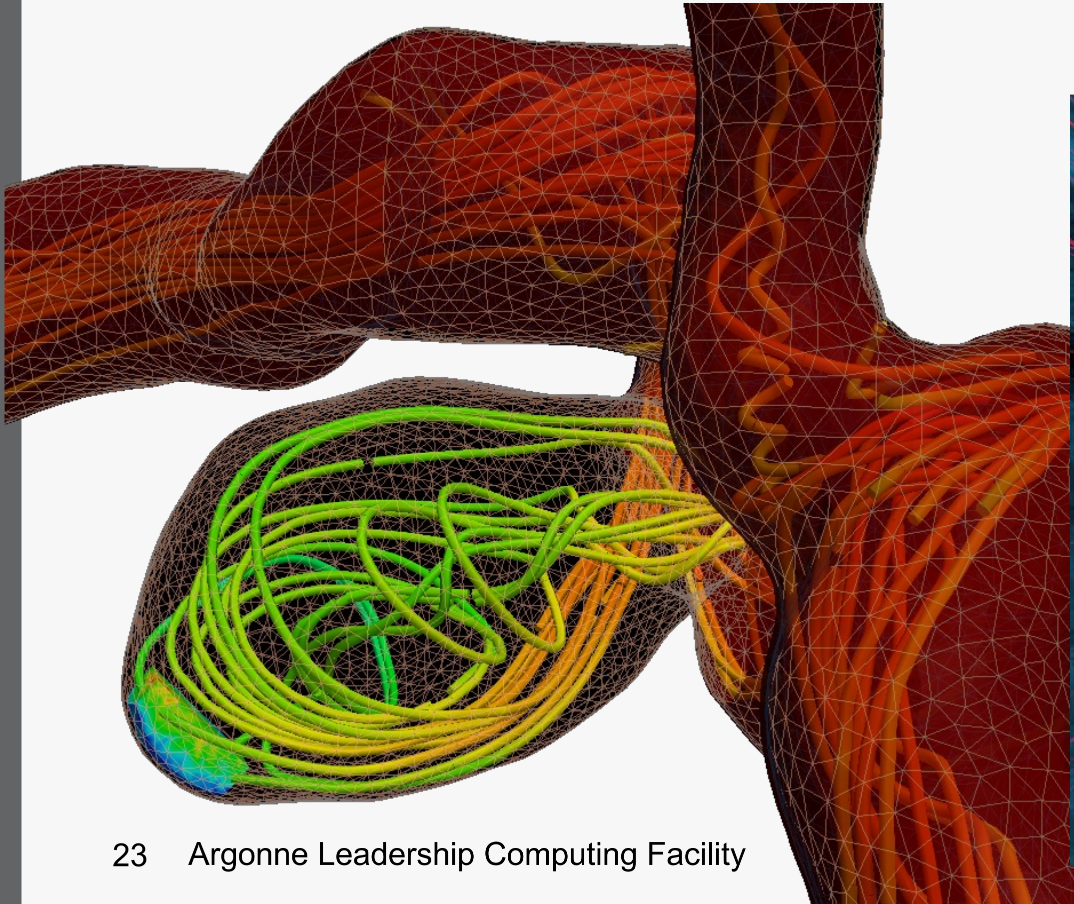


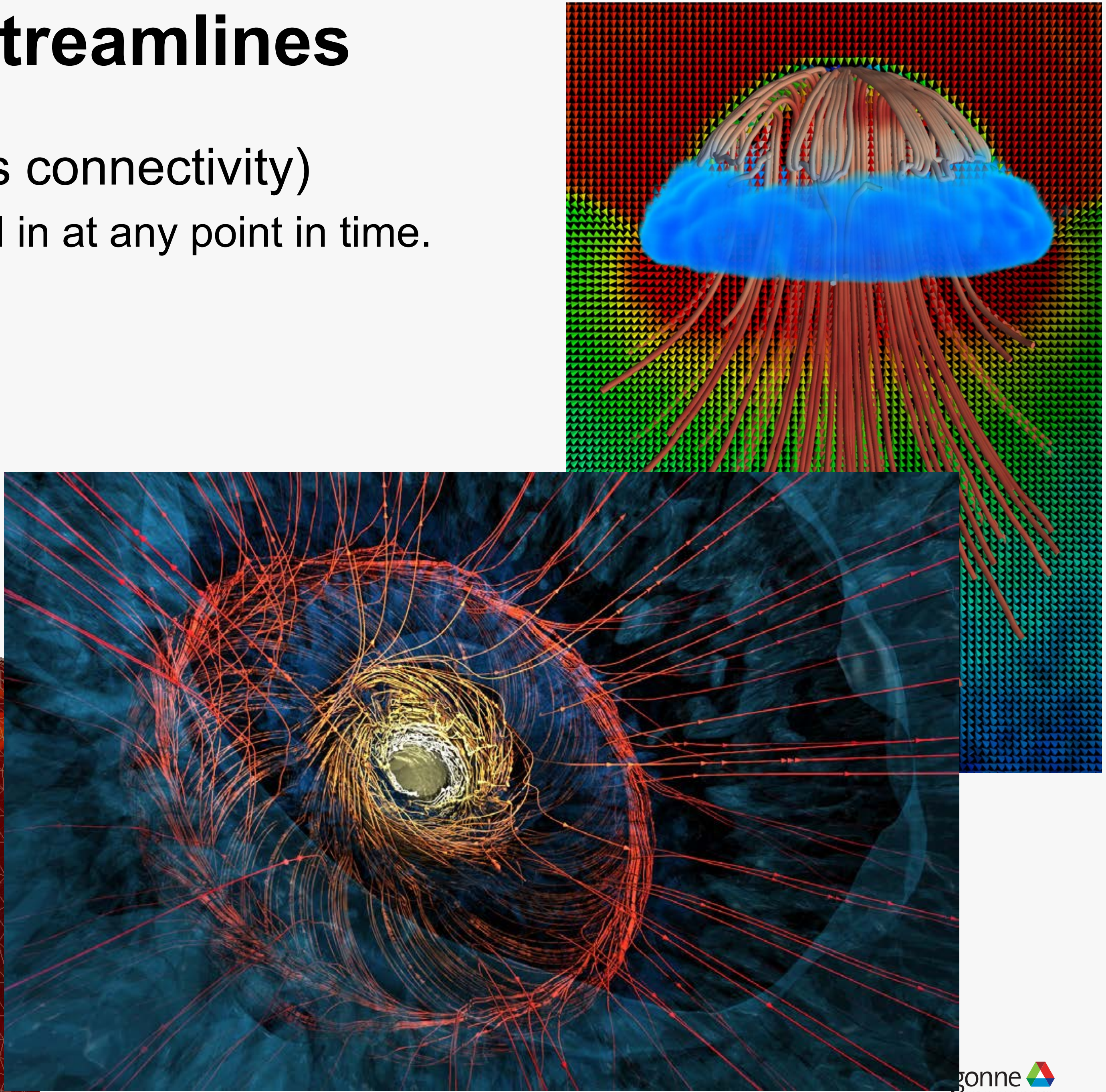




Data Representations: Streamlines

- 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:

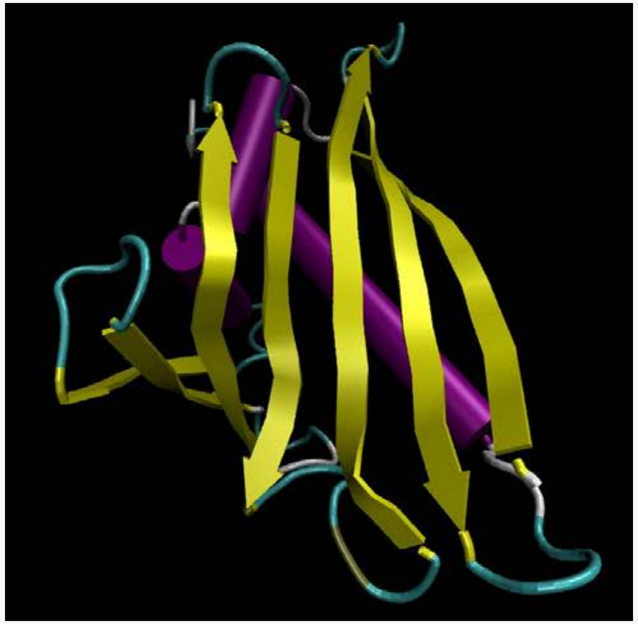
- Lots of domain-specific representations
- Many different file formats
- Animation
- Scriptable

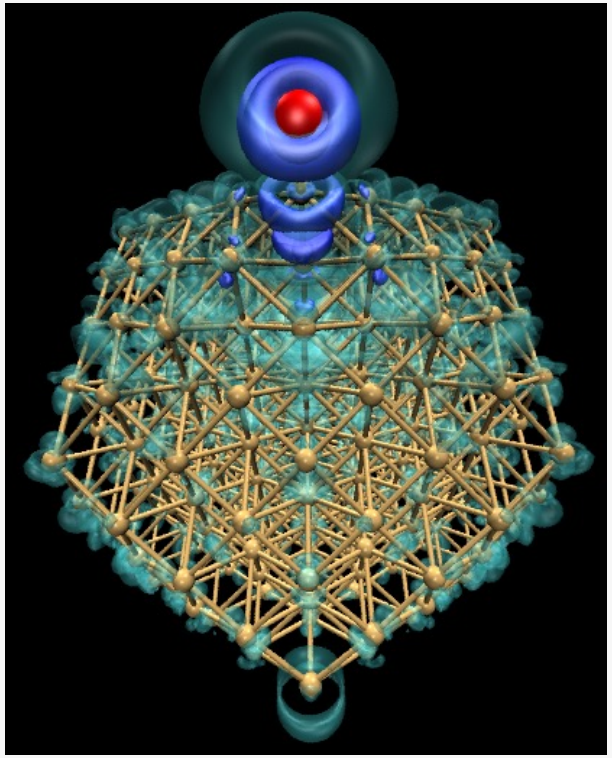
VisIt & ParaView:

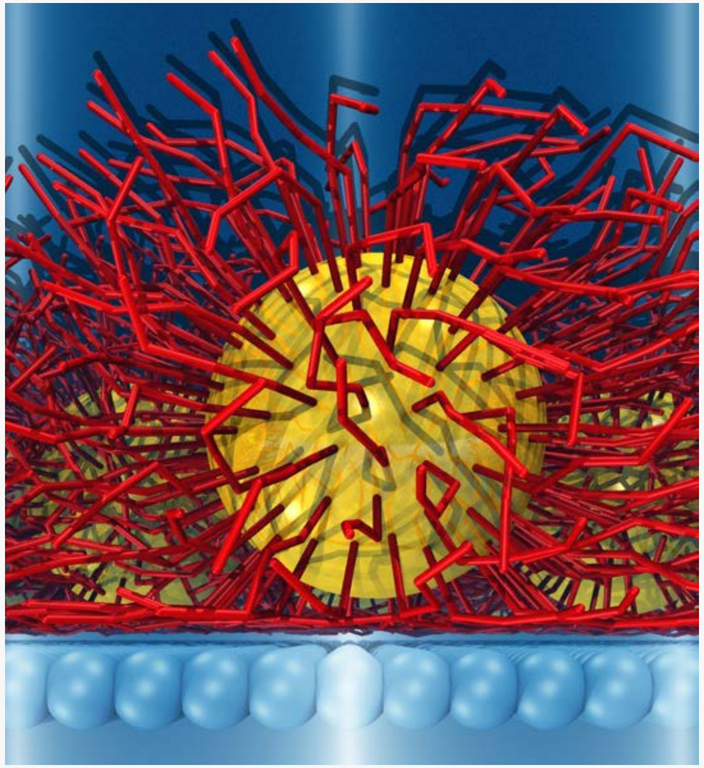
- Limited support for these types of representations, but improving

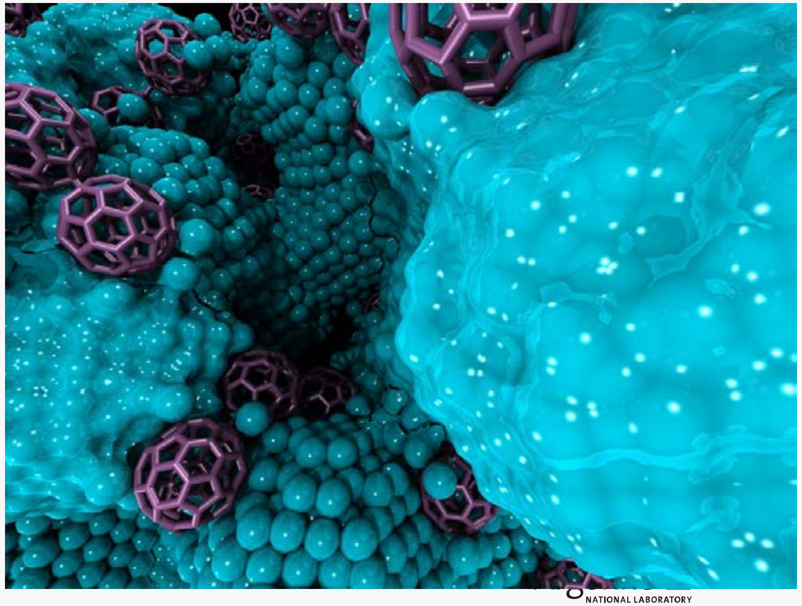
VTK:

- Anything’s possible if you try hard enough



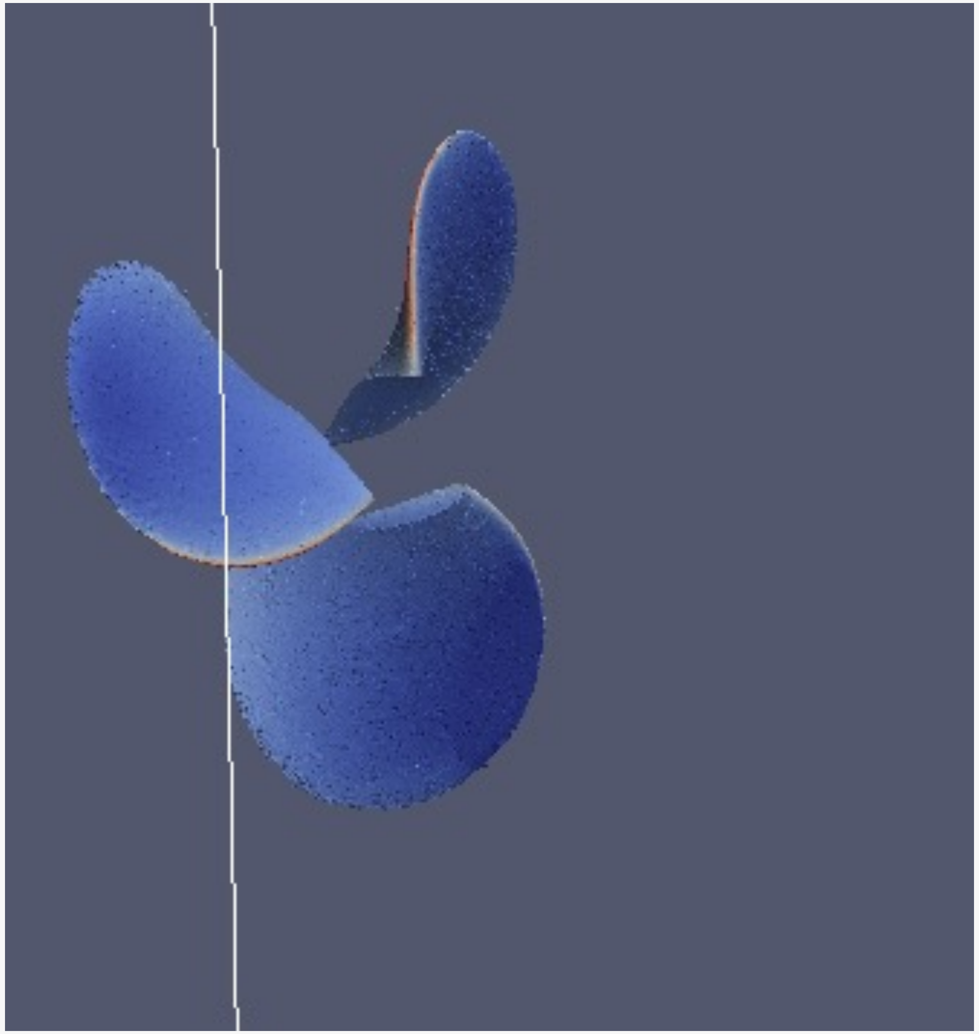




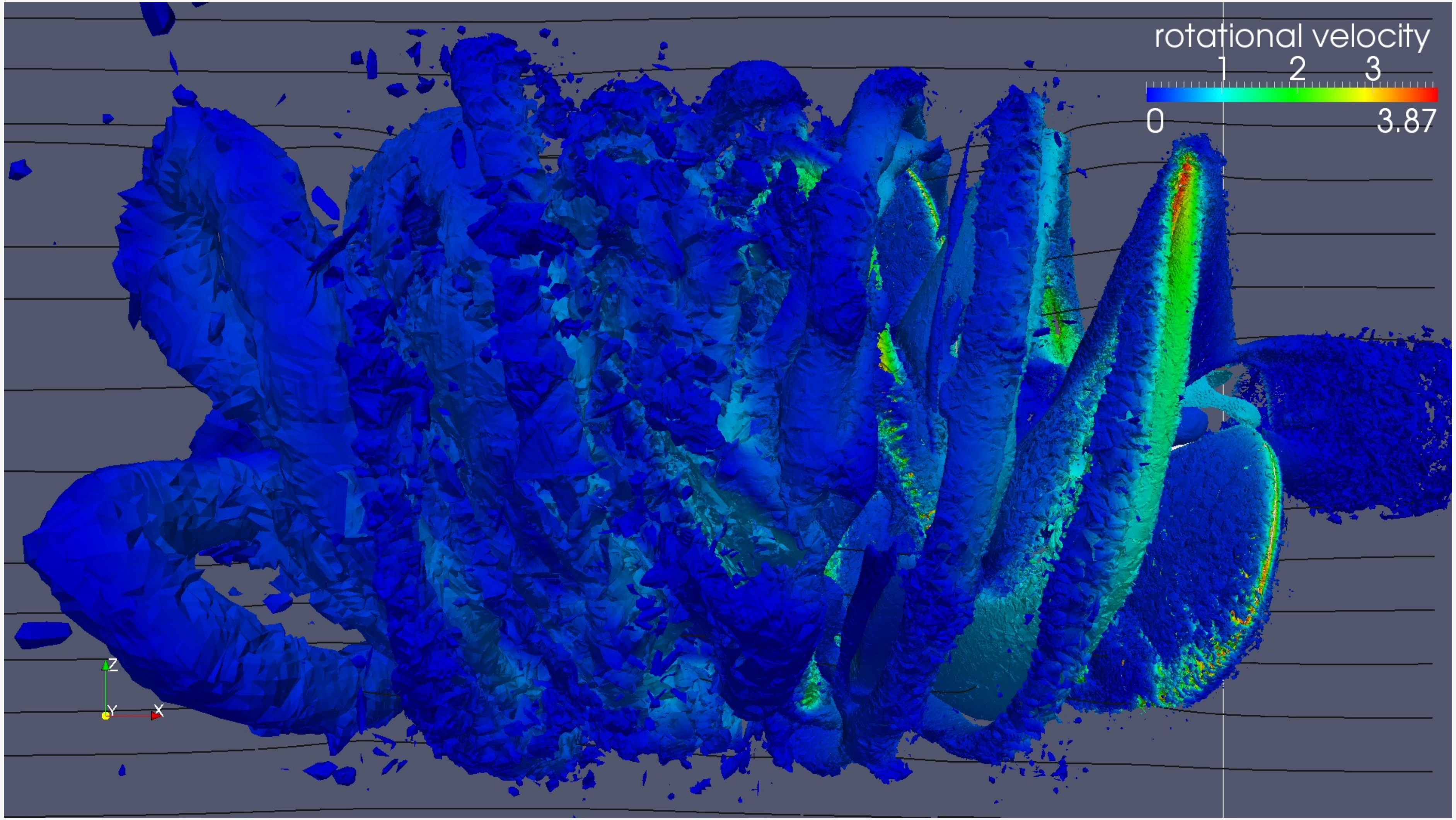


Visualization for Debugging

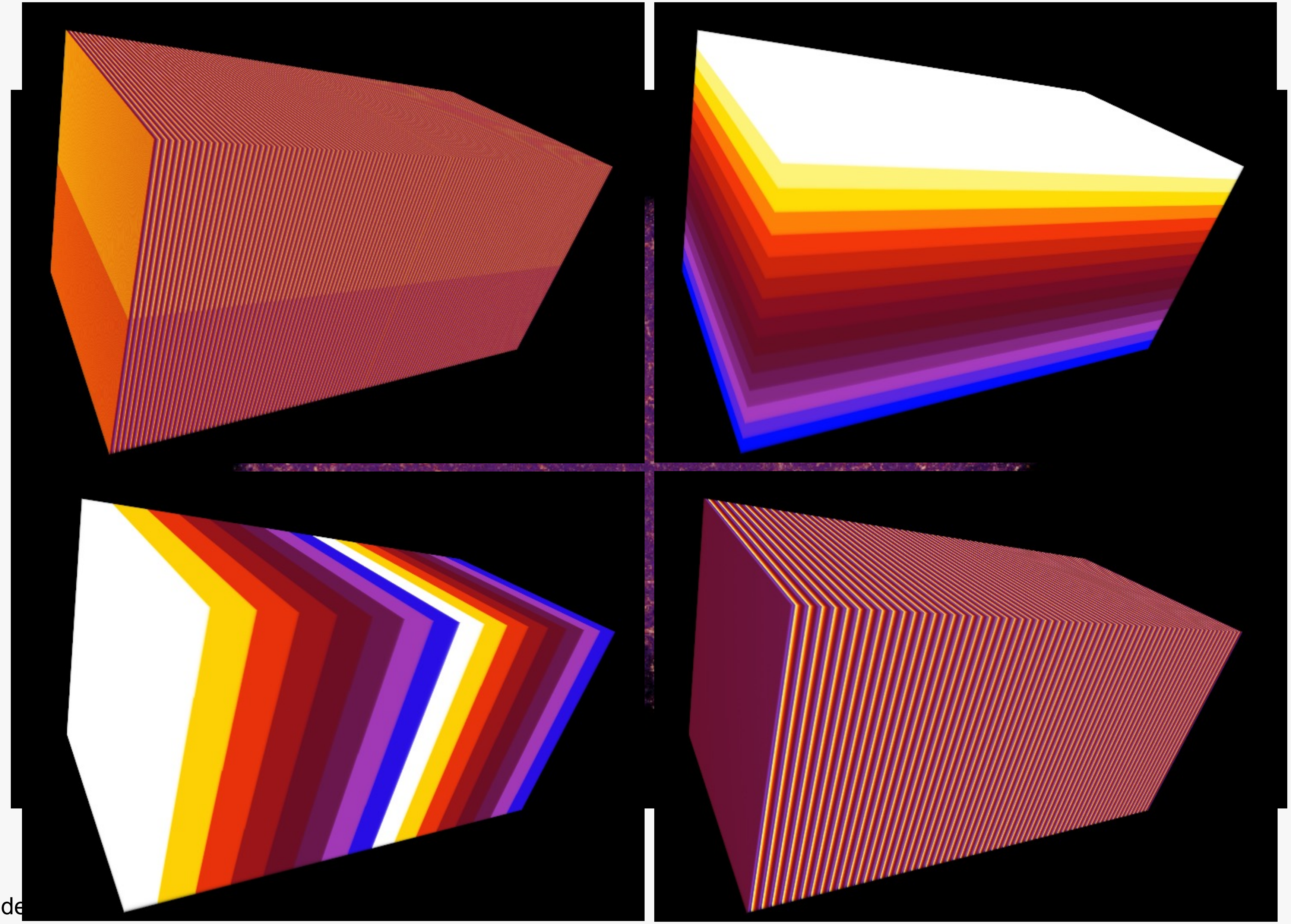
Visualization for Debugging



Visualization for Debugging

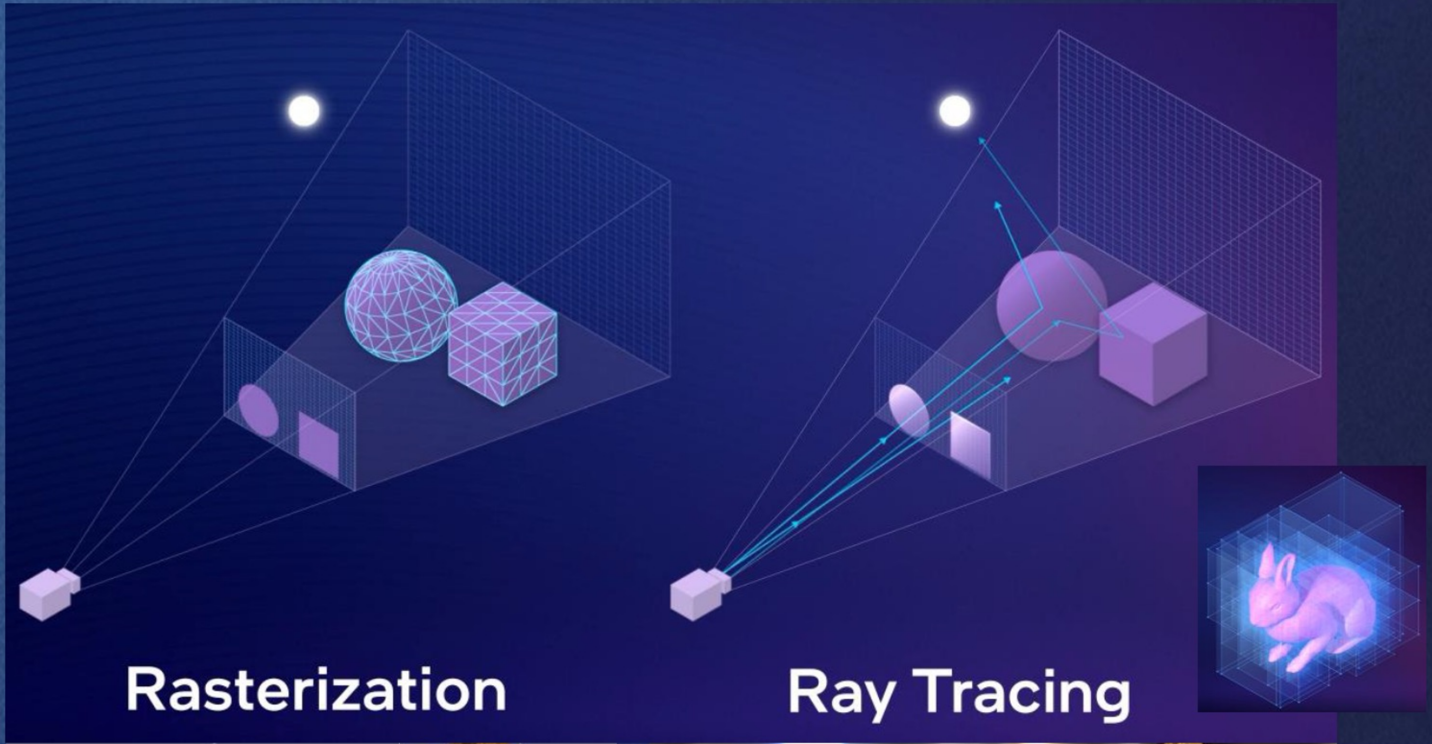


Visualization as Diagnostics: Color by Thread ID

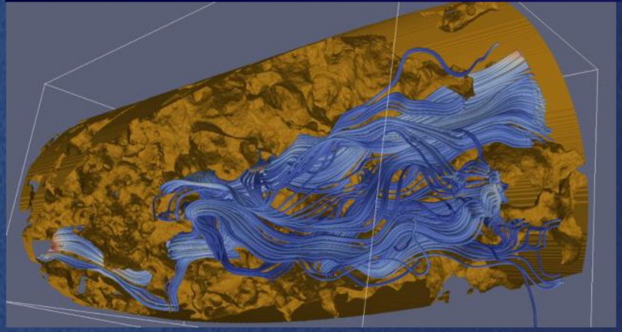


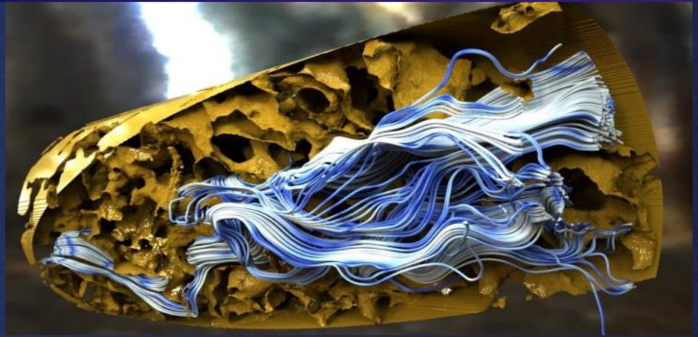
Advanced Rendering

Rendering



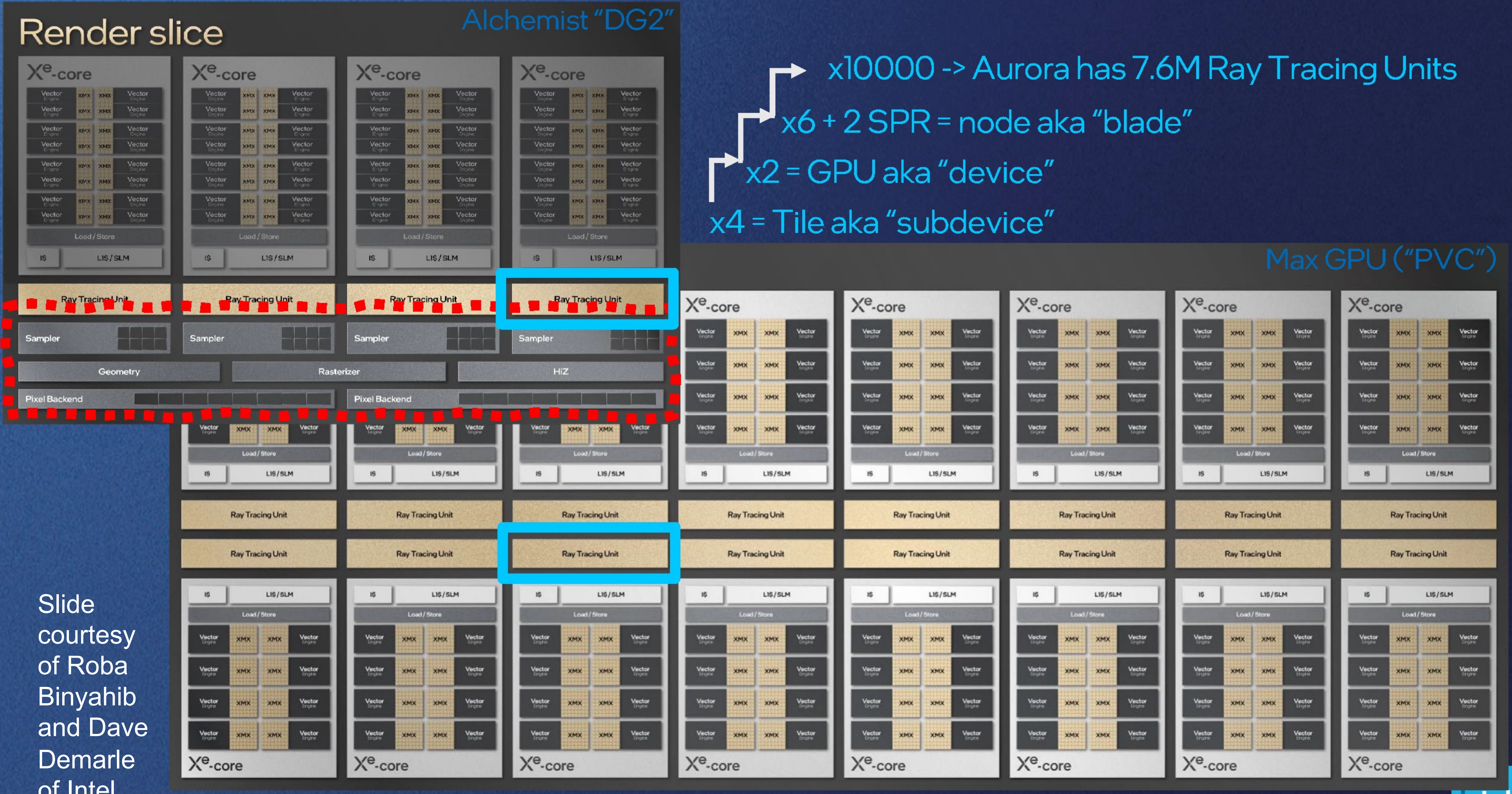
BVH sorts space to accelerate Ray Tracing





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



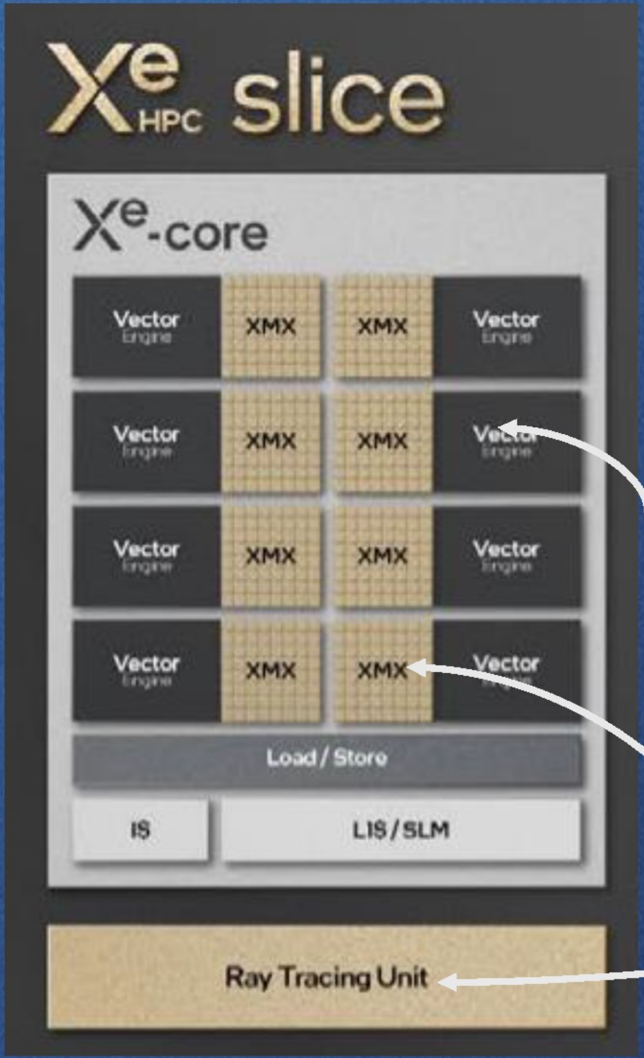
x10000 -> Aurora has 7.6M Ray Tracing Units

x6 + 2 SPR = node aka "blade"

x2 = GPU aka "device"

x4 = Tile aka "subdevice"

Slide courtesy of Roba Binyahib and Dave Demarle of Intel



ONEAPI RENDERING TOOLKIT

Application/InSitu library

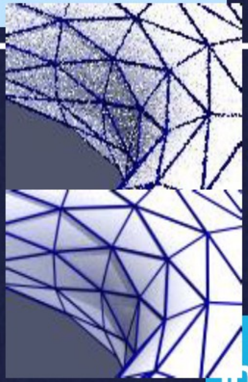
ParaView/Catalyst, VisIt/libSim, /SENSEI, OSPRay Studio/

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

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

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

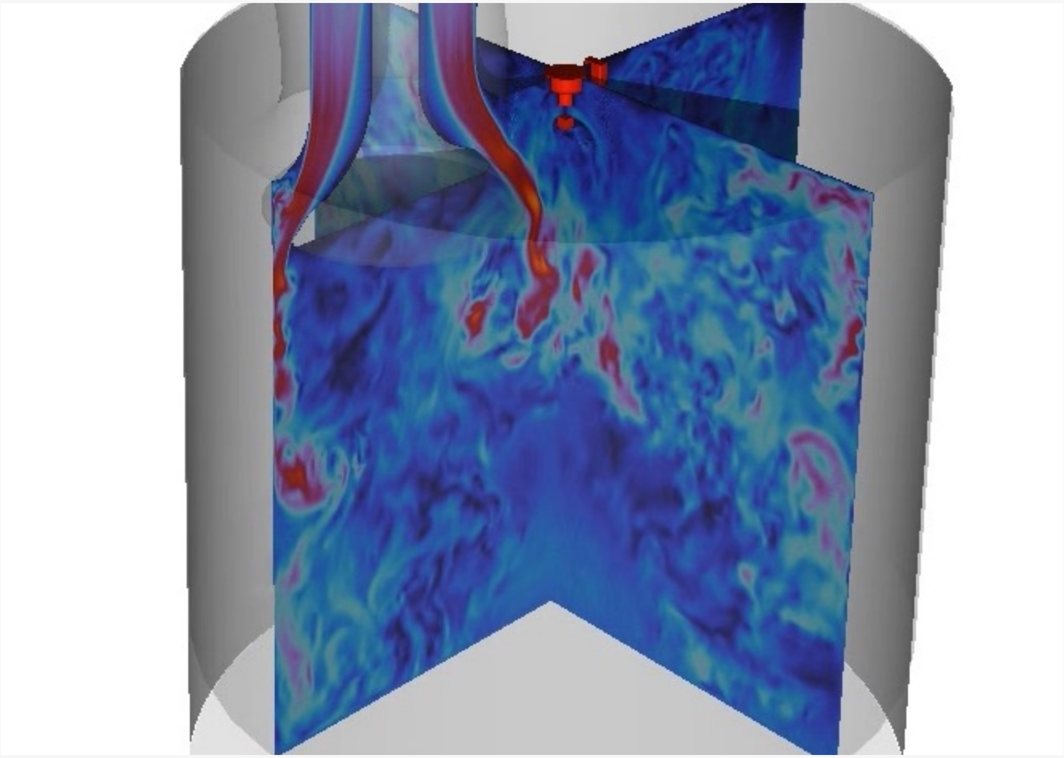


Slide courtesy of Roba Binyahib and Dave Demarle of Intel

The Science

Internal Combustion Engine Simulation

TCC Engine Apparatus



Fluid Dynamics Simulation

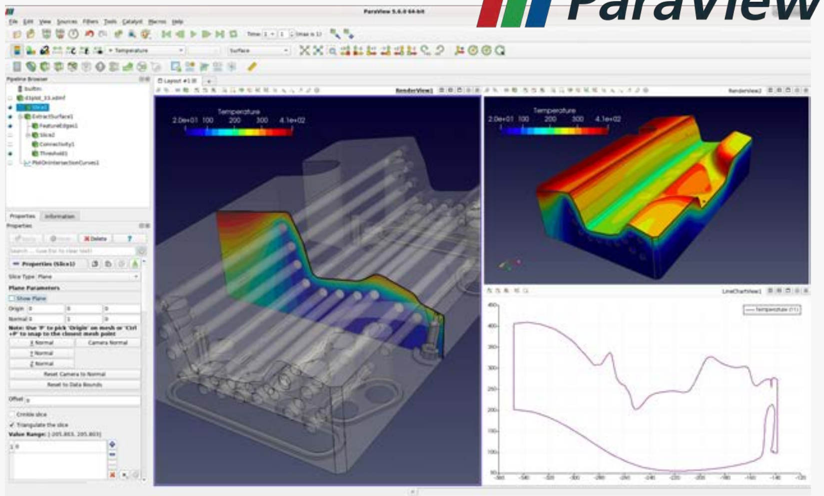
Goal

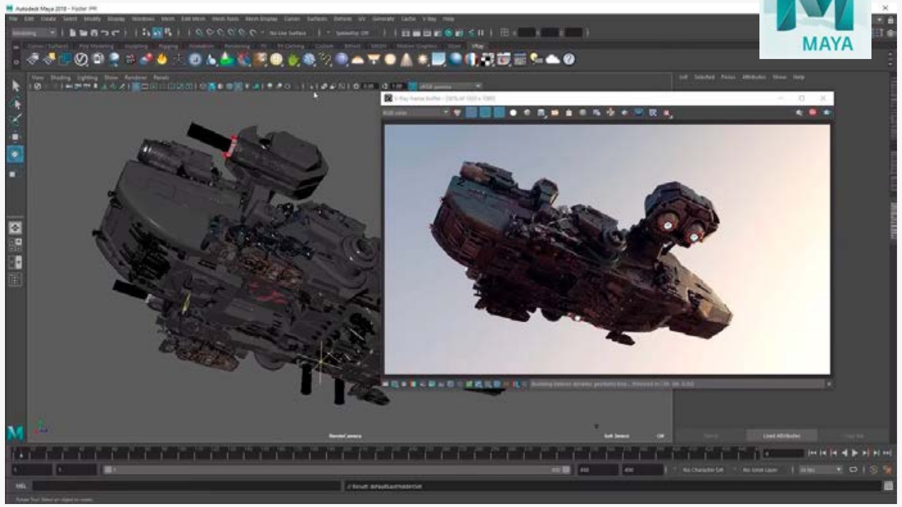
Provide context to tell the story/explain the science

Integrate production tools into the existing visualization pipeline

Tools used:

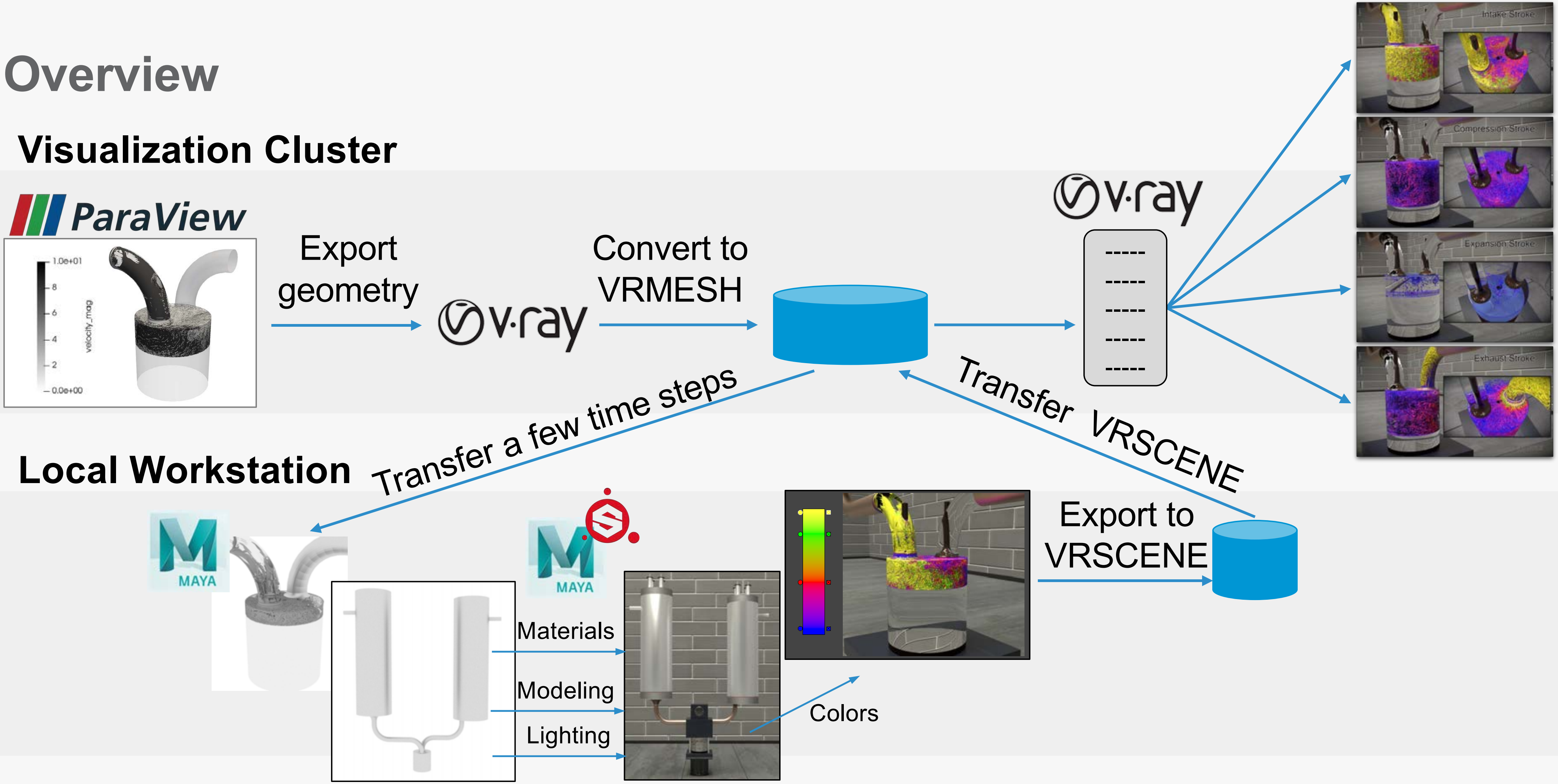
- ParaView
- Maya
- Substance Painter
- V-Ray
- Custom scripts and HPC Resources
- ffmpeg
- Premiere/After Effects





THE VISUALIZATION PIPELINE

Overview



Visualization Cluster

Export geometry

Convert to VRMESH

Transfer a few time steps

Transfer VRSCENE

Local Workstation

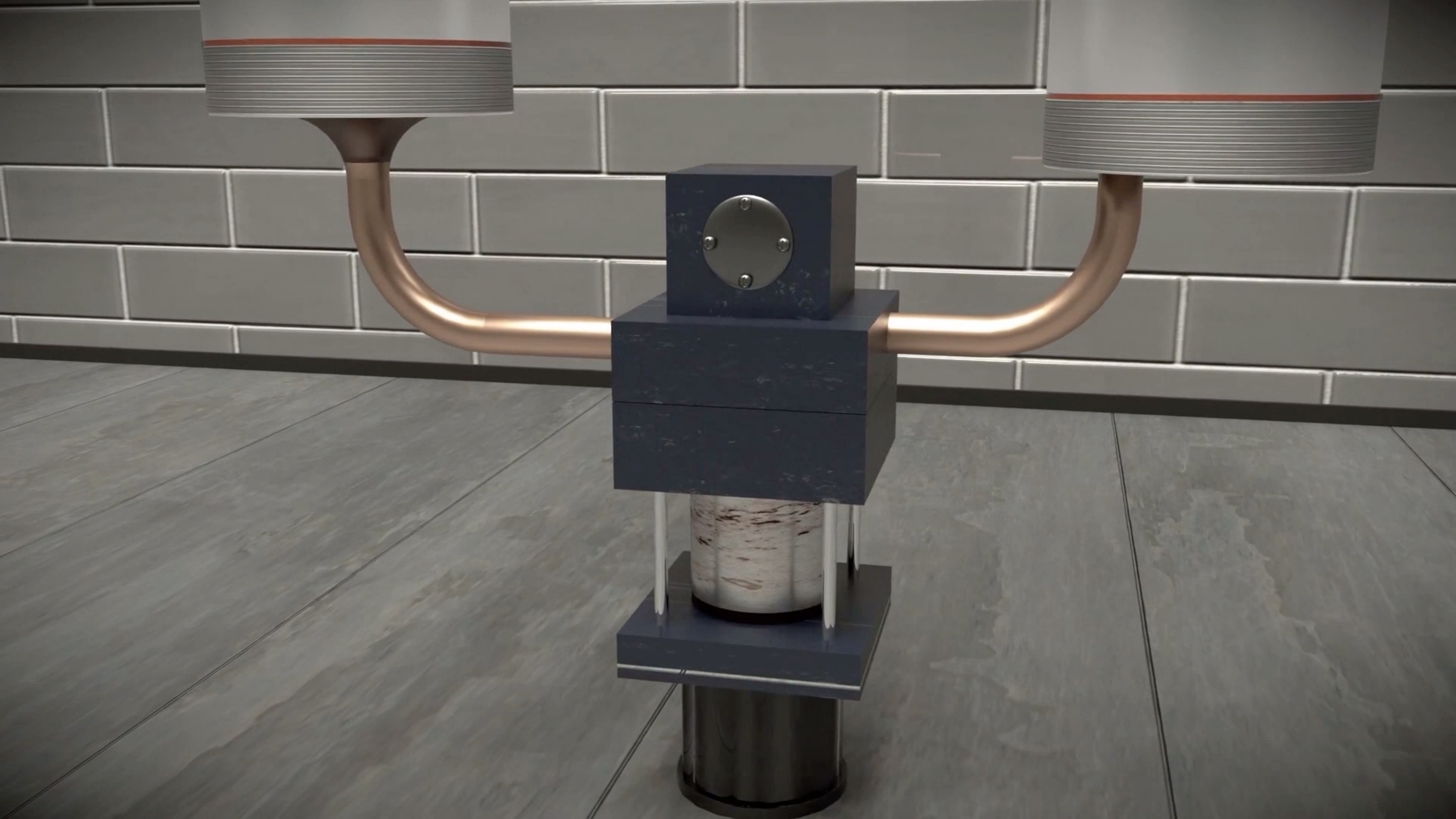
Materials

Modeling

Lighting

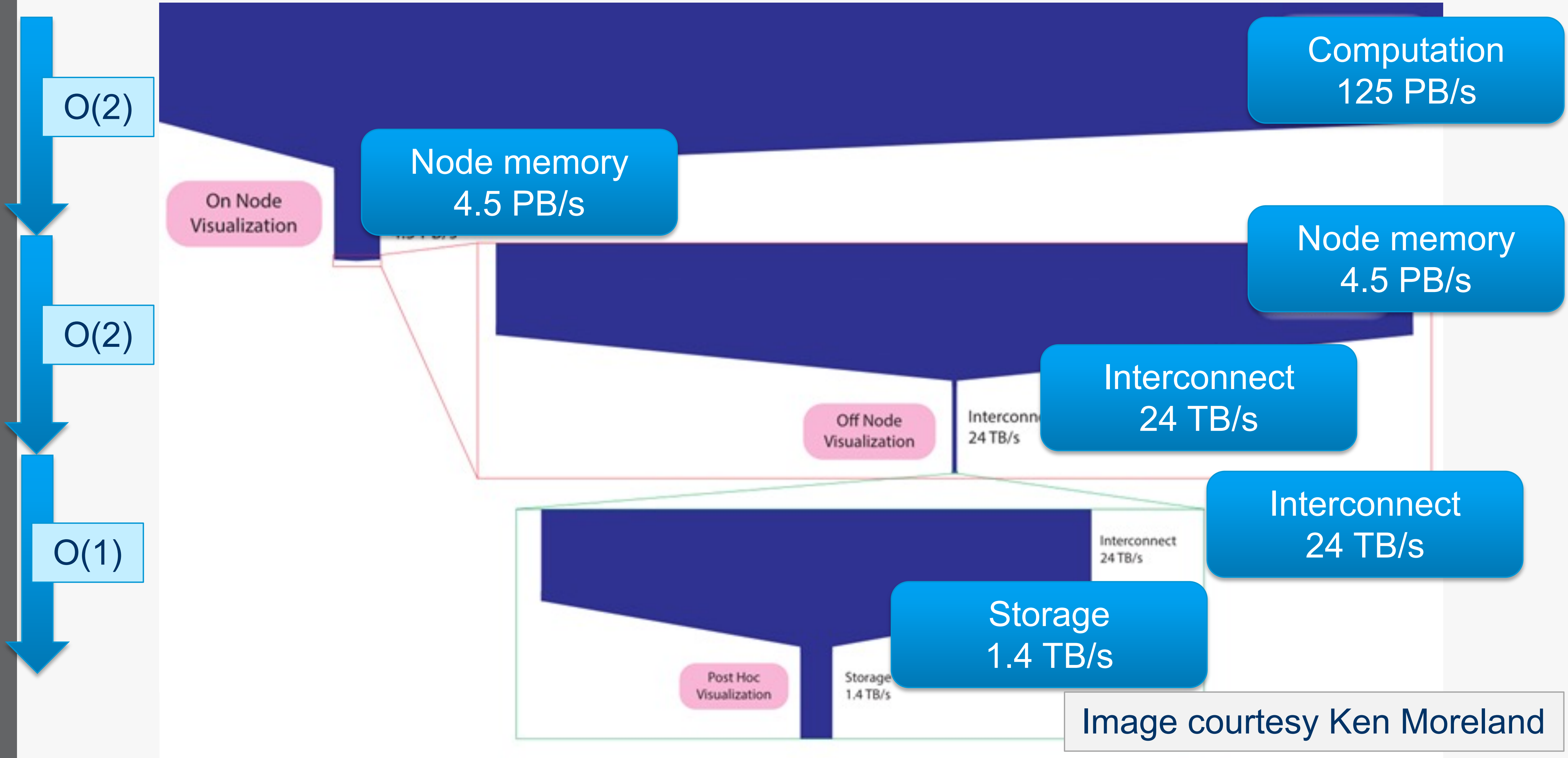
Colors

Export to VRSCENE



In Situ Visualization and Analysis

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



O(2)

O(2)

O(1)

Computation
125 PB/s

Node memory
4.5 PB/s

Node memory
4.5 PB/s

Interconnect
24 TB/s

Interconnect
24 TB/s

Storage
1.4 TB/s

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*

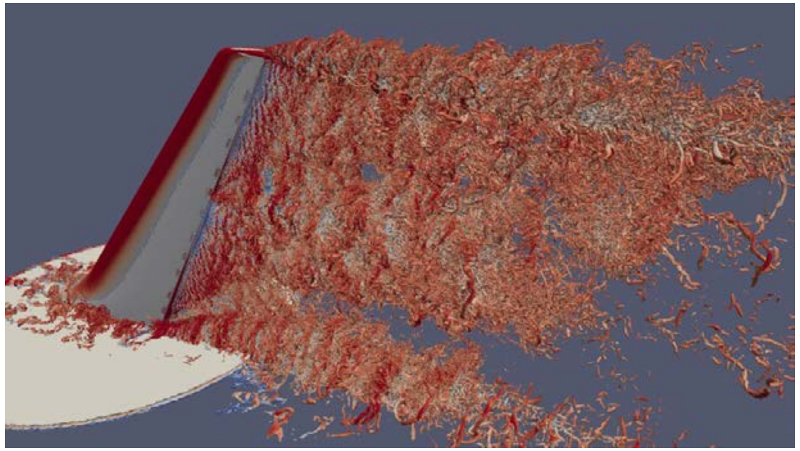
Traditionally visualization and analysis happens post hoc

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

In situ

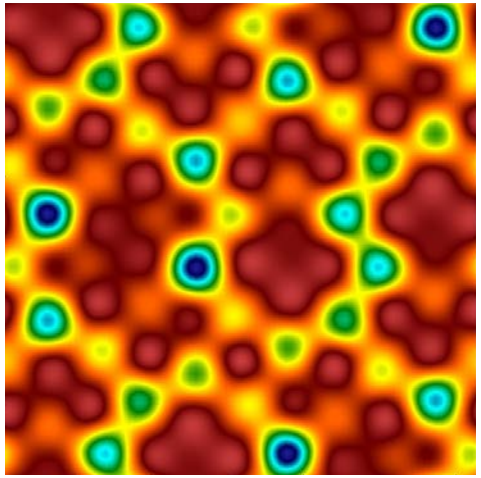
- Data gets visualized/analyzed **while** in memory.
- If zero-copy used, there is no data movement
- Ideally the data is on the GPU and stays on the GPU

In Situ



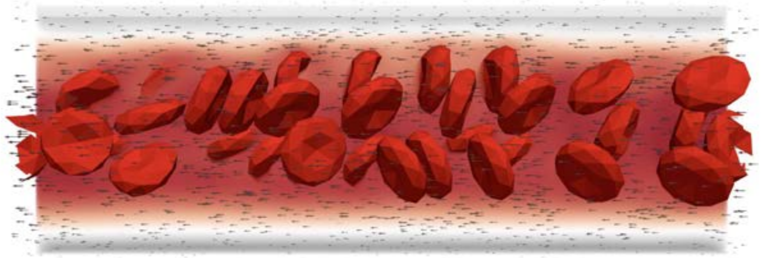
~2014

PHASTA, Catalyst, Ken Jansen



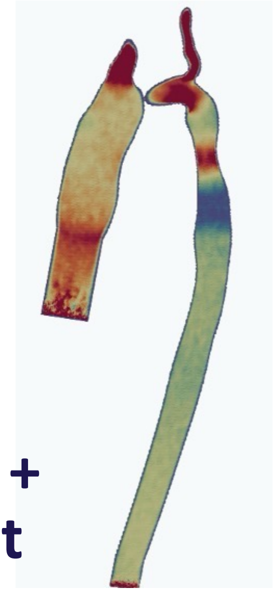
2018

Nek5000, SENSEI



2021 - 2024

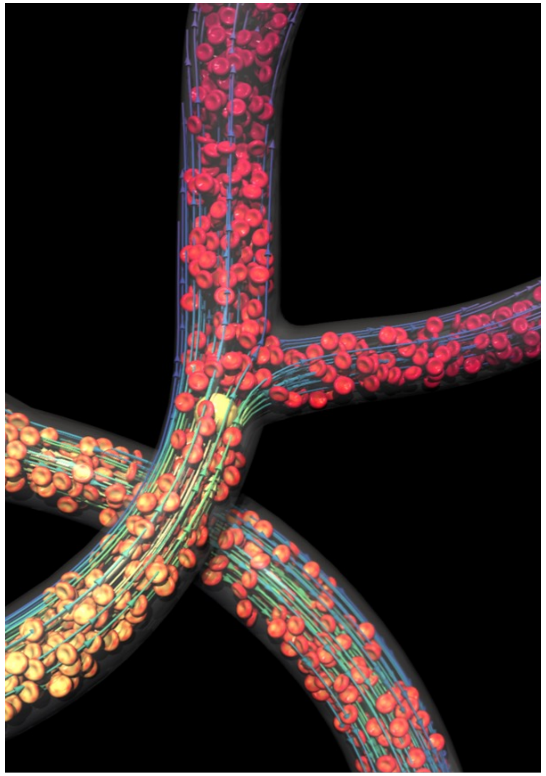
Palabos+LAMMPS, SENSEI + Catalyst, bi-directional



2019

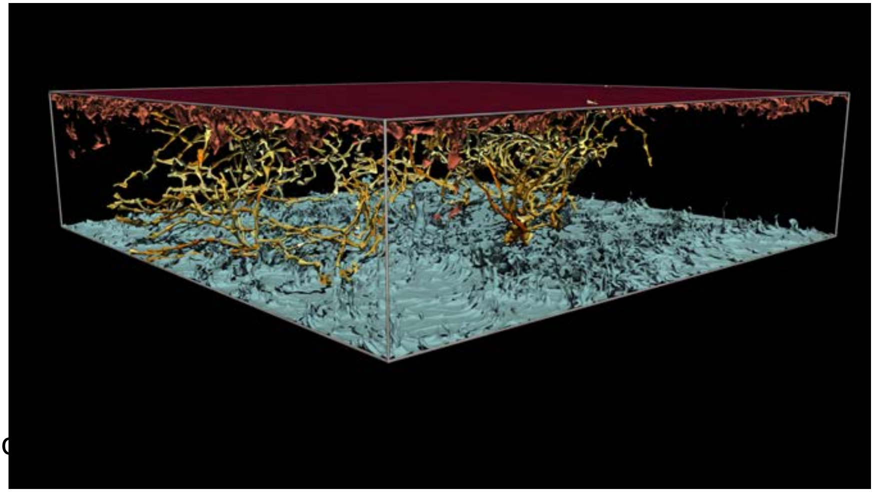
SENSEI + Catalyst

HARVEY



Ascent + Catalyst

2024



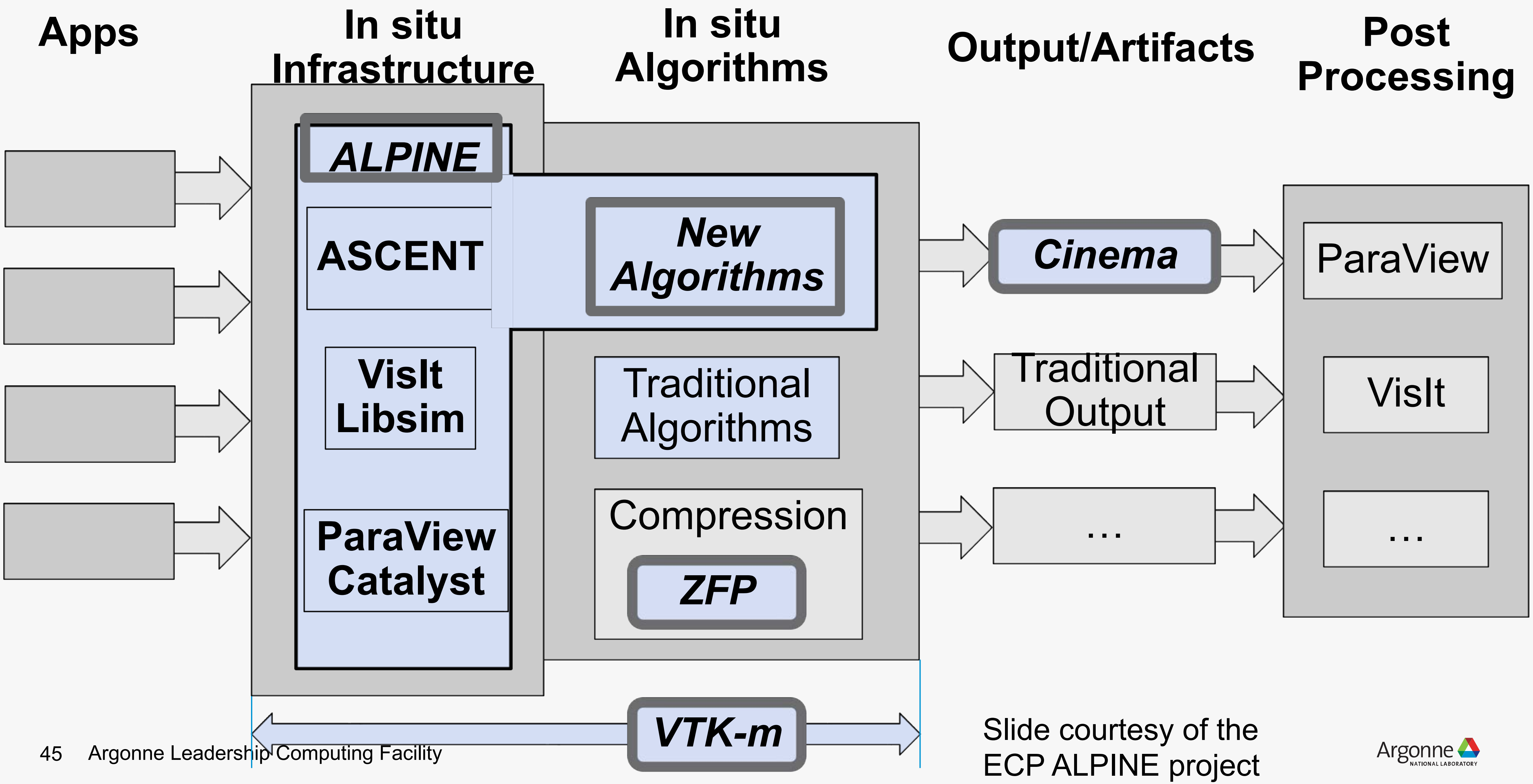
2024

nekRS, Ascent + Catalyst

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
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
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
SmartSim	SmartSim is a software framework that facilitates the convergence of numerical simulations and AI workloads on heterogeneous architectures	Joe Insley / Silvio Rizzi

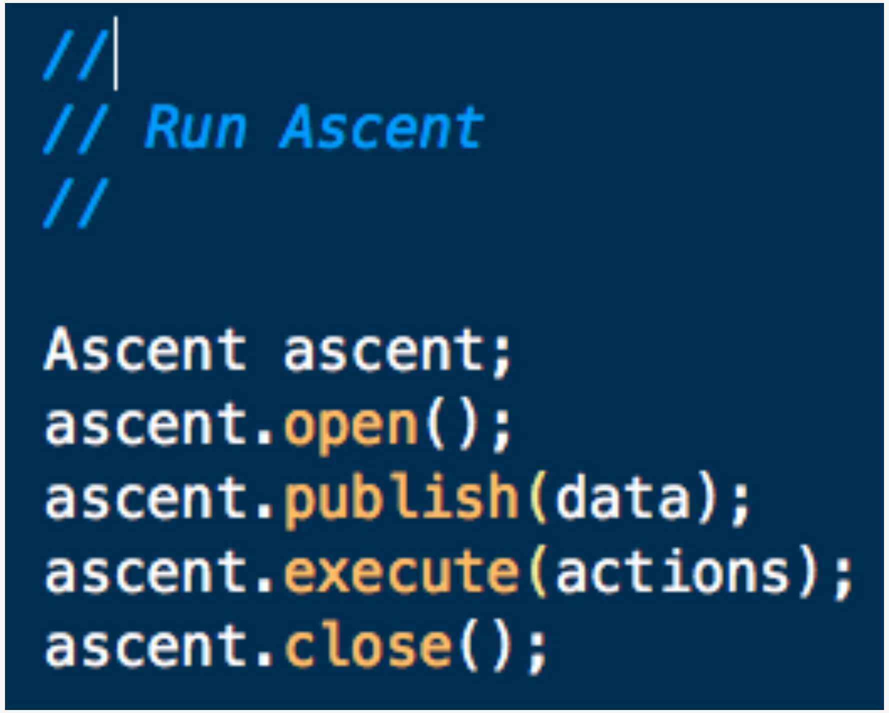
Exascale Computing Project Software Technology Data and Visualization



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

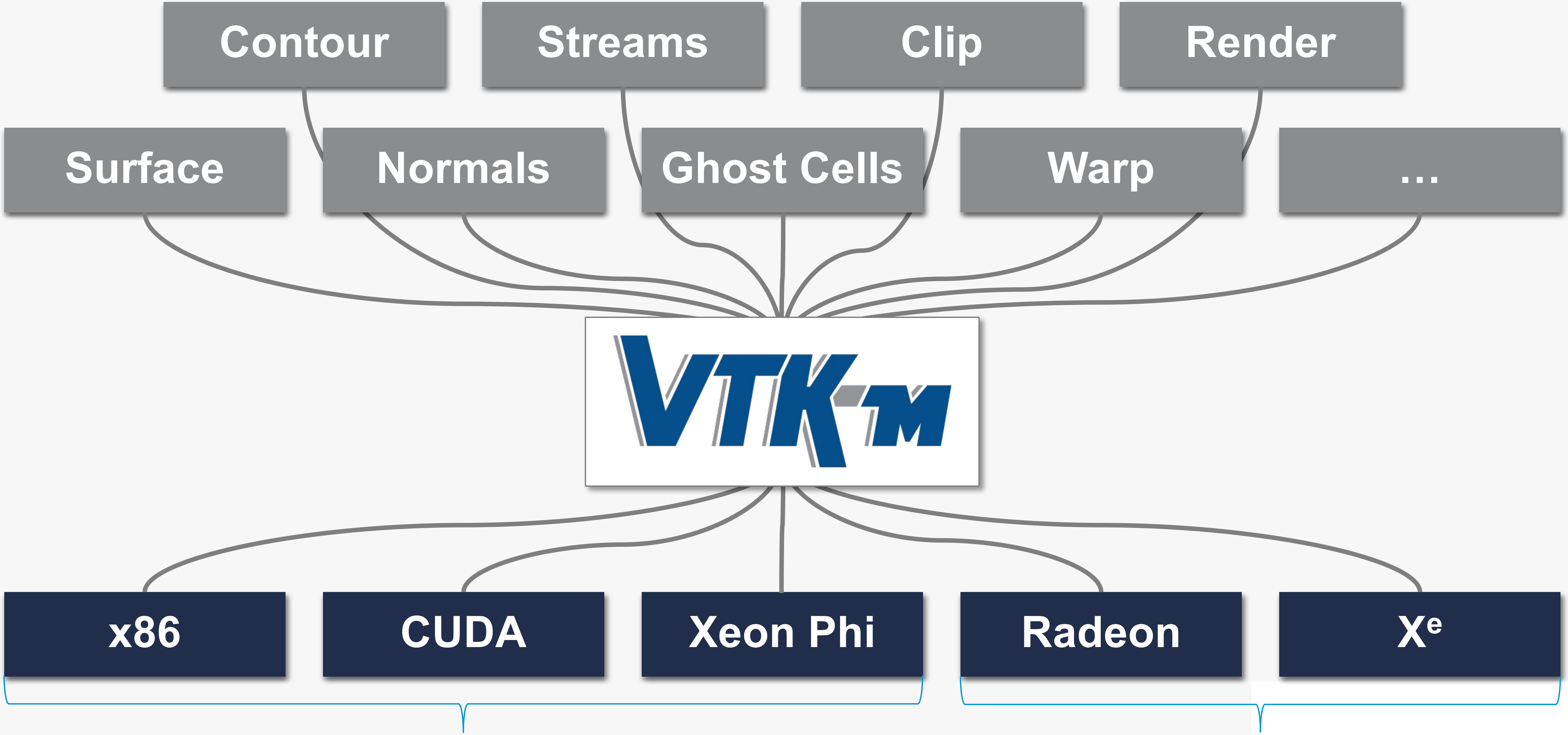


```
//
// Run Ascent
//

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

Slide courtesy of the Ascent team

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

X^e

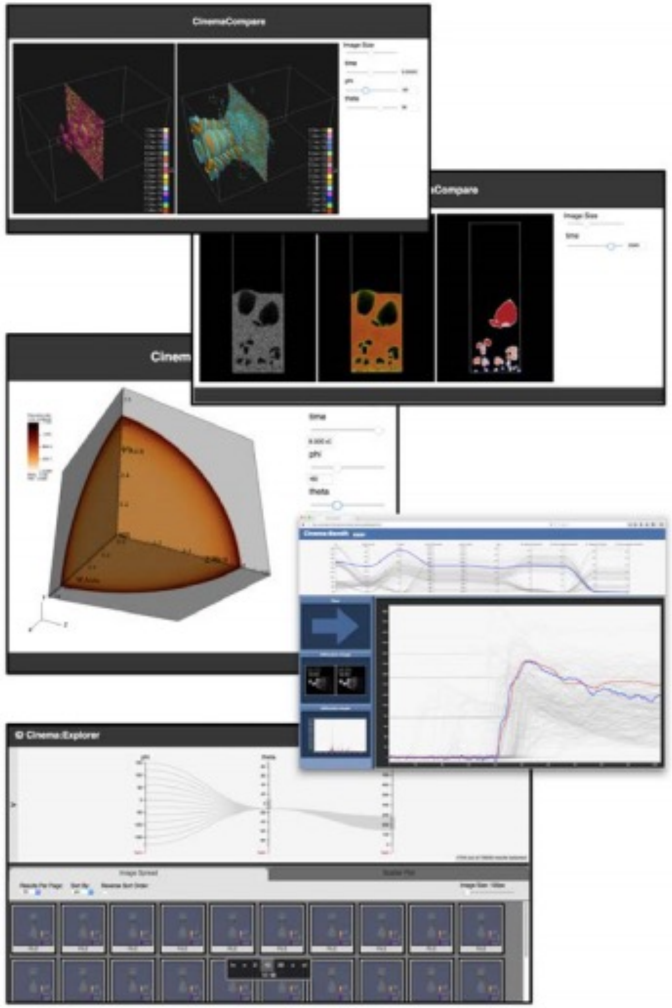
Demonstrated

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.



Slide courtesy of Los Alamos National Lab and the CINEMA project

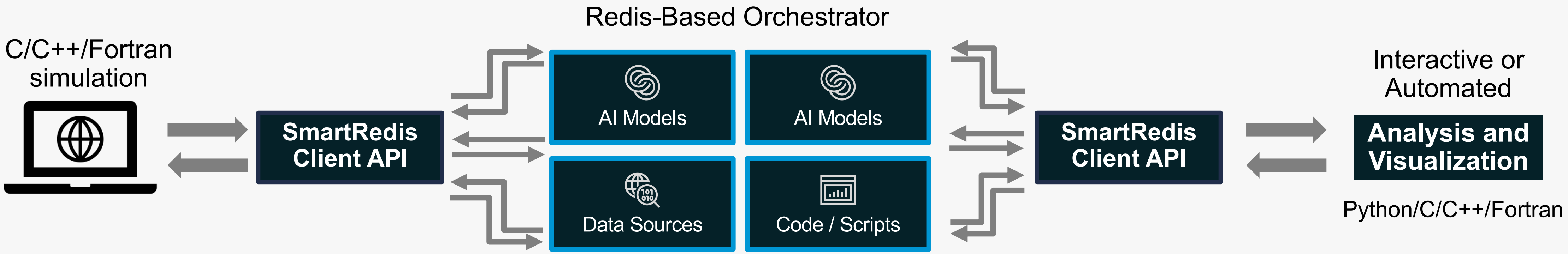
SmartSim Overview

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**



PYTORCH | TENSORFLOW | ONNX

slide courtesy of the HPE SmartSim team

Scaling Computational Fluid Dynamics: In Situ Visualization of NekRS using SENSEI

Nekrs + SENSEI

🏆 Mateevitsi, Victor A., Mathis Bode, Nicola Ferrier, Paul Fischer, Jens Henrik Göbbert, Joseph A. Insley, Yu-Hsiang Lan et al. "Scaling Computational Fluid Dynamics: In Situ Visualization of NekRS using SENSEI." In *Proceedings of the SC'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, pp. 862-867. 2023.

Introduction

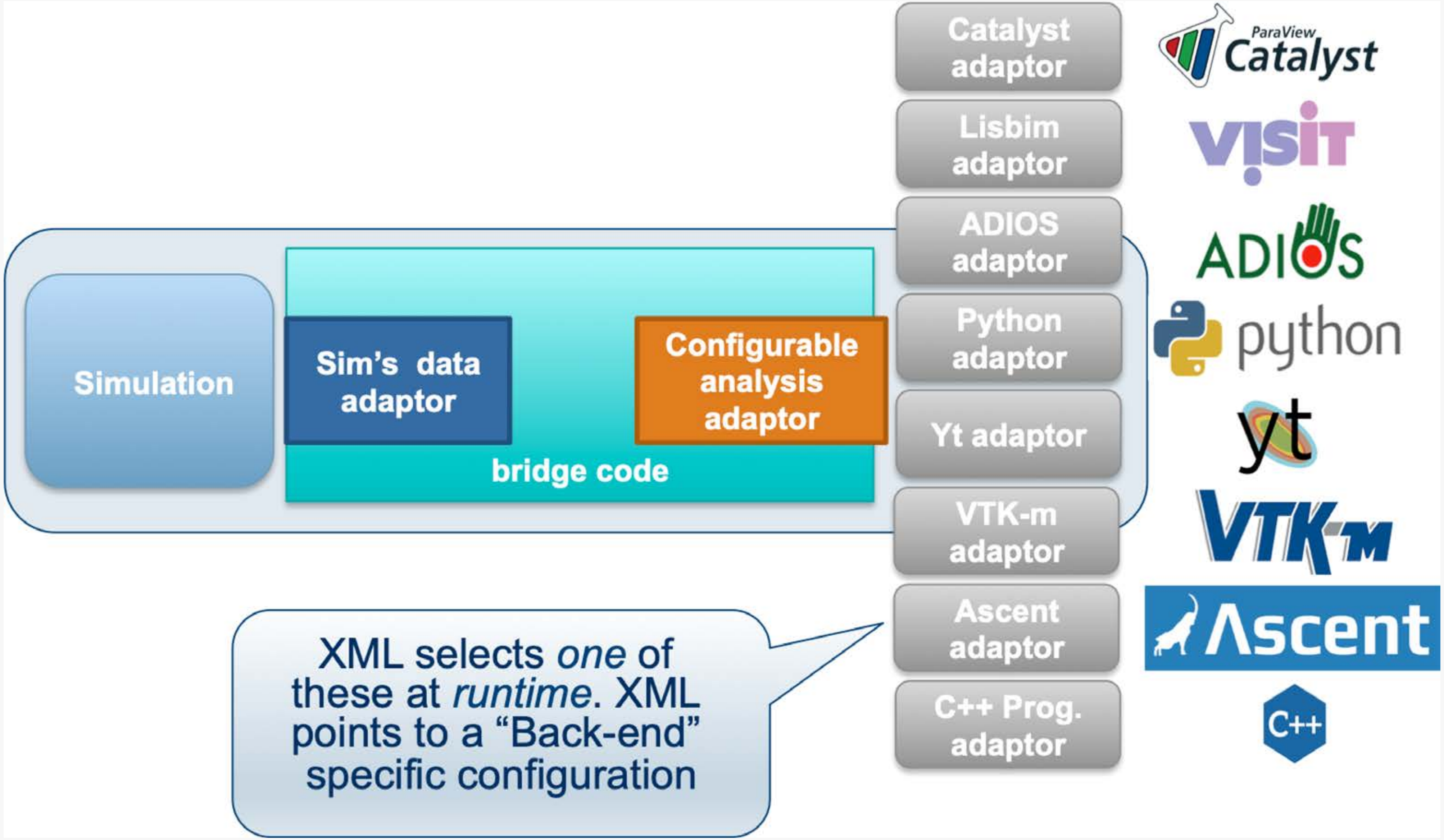
- NekRS
    - Rooted in the Spectral Element Method (SEM)
    - GPU-accelerated thermal-fluid simulation code
    - Predecessor is Nek5000
    - Supports modern heterogenous systems (CPU/GPU)

- Exascale and I/O
    - Exascale machines
        - Disparity between on-chip processing and disk storage is set to widen
    - Data saving to disk notably hampers simulations
        - Tough choice: reduce checkpointing OR simplify the domain

- **Solution**: *In situ* and in transit processing
    - *In situ*: facilitates data processing while in memory
    - In transit: offloads data processing to a set secondary resources

- **How?**
    - SENSEI

SENSEI



XML selects *one* of these at *runtime*. XML points to a "Back-end" specific configuration

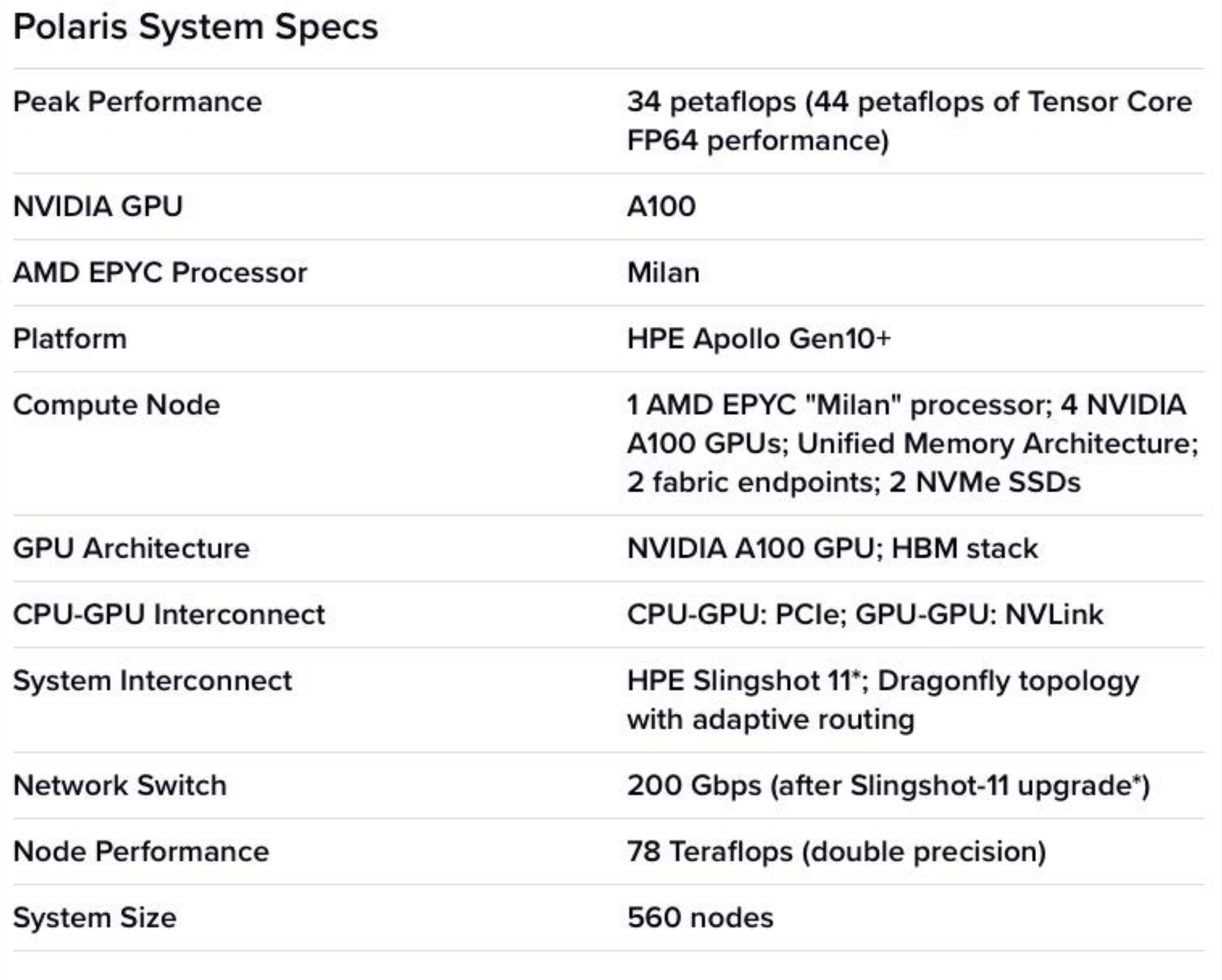
Experiments

- Goal
  - Quantify the **computational overhead** introduced by *in situ* and in transit methodologies to CFD codes
- Resources
  - The *in situ* case run on Polaris, at ALCF
  - The in transit case run on JUWELS Booster, at the Jülich Supercomputing Centre
- Reproducibility
  - All source code, analysis code, and use cases have been made available[1]

---

1. Victor A. Mateevitsi, Mathis Bode, Nicola Ferrier, Paul Fischer, Jens Henrik Göbbert, Joseph A. Insley, Yu-Hsiang Lan, Misun Min, Michael E. Papka, Saumil Patel, Silvio Rizzi, and Jonathan Windgassen. 2023. Software and Analysis for paper: Scaling Computational Fluid Dynamics: In Situ Visualization of NekRS using SENSEI. https://doi.org/10.5281/zenodo.8377974

Polaris



Polaris System Specs

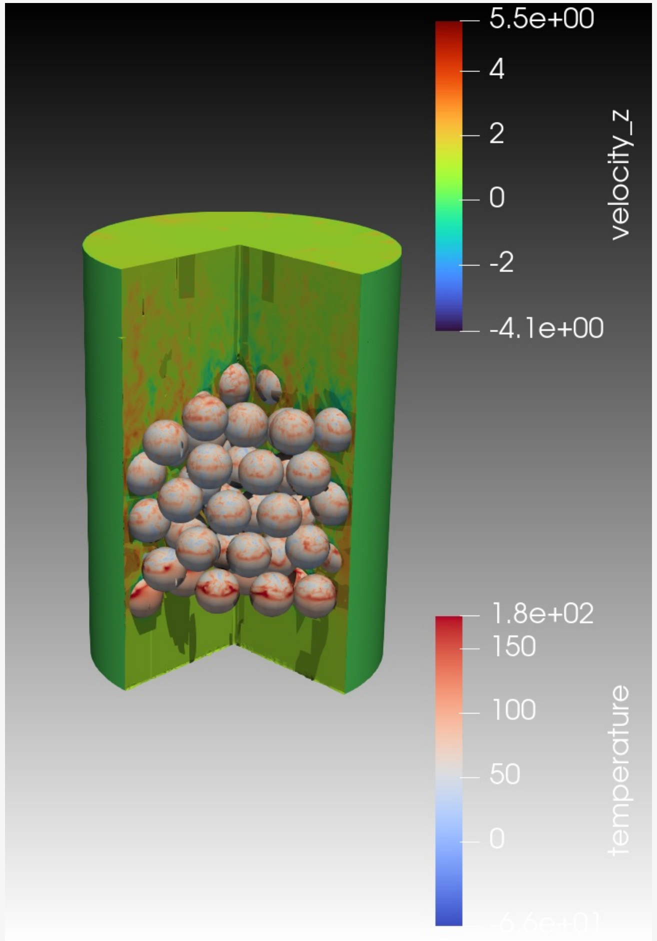
Peak Performance	34 petaflops (44 petaflops of Tensor Core FP64 performance)
NVIDIA GPU	A100
AMD EPYC Processor	Milan
Platform	HPE Apollo Gen10+
Compute Node	1 AMD EPYC "Milan" processor; 4 NVIDIA A100 GPUs; Unified Memory Architecture; 2 fabric endpoints; 2 NVMe SSDs
GPU Architecture	NVIDIA A100 GPU; HBM stack
CPU-GPU Interconnect	CPU-GPU: PCIe; GPU-GPU: NVLink
System Interconnect	HPE Slingshot 11*; Dragonfly topology with adaptive routing
Network Switch	200 Gbps (after Slingshot-11 upgrade*)
Node Performance	78 Teraflops (double precision)
System Size	560 nodes

Results – In situ Pebble-bed reactor case

- Metrics
  - Runtime
    - total elapsed wall-clock time
  - Memory footprint
    - aggregate memory high water mark across all MPI ranks.

- Configurations
  - **Original:** NekRS sans SENSEI
  - **Checkpointing:** NekRS with built-in checkpointing
  - **Catalyst:** NekRS with SENSEI, employing the Catalyst Adaptor

- Pebble-bed reactor case
  - Pb146 use case simulation from NekRS codebase
  - representation of a pebble-bed nuclear reactor core, housing 146 spherical pebbles
  - Such a simulation is of particular interest, given the growing interest in advanced carbon-neutral nuclear fission reactors

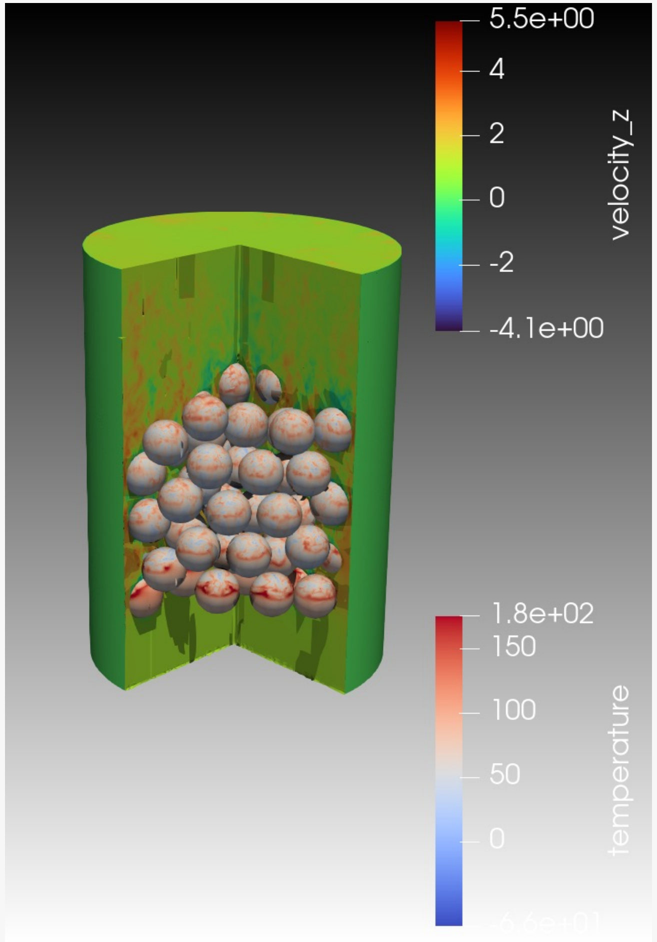


Visualization of the pb146 use case simulation, illustrating flow dynamics within a pebble-bed nuclear reactor

Results – In situ Pebble-bed reactor case

- NekRS simulation
  - Runs on the GPU
  - Ran for 3,000 timesteps
  - Checkpointing and in situ processing at 100 timestep intervals

- Scale
  - 70 nodes – 280 ranks (12.5% of Polaris)
  - 140 nodes – 560 ranks (25% of Polaris)
  - 280 nodes – 1,120 ranks (50% of Polaris)



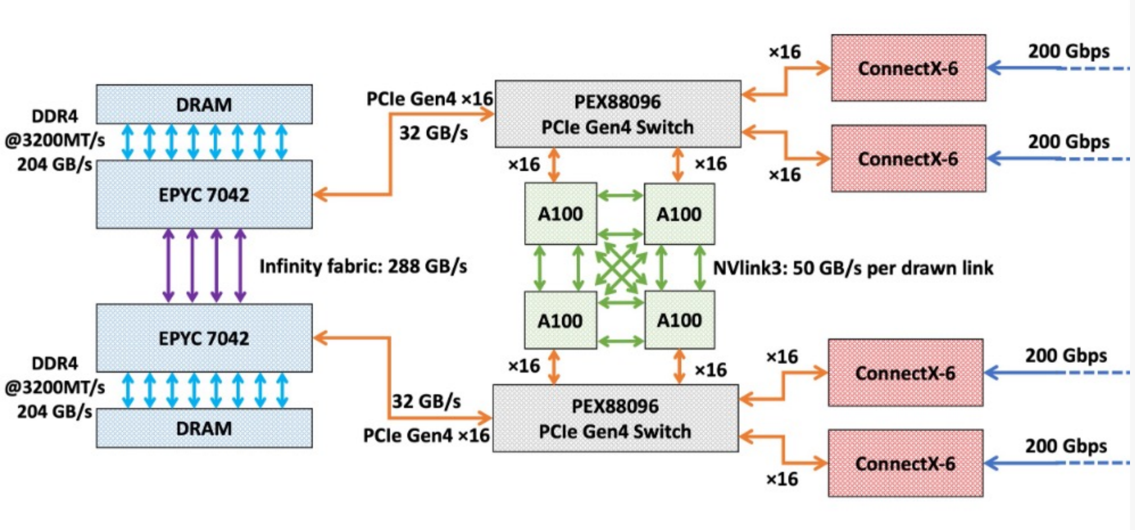
Visualization of the pb146 use case simulation, illustrating flow dynamics within a pebble-bed nuclear reactor

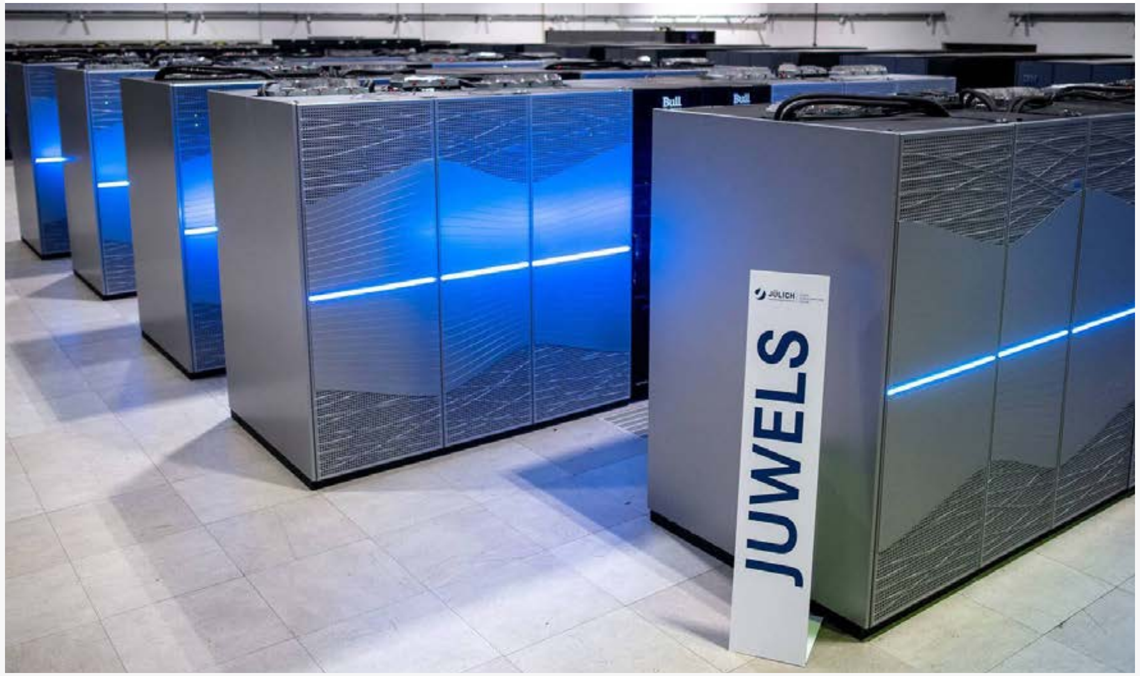
JUWELS Booster

Peak Performance	70.98 PFLOPs
System Size	936 nodes
Platform	ATOS BullSequana
Setup	2020
Top500	13. (06/2023)

- **Compute Node**
  - 2x AMD EPYC 7402 24-core, 2.8GHz
  - 512 GB DDR memory
  - 4x NVIDIA A100 GPUs
  - 4x Mellanox HDF200 Infiniband
  - 78 TFLOPs (GPUs)

- **System Interconnect**
  - Mellanox Infiniband
  - DragonFly+ topology
  - Adaptive routing





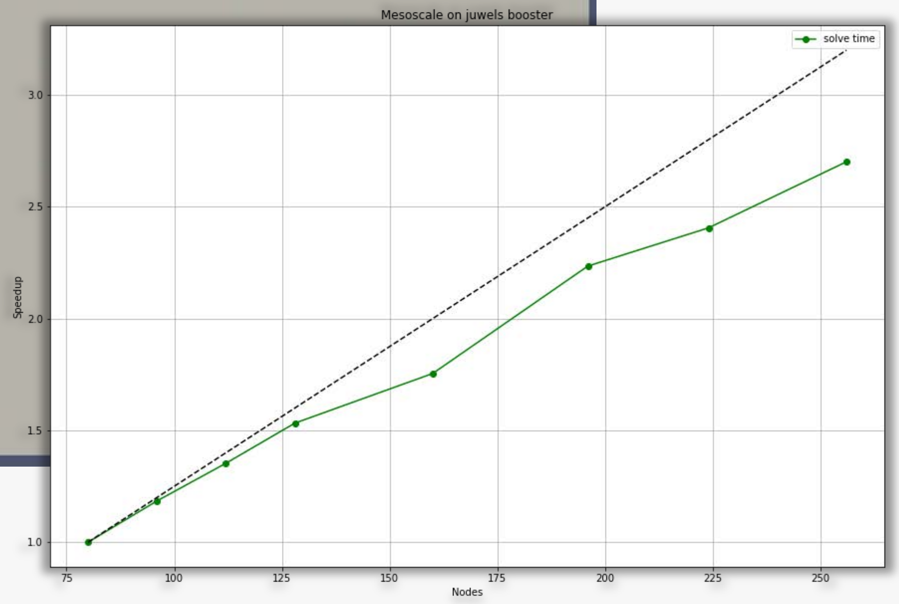
Results – In transit Mesoscale case

- **Mesoscale case**
  - Rayleigh-Bénard convection (RBC)
    - classical natural convection type

    Basic setup leading to RBC
    - fluid heated from below
  - Such simulation is of particular interest to study unusual dynamics of turbulent convection in the sun [1].

- **Simulation**
  - Periodic BCs in width and length direction
  - In z direction:
    Temperature: Dirichlet, Velocity: no slip
  - Rayleigh number up to 1e12 (full JUWELS Booster runs)
    - examples here are 1e5

Visualization of the temperature field



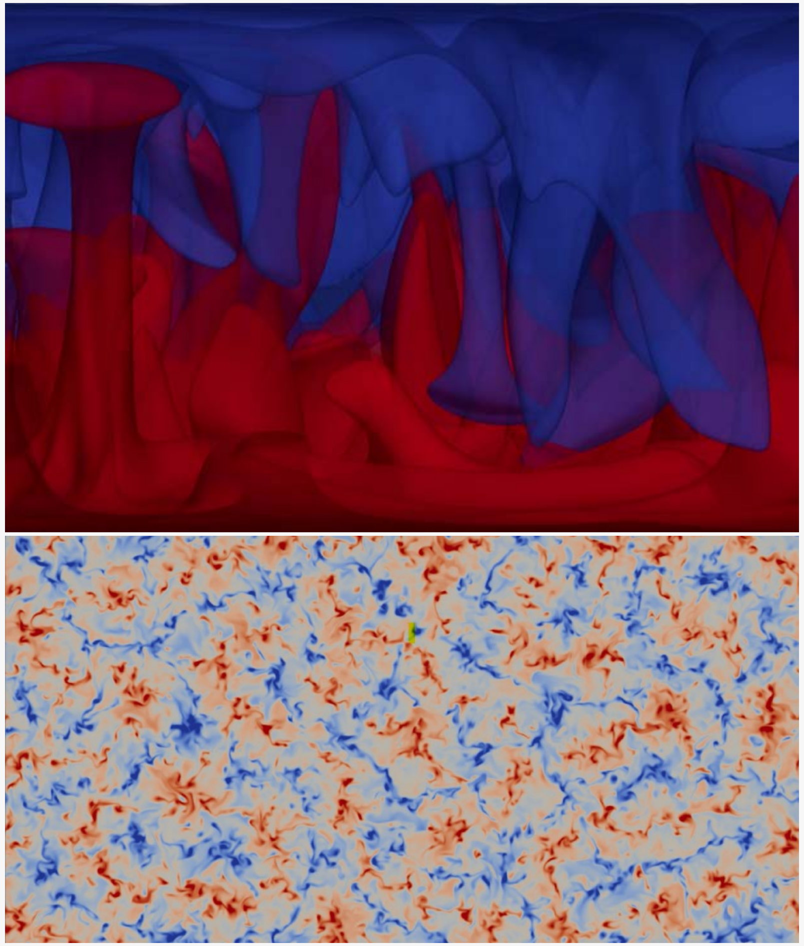
Strong-scaling plot for JUWELS Booster

[1] Convective mesoscale turbulence at very low Prandtl numbers
Ambrish Pandey, Dmitry Krasnov, Katepalli R. Sreenivasan and Jörg Schumacher

Results – In transit Mesoscale case

- **In transit configurations**
  - **No Transport:** No SENSEI endpoint
    - Reference measurement
    - No SENSEI analysis adapter connected
  - **Checkpointing:** SENSEI endpoint writes VTU files
    - pressure and velocity fields
  - **Catalyst:** SENSEI endpoint passes data to Catalyst
    - Renders two images using ParaView over Python

  - Endpoint: SENSEI data consumer
  - Ratio of simulation- to endpoint nodes: 4:1
  - Sustainable Staging Transport (SST) engine of ADIOS2
    - Communication: UCX
    - Control operations: TCP sockets on Infiniband
    - Data marshaling option: BP4



Visualization of the RBC case.
A side view and a top view colored by temperature.