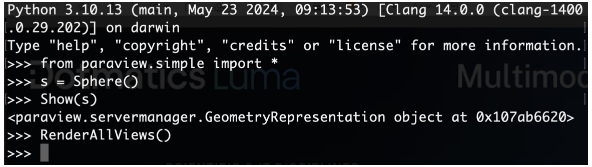
CASS ParaView and Catalyst BOF Agenda

- Introduction - 5 minutes
- Talks - 30 minutes
  - ParaView Status and Future Directions, Cory Quammen, Kitware, Inc.
  - Catalyst, Corey Wetterer-Nelson, Kitware, Inc.
- Panel presentations - 40 minutes
  - Ken Moreland, Oak Ridge National Laboratory
  - Andy Bauer, US Army Corps of Engineers
  - Axel Huebl, Berkeley Lab
  - Joe Insley, Argonne Leadership Computing Facility
- Q&A with the panel - 20 minutes

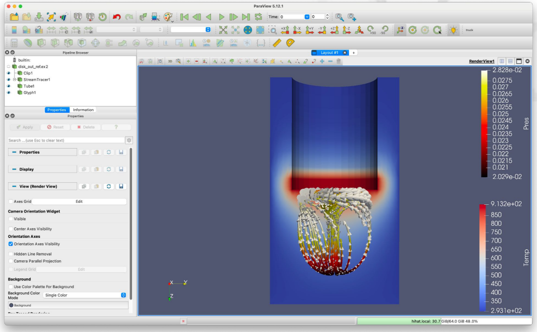
ParaView Status and Future Directions

Cory Quammen, Kitware, Inc.

ParaView is a suite of applications for visualizing and analyzing massive datasets



```
Python 3.10.13 (main, May 23 2024, 09:13:53) [Clang 14.0.0 (clang-1400
.0.29.202)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from paraview.simple import *
>>> s = Sphere()
>>> Show(s)
<paraview.servermanager.GeometryRepresentation object at 0x107ab6620>
>>> RenderAllViews()
>>> 
```



pvserver

Remote MPI process for loading very large data, processing it, and rendering

pvpython

Python console equivalent to the ParaView application

ParaView

Desktop application for general purpose visualization and analysis

trame

Web-based applications with visualizations generated by ParaView

pvbatch

Python interpreter for batch processing of large data

catalyst

ParaView is a Catalyst 2 implementation

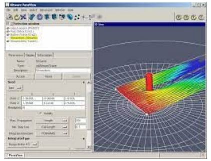
A brief history of ParaView

2000

ParaView Project Started

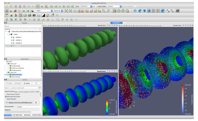
2002

V0.6 released
Used TCL/TK



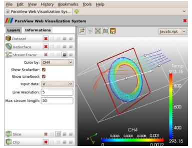
2007

V3.0 released
Redesigned application
using Qt and Python



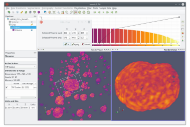
CMB project started!

ParaViewWeb was born!



2010

Catalyst was released

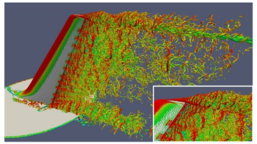


TomViz was released!

2016

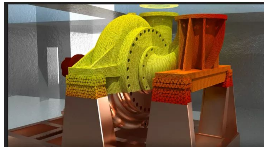
V5.0 released
Refactored rendering using
OpenGL 4+

Catalyst: 1 million rank run!

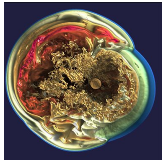


2018

V5.6 released
With advanced /realistic
rendering using OSPRay /
Optix Integration



2021

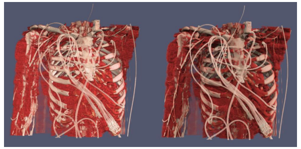


V5.10 released

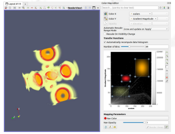
Catalyst 2 was
released

2022

V5.11 released
With Global Illumination



2024



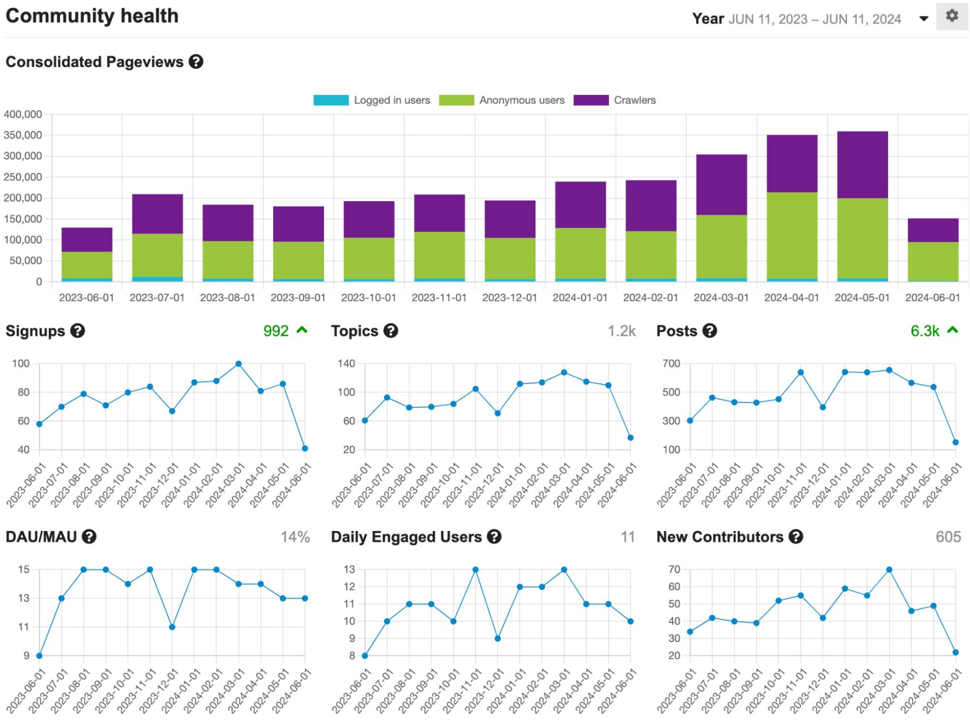
V5.12.1 released
(most recent version)

Latest release - ParaView 5.12

- **5.12.0 released on March 1, 2024**
  - ~119 issues fixed
- **5.12.1 released on May 24, 2024**
  - 14 issues fixed
- **Binary downloads for 5.12.x from paraview.org (up to 6/11/2024):**
  - Windows binaries: **39,113**
  - Linux binaries: **13,912**
  - macOS binaries: **9,669** arm64 + x86_64
  - osmesa + egl: **1,187**

ParaView community

- discourse.paraview.org
- gitlab.kitware.com
- docs.paraview.org



Community health Year JUN 11, 2023 – JUN 11, 2024

Consolidated Pageviews

Signups 992

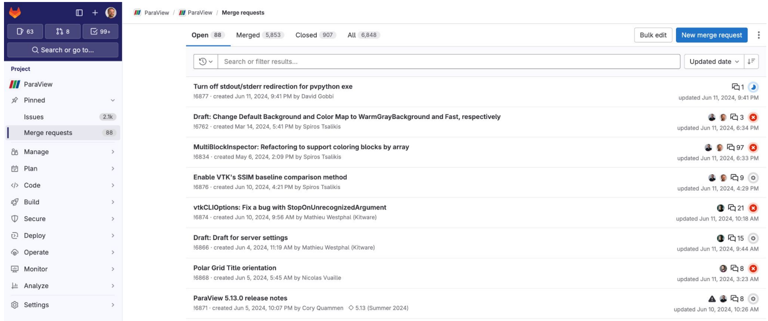
Topics 1.2k

Posts 6.3k

DAU/MAU 14%

Daily Engaged Users 11

New Contributors 605



Turn off stdout/stderr redirection for pvpython exe
!6877 · created Jun 11, 2024, 9:41 PM by David Gobbi · 1 · updated Jun 11, 2024, 9:41 PM

Draft: Change Default Background and Color Map to WarmGrayBackground and Fast, respectively
!6762 · created Mar 14, 2024, 5:41 PM by Spiros Tsalikis · 3 · updated Jun 11, 2024, 6:34 PM

MultiBlockInspector: Refactoring to support coloring blocks by array
!6834 · created May 6, 2024, 2:09 PM by Spiros Tsalikis · 97 · updated Jun 11, 2024, 6:33 PM

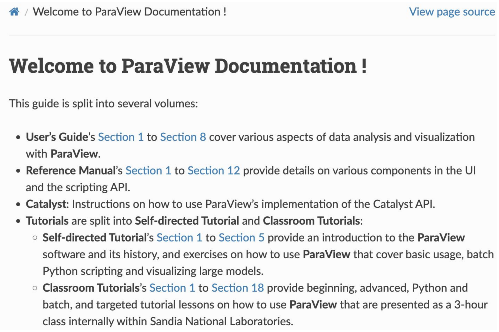
Enable VTK's SSIM baseline comparison method
!6876 · created Jun 10, 2024, 4:21 PM by Spiros Tsalikis · 9 · updated Jun 11, 2024, 4:29 PM

vtkCLIOptions: Fix a bug with StopOnUnrecognizedArgument
!6874 · created Jun 10, 2024, 9:56 AM by Mathieu Westphal (Kitware) · 21 · updated Jun 11, 2024, 10:18 AM

Draft: Draft for server settings
!6866 · created Jun 4, 2024, 11:19 AM by Mathieu Westphal (Kitware) · 15 · updated Jun 11, 2024, 9:44 AM

Polar Grid Title orientation
!6868 · created Jun 5, 2024, 5:45 AM by Nicolas Vuaille · 8 · updated Jun 11, 2024, 3:23 AM

ParaView 5.13.0 release notes
!6871 · created Jun 5, 2024, 10:07 PM by Cory Quammen ◇ 5.13 (Summer 2024) · 8 · updated Jun 10, 2024, 10:26 AM



Welcome to ParaView Documentation !

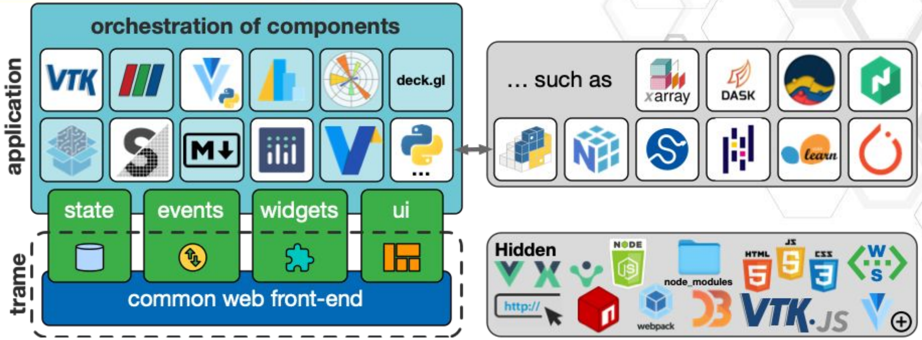
This guide is split into several volumes:

- **User's Guide**'s Section 1 to Section 8 cover various aspects of data analysis and visualization with **ParaView**.
- **Reference Manual**'s Section 1 to Section 12 provide details on various components in the UI and the scripting API.
- **Catalyst**: Instructions on how to use ParaView's implementation of the Catalyst API.
- **Tutorials** are split into **Self-directed Tutorial** and **Classroom Tutorials**:
  - **Self-directed Tutorial**'s Section 1 to Section 5 provide an introduction to the **ParaView** software and its history, and exercises on how to use **ParaView** that cover basic usage, batch Python scripting and visualizing large models.
  - **Classroom Tutorials**'s Section 1 to Section 18 provide beginning, advanced, Python and batch, and targeted tutorial lessons on how to use **ParaView** that are presented as a 3-hour class internally within Sandia National Laboratories.

Current work

Web visualization with trame and ParaView

- **trame** is a Python library for creating interactive applications with a web front-end without the complexity of web development
- Builtin:
  - *Ubiquitous* (same code can be used in many contexts)
  - *Client/Server* architecture & graphical interface
  - *Asynchronous*, *reactive*, *collaborative* and *distributed*



application

orchestration of components

... such as

state

events

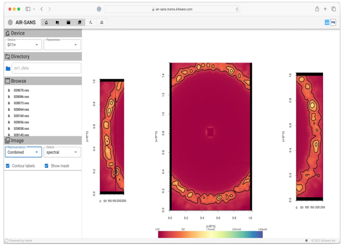
widgets

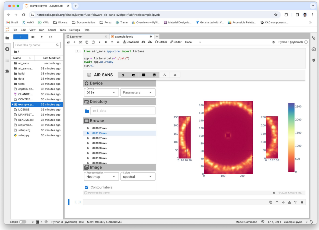
ui

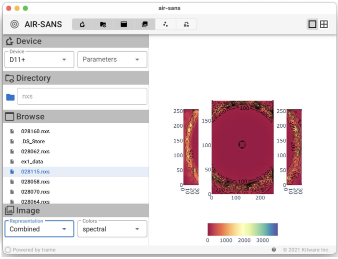
trame

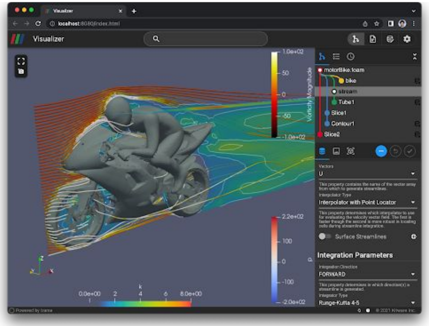
common web front-end

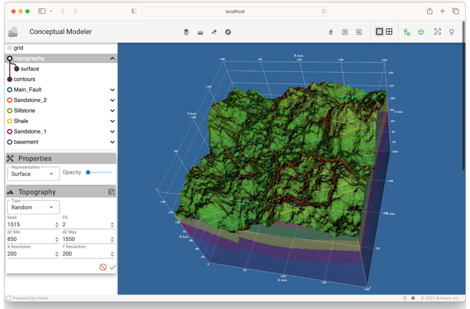
Hidden











I/O improvements

- **ADIOS2** - The Adaptable Input Output (I/O) System V2
  - Available since ParaView 5.8
  - Ongoing version updates and bug fixes
- **VTKHDF** - VTK-native format intended as an easy way to write data for ParaView using only HDF5
  - More complete dataset type support coming

Accelerated filtering

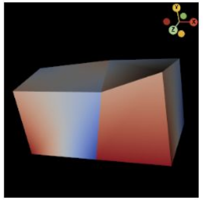
- Multithreading VTK filters is well underway
  - Most core algorithms now make full use of all available cores
  - Identifying more filters for improvement
- Publication:

Spiros Tsalikis, Will Schroeder, Daniel Szafir, Kenneth Moreland:
An Accelerated Clip Algorithm for Unstructured Meshes - A Batch-Driven Approach,
Eurographics Symposium on Parallel Graphics and Visualization 2024

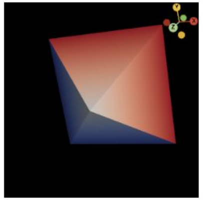
- More filters that use Viskores/VTK-m have been integrated in ParaView 5.12
  - Clip
  - Slice
  - Threshold

Viskores

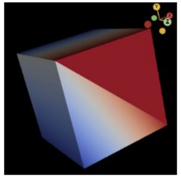
Discontinuous Galerkin and other higher order elements



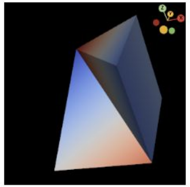
Hexahedra



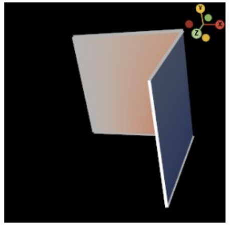
Tetrahedra



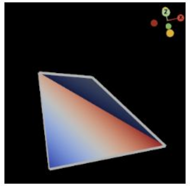
Wedges



Pyramids



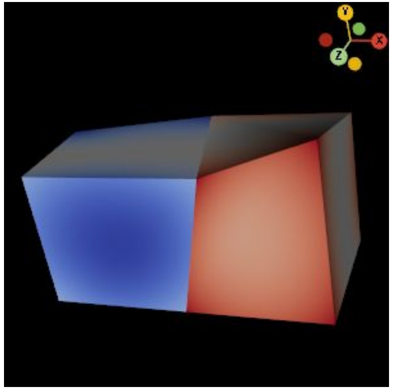
Quadrilaterals



Triangles

Discontinuous (and continuous) attributes supported on these cell types

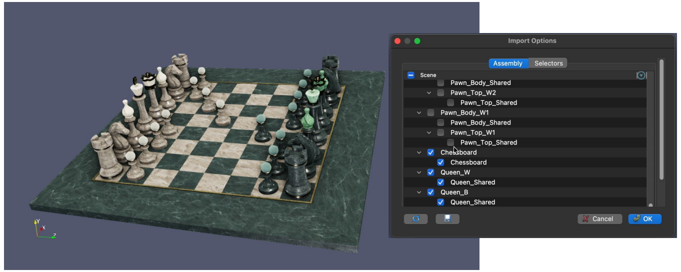
Colors evaluated on GPU - no discretization

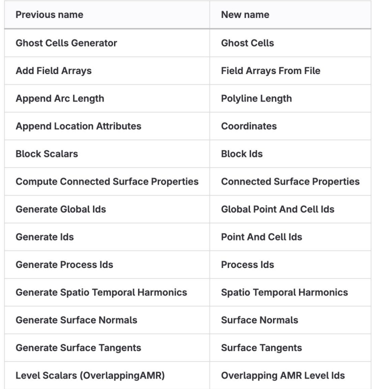


Non-isoparametric elements (shape and color attributes do not have the same polynomial order)

ParaView 5.13 - release in progress

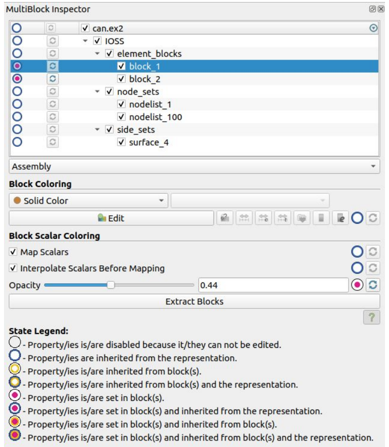
GLTF importer





Previous name	New name
Ghost Cells Generator	Ghost Cells
Add Field Arrays	Field Arrays From File
Append Arc Length	Polyline Length
Append Location Attributes	Coordinates
Block Scalars	Block Ids
Compute Connected Surface Properties	Connected Surface Properties
Generate Global Ids	Global Point And Cell Ids
Generate Ids	Point And Cell Ids
Generate Process Ids	Process Ids
Generate Spatio Temporal Harmonics	Spatio Temporal Harmonics
Generate Surface Normals	Surface Normals
Generate Surface Tangents	Surface Tangents
Level Scalars (OverlappingAMR)	Overlapping AMR Level Ids

Renamed filters



Block Coloring

Solid Color

Block Scalar Coloring

☑ Map Scalars

☑ Interpolate Scalars Before Mapping

Opacity 0.44

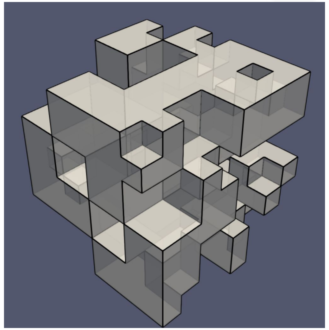
State Legend:

- ○ - Property/ies is/are disabled because it/they can not be edited.
- ○ - Property/ies are inherited from the representation.
- ○ - Property/ies is/are inherited from block(s).
- ○ - Property/ies is/are inherited from block(s) and the representation.
- ● - Property/ies is/are set in block(s).
- ● - Property/ies is/are set in block(s) and inherited from the representation.
- ● - Property/ies is/are set in block(s) and inherited from block(s).
- ● - Property/ies is/are set in block(s) and inherited from block(s) and the representation.

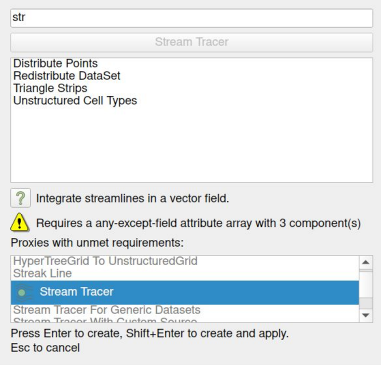
More per-block properties available

Color different blocks by different arrays!

Improved Hyper-tree Grid support



Better filter search



str

Stream Tracer

Distribute Points
Redistribute DataSet
Triangle Strips
Unstructured Cell Types

? Integrate streamlines in a vector field.

⚠ Requires a any-except-field attribute array with 3 component(s)

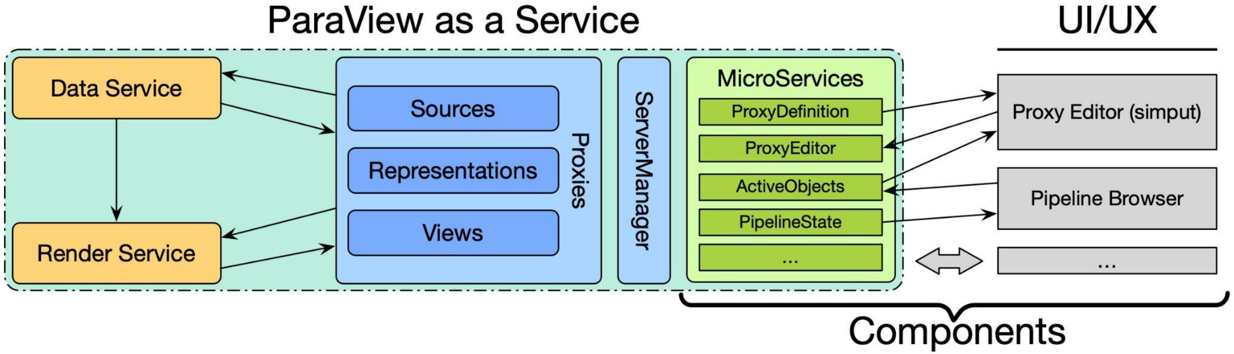
Proxies with unmet requirements:

HyperTreeGrid To UnstructuredGrid
Streak Line
Stream Tracer
Stream Tracer For Generic Datasets

Press Enter to create, Shift+Enter to create and apply.
Esc to cancel

What’s next

ParaView Async



Responsive backend that can respond to user interactions promptly

Interruptible backend that can react to changes in processing needs quickly

Modular elements for composing desktop and web solutions

Future of rendering in ParaView

Rasterization

- OpenGL is waning
  - 2017 last specification
  - Support on Macs declining
- Fragmented graphics libraries left in its place
  - Vulkan, Metal, DirectX
- WebGPU
  - Cross platform and web-ready
  - Still pretty new

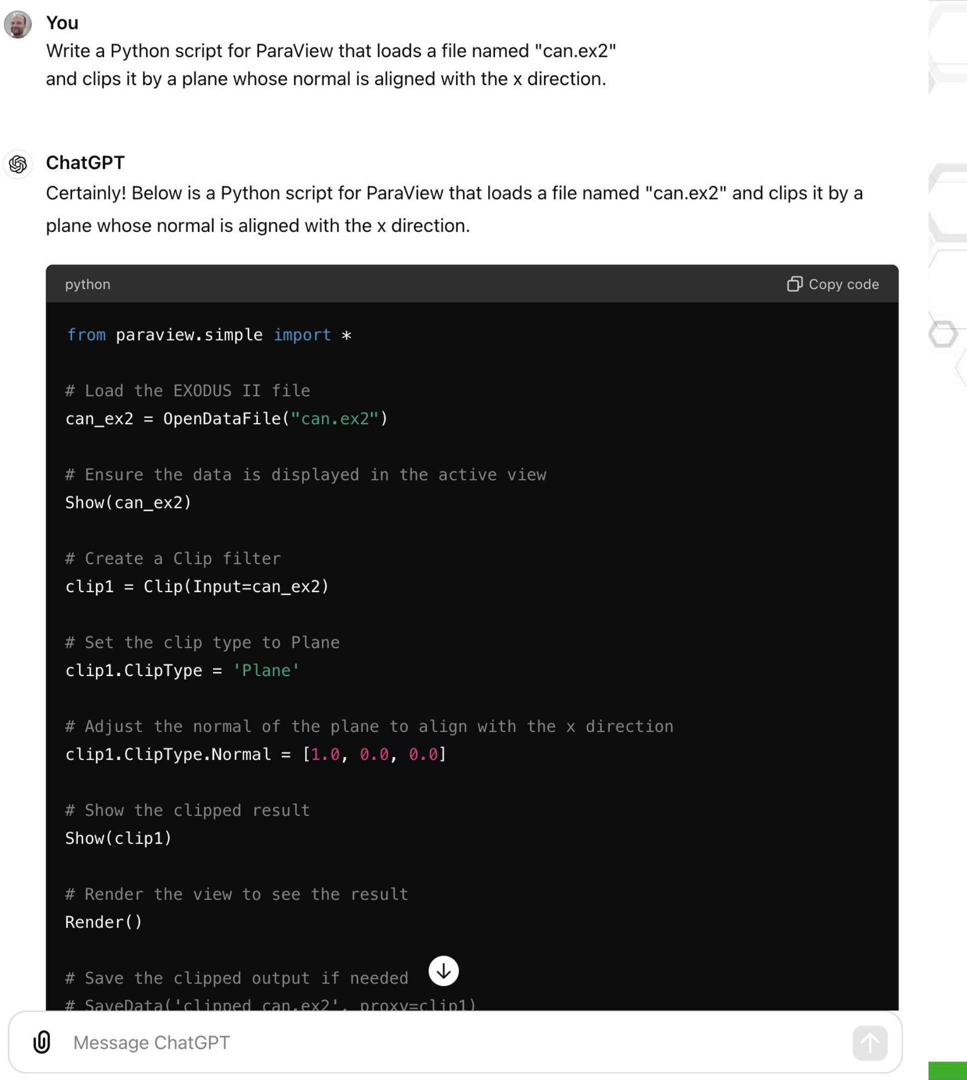
Ray tracing

- Current ray tracing options
  - intel. OSPRay
  - NVIDIA. OPTIX™
- ANARI backend is in VTK, in development for ParaView
  - Unified API for ray tracing backends from Intel, NVIDIA, AMD
  - Work in progress

ANARI™

AI for visualization

- Use natural language to design visualizations through Python scripting
- Example from ChatGPT
- Add an AI console to ParaView?
    - How to train and provide LLM?
    - How to use it securely?



You

Write a Python script for ParaView that loads a file named "can.ex2" and clips it by a plane whose normal is aligned with the x direction.

ChatGPT

Certainly! Below is a Python script for ParaView that loads a file named "can.ex2" and clips it by a plane whose normal is aligned with the x direction.

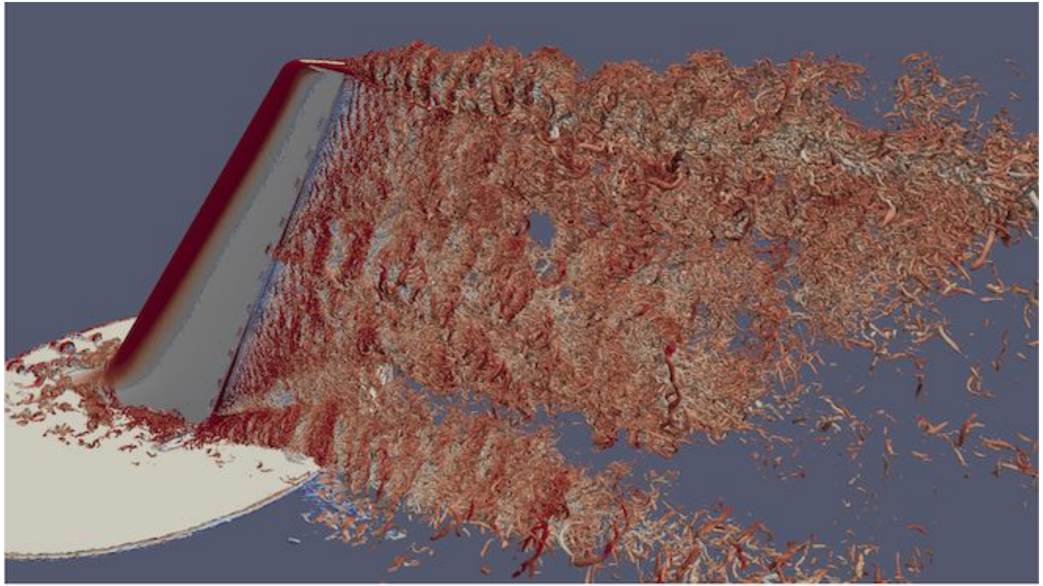
```python
from paraview.simple import *

# Load the EXODUS II file
can_ex2 = OpenDataFile("can.ex2")

# Ensure the data is displayed in the active view
Show(can_ex2)

# Create a Clip filter
clip1 = Clip(Input=can_ex2)

# Set the clip type to Plane
clip1.ClipType = 'Plane'

# Adjust the normal of the plane to align with the x direction
clip1.ClipType.Normal = [1.0, 0.0, 0.0]

# Show the clipped result
Show(clip1)

# Render the view to see the result
Render()

# Save the clipped output if needed
# SaveData('clipped_can.ex2', proxy=clip1)
```

Thank you!

Catalyst

Corey Wetterer-Nelson Ph.D.

kitware

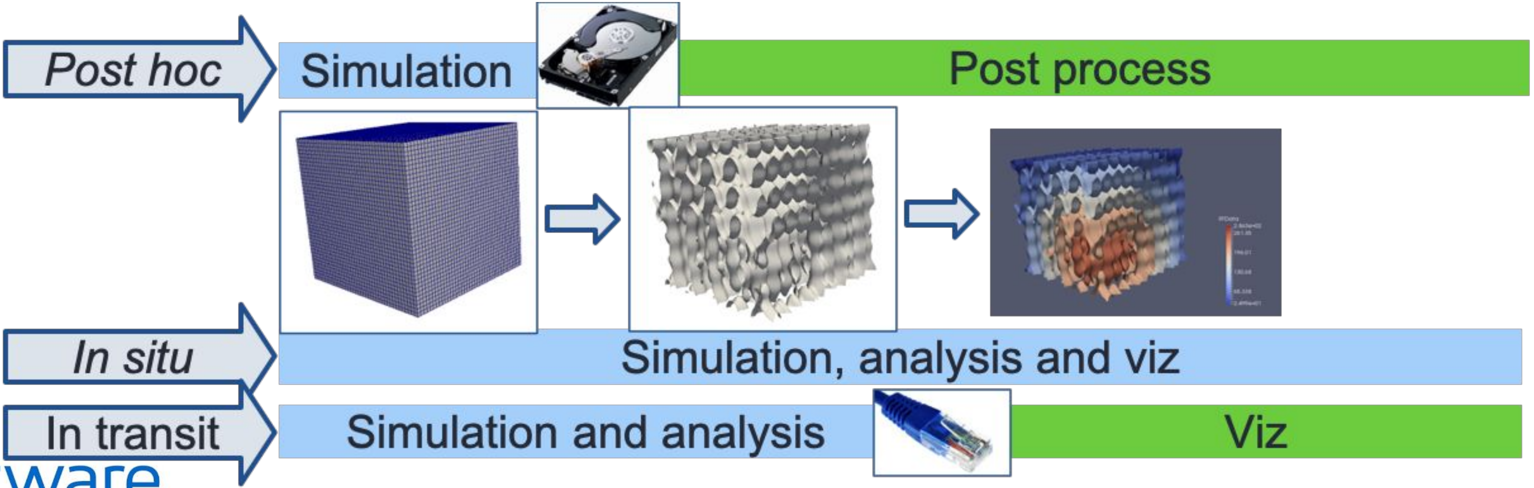


- In Situ With ParaView
- ECP Successes
- GPU Native Workflows
- In Transit
- Future Work

In Situ Analysis & Visualization

Concurrently computing analysis and visualization tasks while a simulation runs

AKA: Covisualization, concurrent processing, coprocessing



Post hoc

Simulation

Post process

In situ

Simulation, analysis and viz

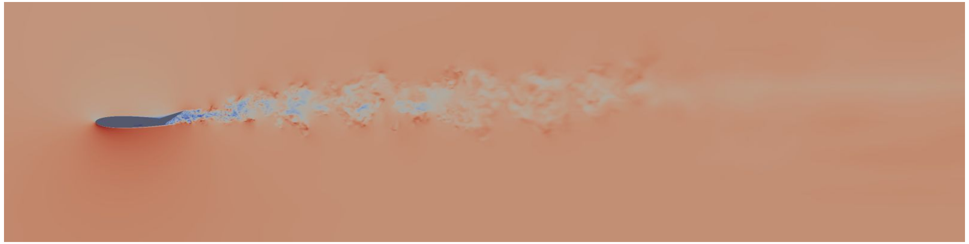
In transit

Simulation and analysis

Viz

In Situ Analysis with Catalyst

- Production-ready Software
- Scaled to 1+ Million MPI ranks
- HPCWire Best HPC Visualization Product or Technology
- Editor's Choice - 2011, 2012, 2014 (runner-up), 2016
- Reader's Choice - 2015 tie (Paraview)
- Used on HPC architectures



ParaView Catalyst visualizes flow around a jet for 1+ million MPI ranks in situ simulation run.

ECP Successes

MFix-Exa Catalyst Integration

Demonstrated on Frontier

649 nodes, 5187 GPUs

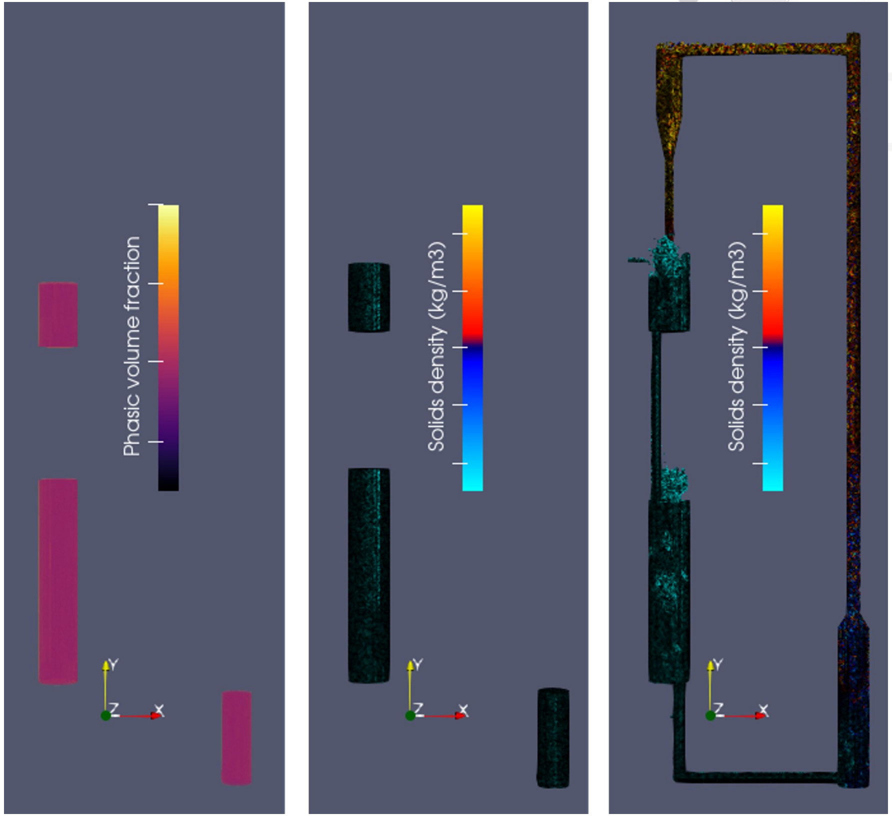
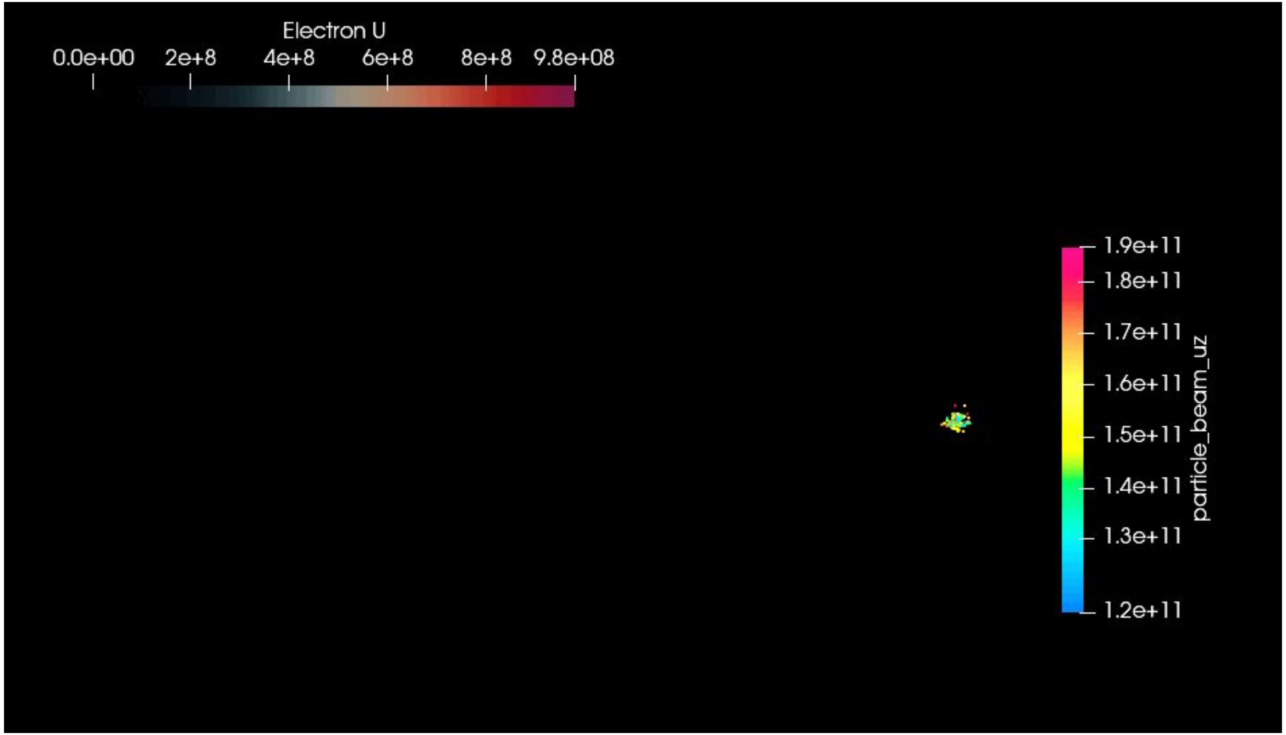


Image credits: Abhishek Yenpure

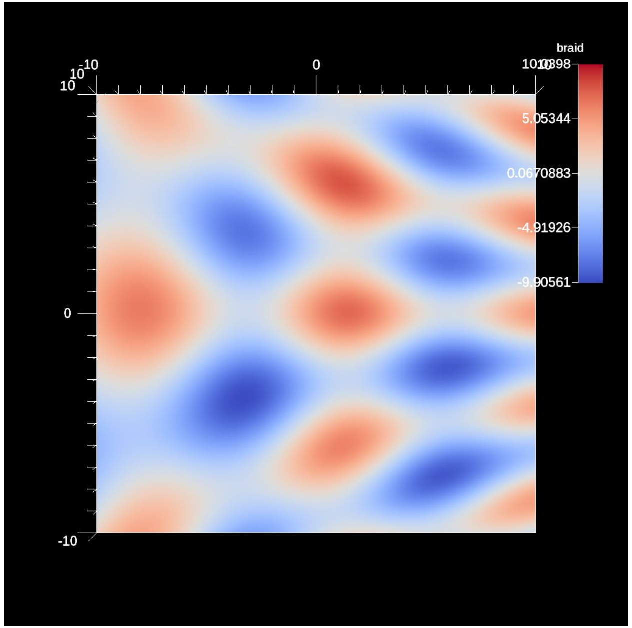
ECP Successes

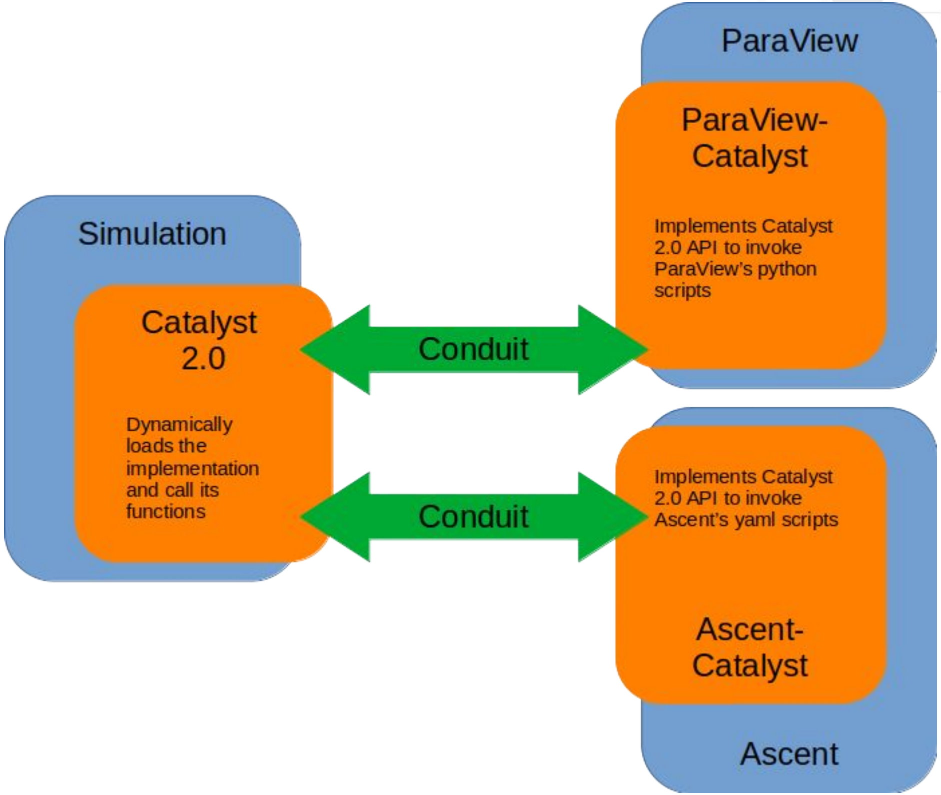
WarpX Integration



ECP Successes

Ascent Backend





Simulation

Catalyst 2.0

Dynamically loads the implementation and call its functions

Conduit

Conduit

ParaView

ParaView-Catalyst

Implements Catalyst 2.0 API to invoke ParaView's python scripts

Implements Catalyst 2.0 API to invoke Ascent's yaml scripts

Ascent-Catalyst

Ascent

Image credits: Abhishek Yenpure

ECP Successes

Native Fortran and Python Support

CATALYST_WRAP_PYTHON=ON

CATALYST_WRAP_FORTRAN=ON

GPU Native Workflows

```
channel["memoryspace"].set("cuda");
```

VTK-m **`ArrayHandle`** wraps GPU pointers, enabling accelerated pipelines while keeping GPU-resident data on the GPU.

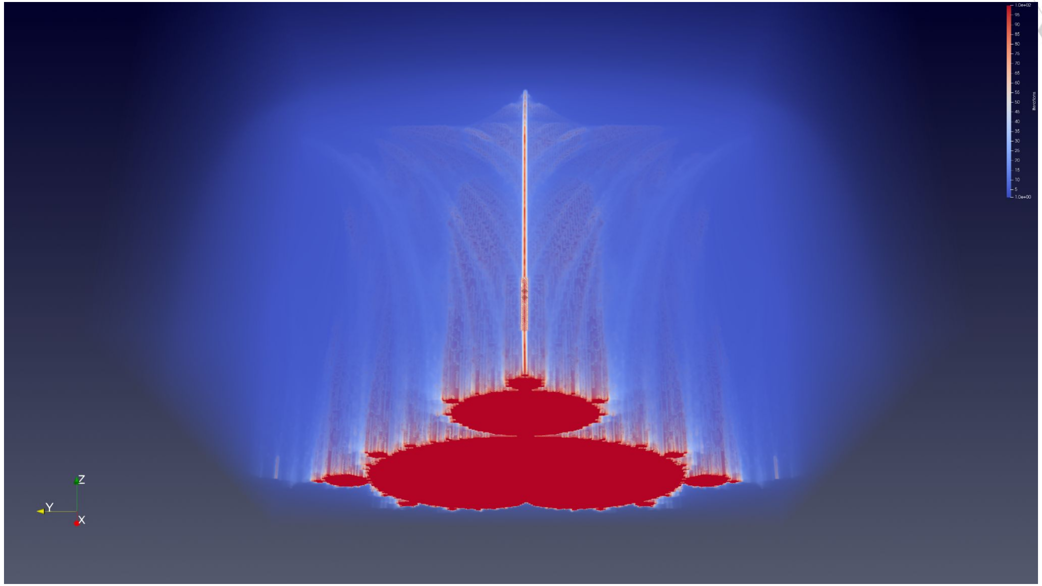
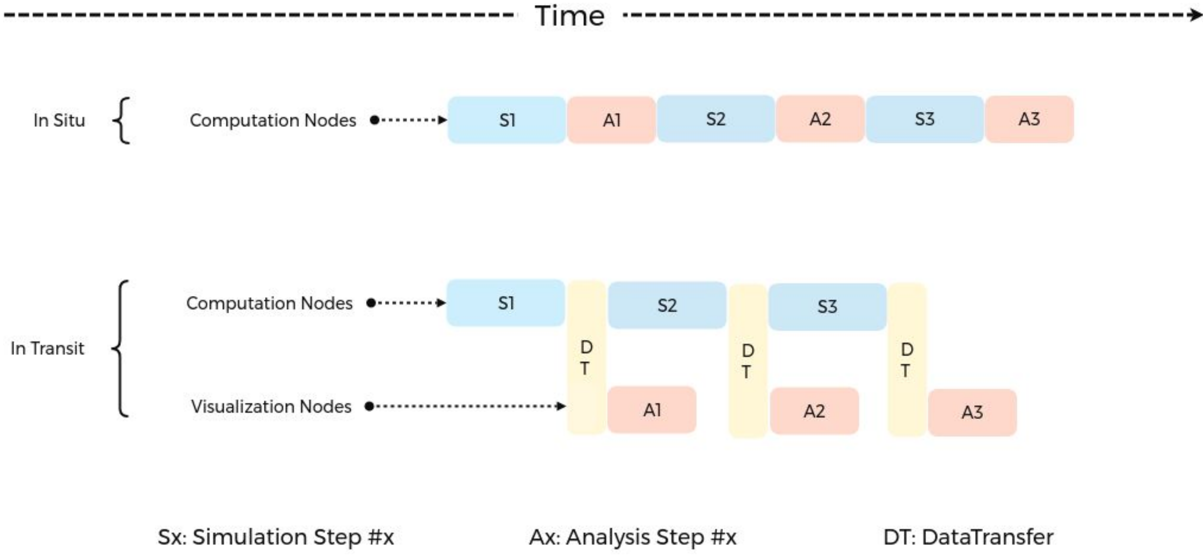


Image credits: Jaswant Panchumarti

In Transit with ADIOS



ADIOSCatalyst

Streams data from simulation to a visualization resource

ADIOS Replay

Converts streamed data to Catalyst data for visualization ParaView

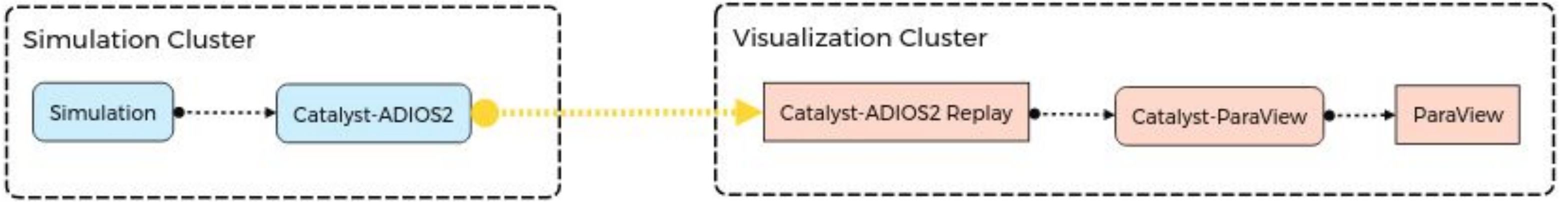


Image credits: Lucas Givord, Francois Mazen, and Charles Gueunet

What the Future Holds

More GPU Integration

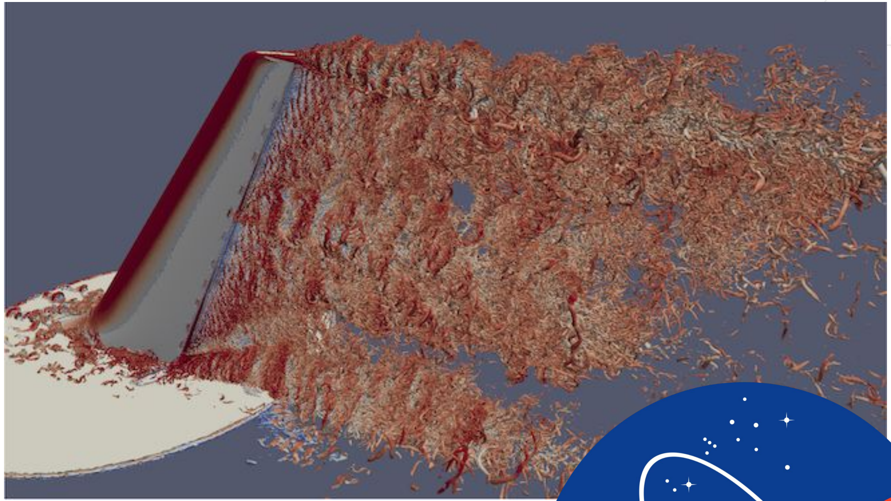
AI/ML Experiments

High Performance Data Extract I/O

Accelerating Design of ML and AI Experiments in Scientific Simulation

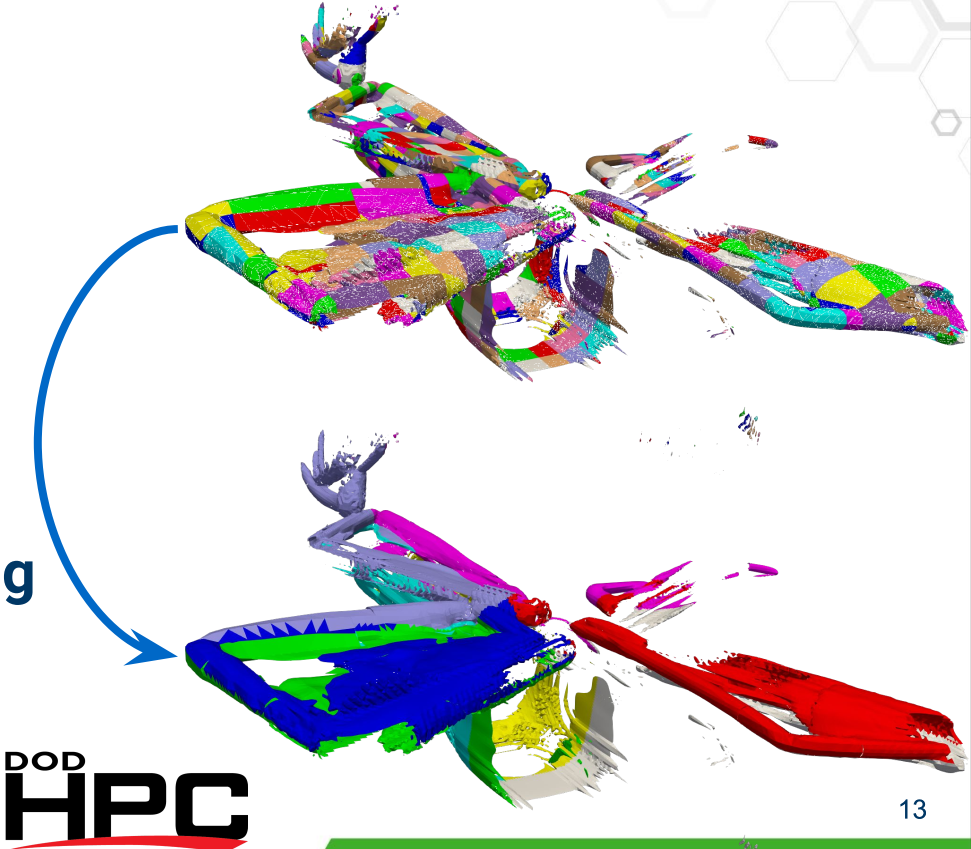
Just-Announced NASA SBIR Award

Coupling mature CFD codes to AI and ML tools



Automatic Repartitioning & Tuned I/O for Data Extracts

- Repartition data extract
- Aggregate extract data onto writer node ranks
- Write the extract to disk using a machine-specific write configuration



Catalyst 2.0 is available today!

https://gitlab.kitware.com/paraview/catalyst

Using VTK-m in ParaView and Catalyst

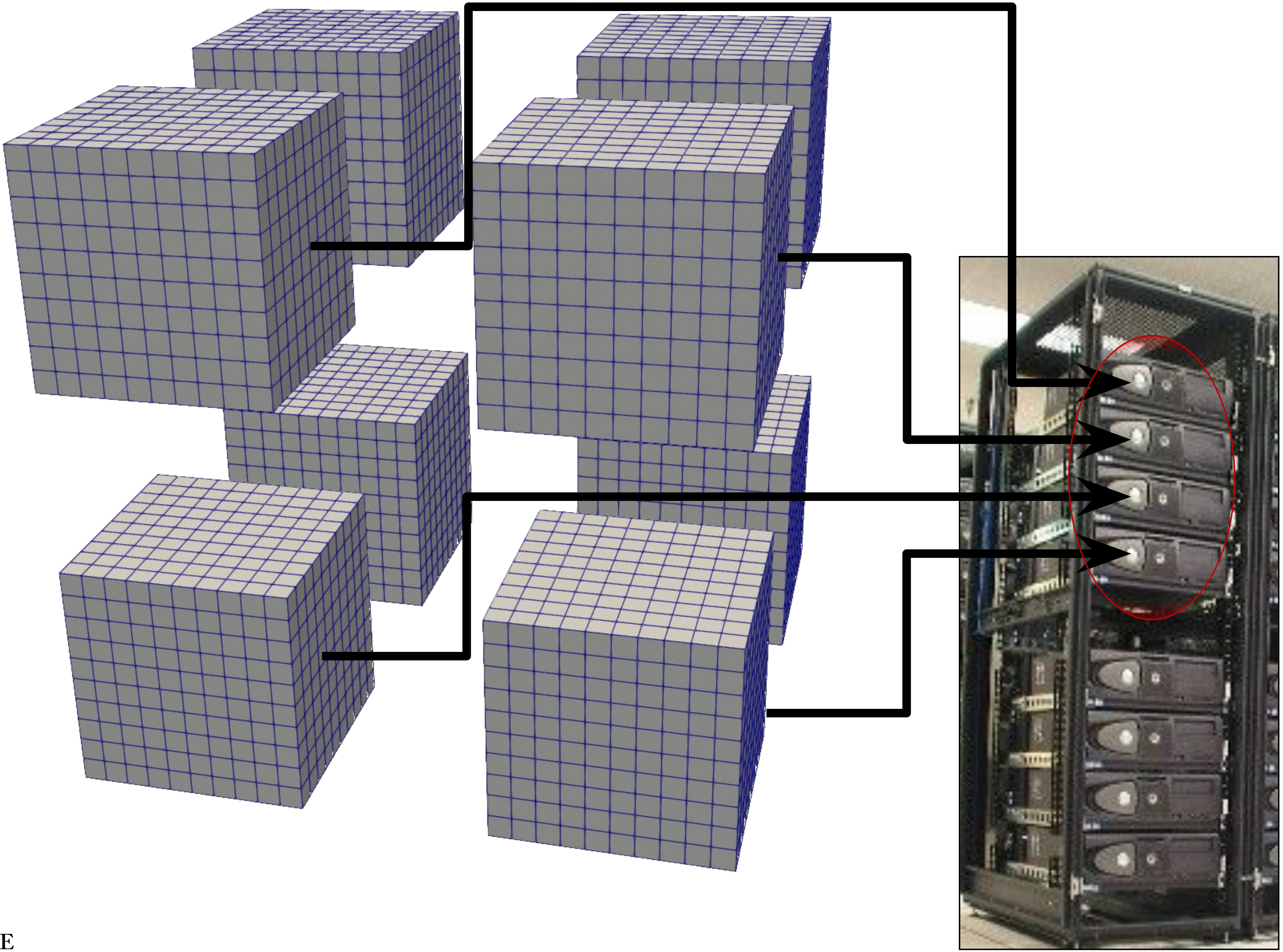
Kenneth Moreland, ORNL

ParaView and Catalyst
CASS Community BOF

June 12, 2024

ORNL is managed by UT-Battelle LLC for the US Department of Energy

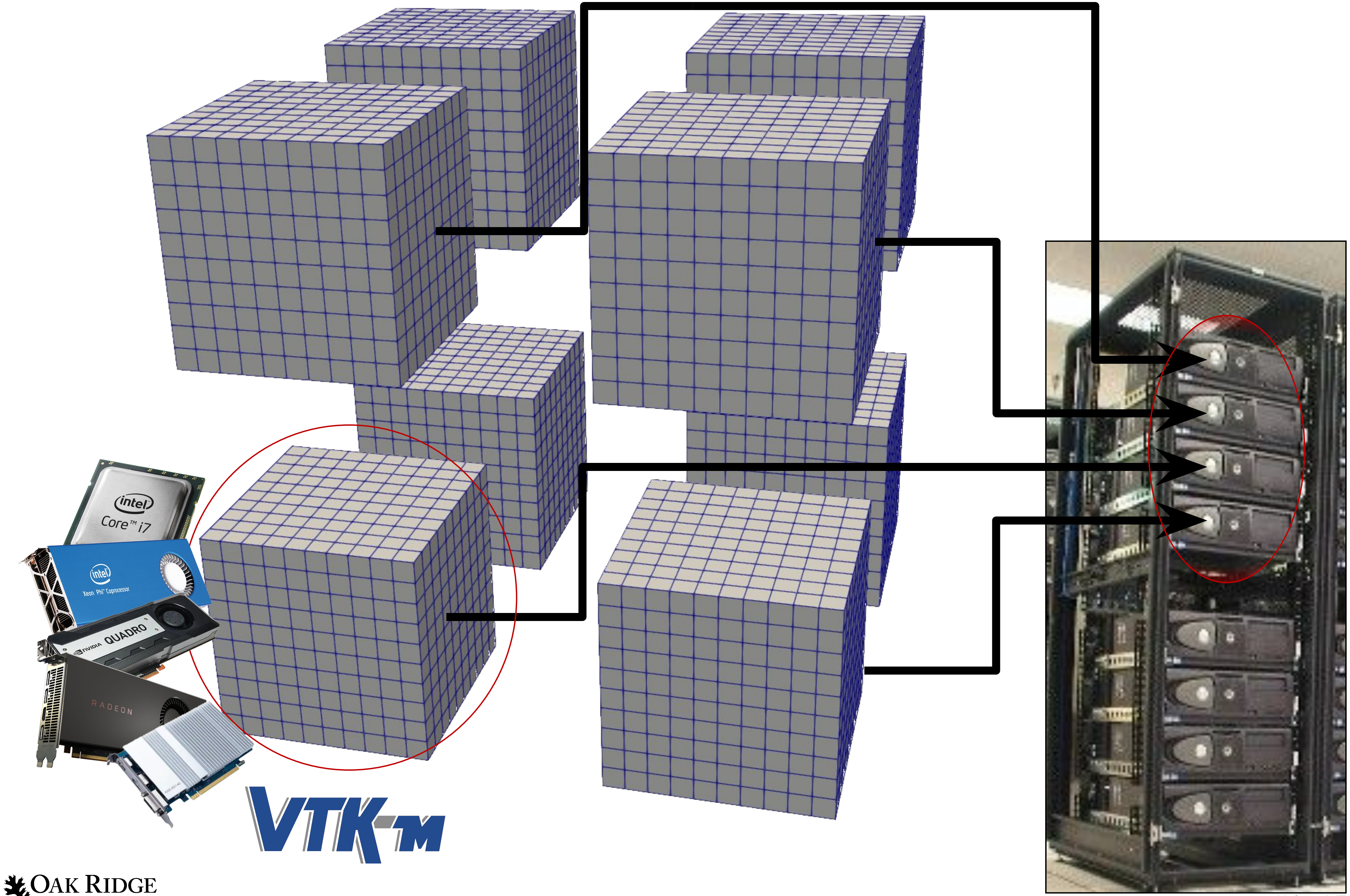
What is VTK-m?



Distributed Parallelism

ParaView

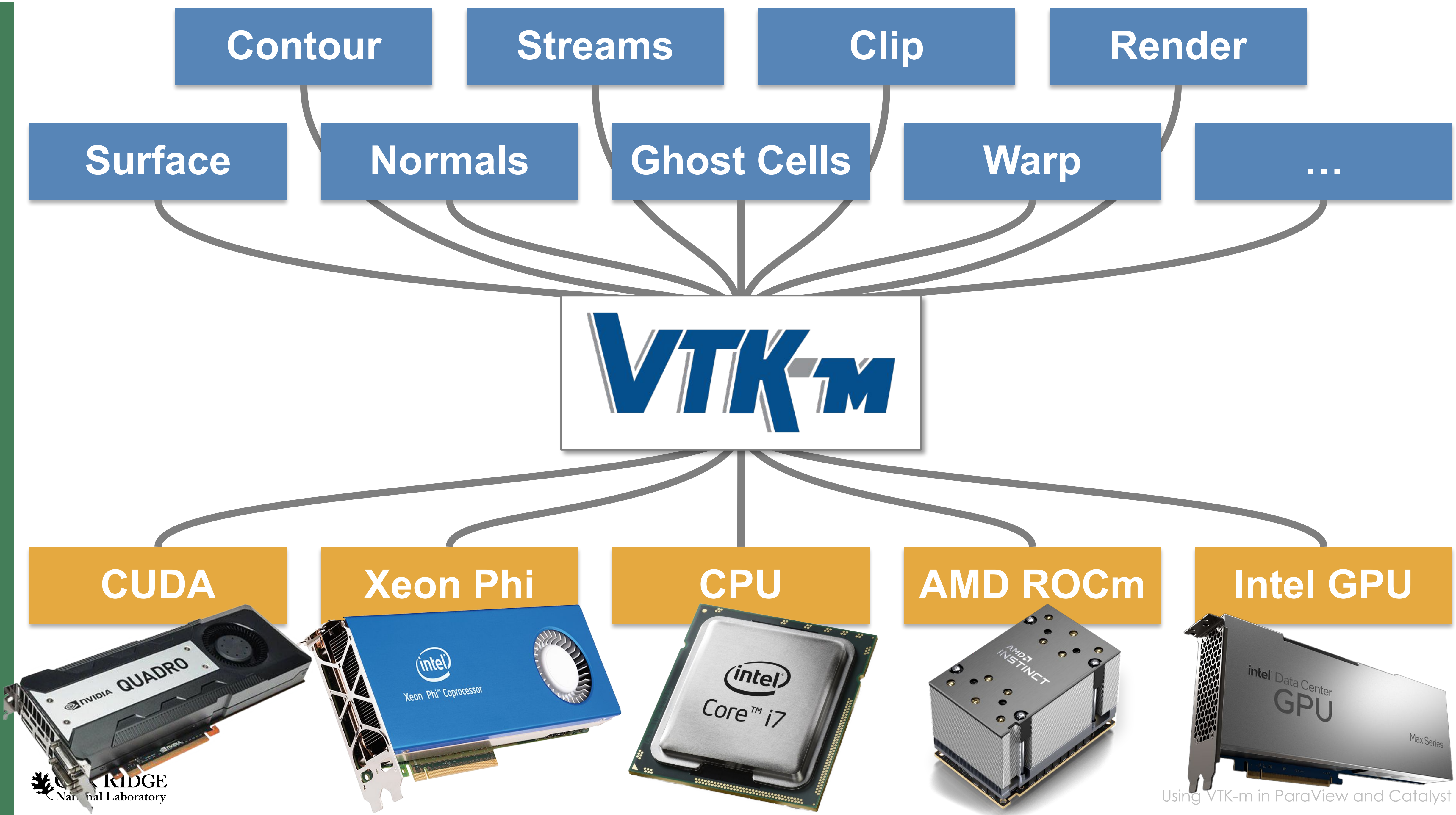
ParaView Catalyst



Distributed Parallelism

ParaView

ParaView Catalyst



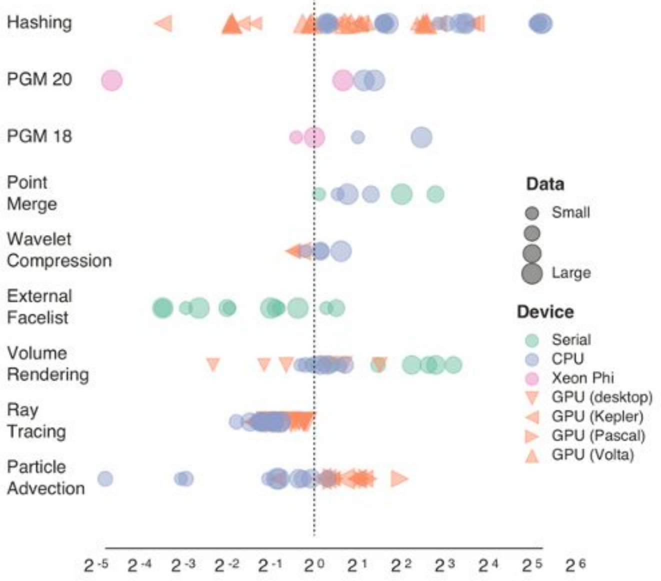
Contour
Streams
Clip
Render
Surface
Normals
Ghost Cells
Warp
…

VTK-m

CUDA
Xeon Phi
CPU
AMD ROCm
Intel GPU

VTK-m Demonstrates Performance Portability

- Science Problem
  - Leadership class facilities are using a variety of processor technologies.
    - Hardware for the Exascale Computing Project (ECP) takes advantage of processors from a variety of vendors — Intel, NVIDIA, AMD — each with their own programming language and API.
  - There are many scientific visualization algorithms that need to work well on these accelerator processors.
    - It is infeasible to update every algorithm for every device.
- Technical Solution
  - The VTK-m software framework provides abstractions that make it possible to implement an algorithm once and port it to any of the devices used by ECP.
  - A recent literature review of algorithms implemented in VTK-m and implemented for a particular hardware device show that the portable VTK-m implementation works as well as the specialized implementations.
- Science Impact
  - VTK-m makes it possible to support scientific visualization on the heterogeneous architectures of today's leadership class facilities.

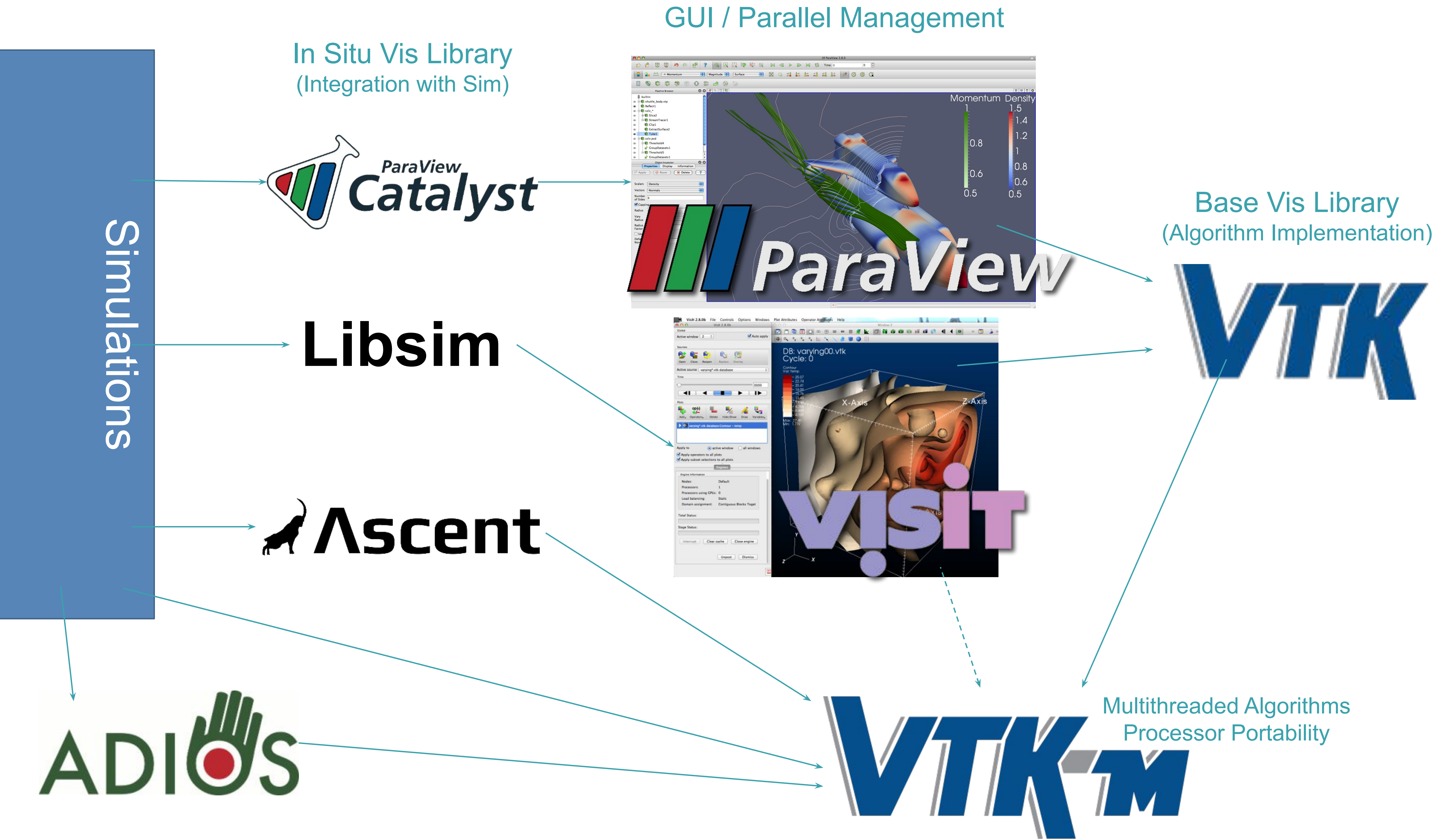


Algorithm	CPUs	GPUs	X. Phi	Serial	Total
External facelist	-	-	-	0.34	0.34
PGM 18	3.32	-	0.87	-	1.69
PGM 20	2.39	-	0.25	-	0.78
Particle advection	0.38	1.53	-	-	0.76
Point merge	1.82	-	-	3.10	2.38
Ray tracing	0.47	0.55	-	-	0.51
Volume rendering	1.13	0.83	-	3.10	1.43
Wavelet compression	1.13	0.75	-	-	0.92
Hashing	5.97	1.45	-	-	2.94
Total	1.45	0.95	0.47	1.48	**1.14**

This plot and table demonstrate the speedup from implementing a visualization algorithm in VTK-m vs. directly for a particular type of hardware. The geometric means indicate a speedup near 1, meaning that the VTK-m implementations, perform about the same as code optimized for a specific platform. From Moreland, et al. (2021). Minimizing Development Costs for Efficient Many-Core Visualization Using MCD^3. *Parallel Computing*, *108*(102834). doi:10.1016/j.parco.2021.102834

UNIVERSITY OF OREGON
OAK RIDGE National Laboratory
Kitware
EXASCALE COMPUTING PROJECT
Lawrence Livermore National Laboratory

Using VTK-m



GUI / Parallel Management

In Situ Vis Library
(Integration with Sim)

Base Vis Library
(Algorithm Implementation)

Simulations

ParaView Catalyst

ParaView

Libsim

VisIt

Ascent

VTK

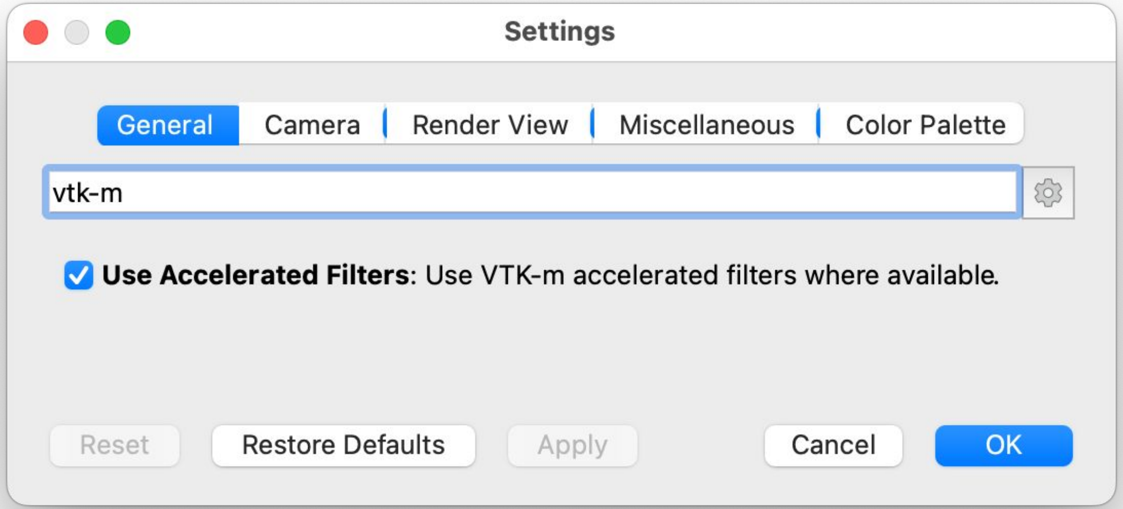
ADIOS

VTK-m

Multithreaded Algorithms
Processor Portability

Using VTK-m in ParaView

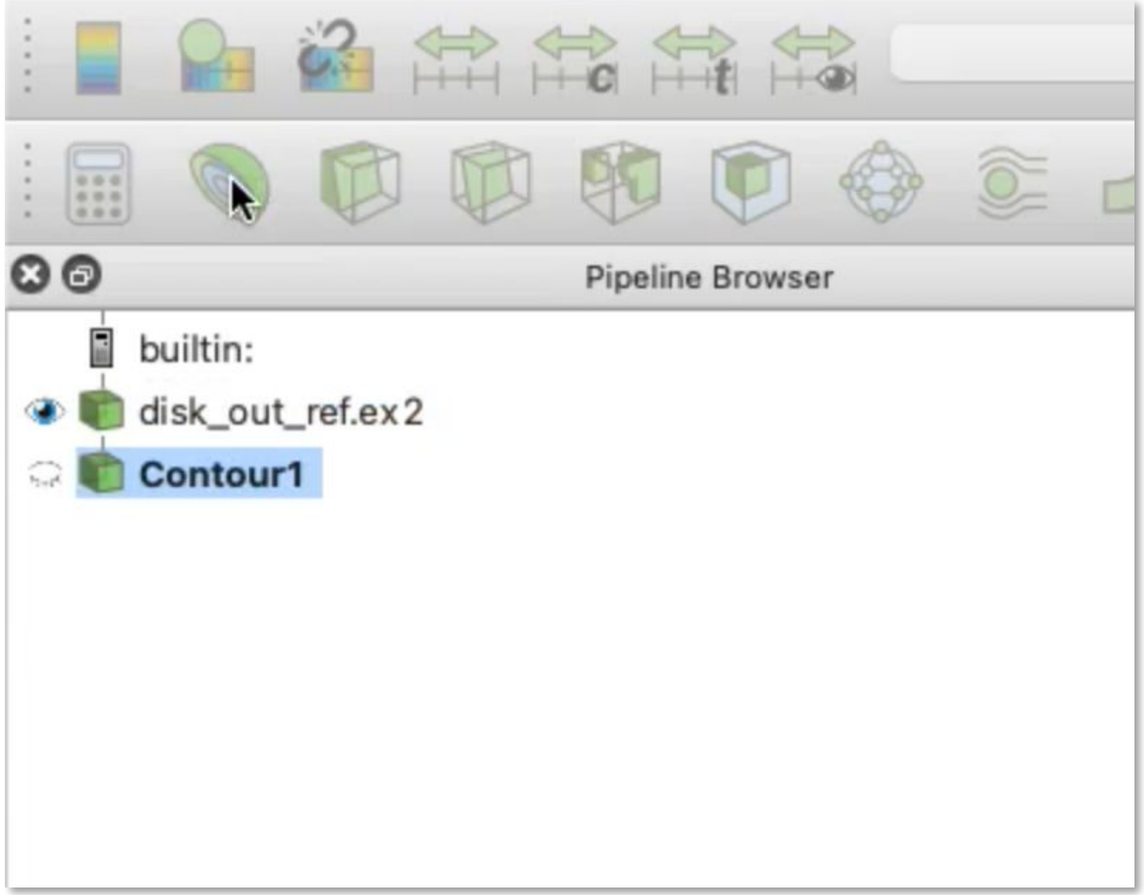
1. Turn on VTK-m in Preferences

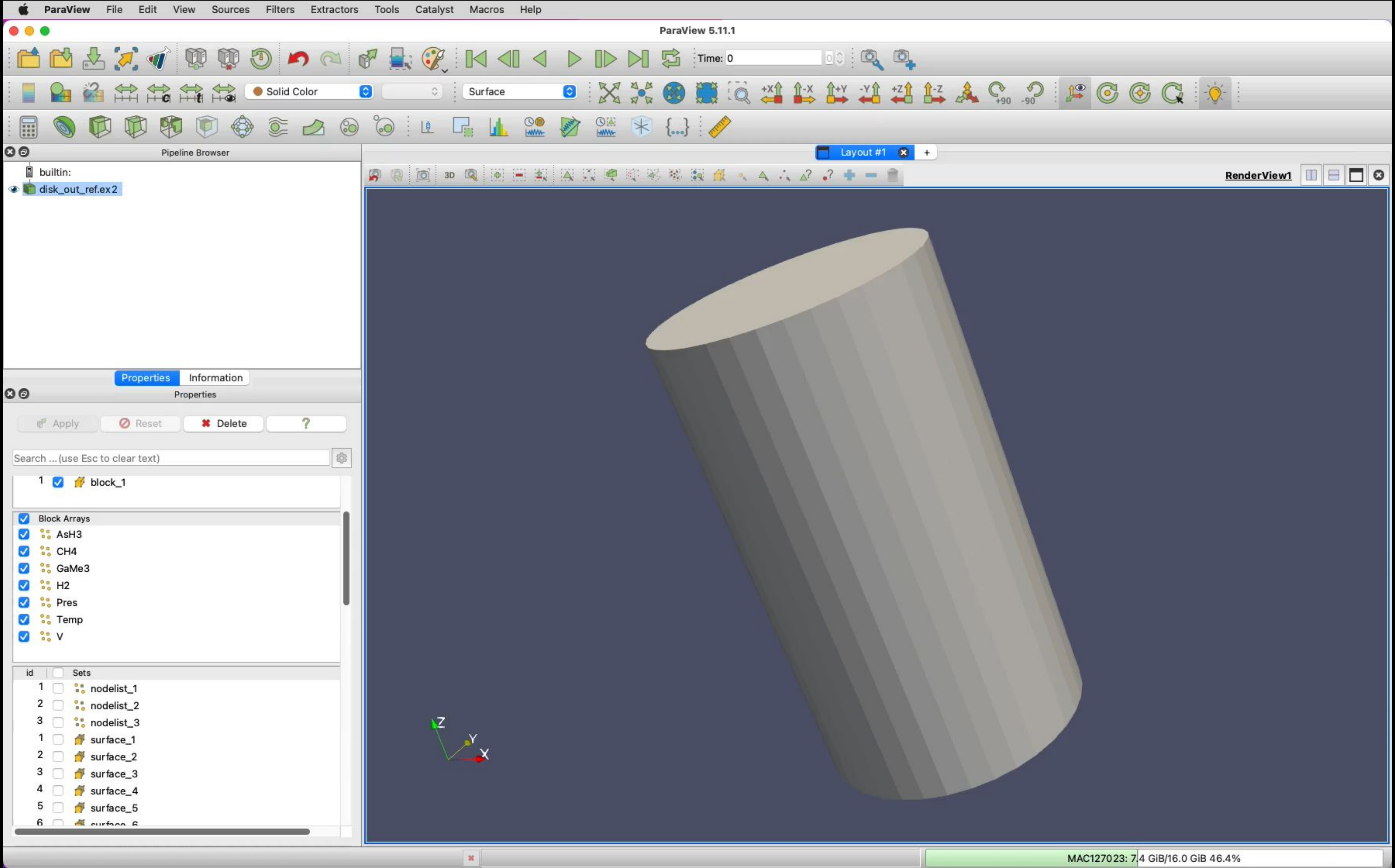


vtk-m

☑ **Use Accelerated Filters**: Use VTK-m accelerated filters where available.

2. Use VTK-m-enabled filters as normal





Time: 0

Solid Color

Surface

builtin:
disk_out_ref.ex2

1 ☑ block_1

☑ Block Arrays
☑ AsH3
☑ CH4
☑ GaMe3
☑ H2
☑ Pres
☑ Temp
☑ V

id	☐ Sets
1	☐ nodelist_1
2	☐ nodelist_2
3	☐ nodelist_3
1	☐ surface_1
2	☐ surface_2
3	☐ surface_3
4	☐ surface_4
5	☐ surface_5
6	☐ surface_6

MAC127023: 7.4 GiB/16.0 GiB 46.4%

Current and Future Status

- ParaView and Catalyst installed on Frontier include GPU support from VTK-m
  - Future installs on DOE systems will have similar capability
- Spack builds support VTK-m integration
- VTK-m will soon be rebranding as Viskores
  - Part of VTK-m joining the High Performance Software Foundation (HPSF)
  - Will not affect use of ParaView/Catalyst
  - Transition will be gradual to minimize impact on developers

https://hpsf.io/

Acknowledgements

- This work was supported by the U.S. Department of Energy (DOE) RAPIDS SciDAC project under contract number DE-AC05-00OR22725.
- This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative.
- This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Award Numbers 10-014707, 12-015215, and 14-017566.

- **Thanks to many, many partners in labs, universities, and industry.**

PARAVIEW CATALYST USE IN THE US DEPARTMENT OF DEFENSE

Andrew C. Bauer
Consortium for the Advanced Scientific Software (CASS) Birds-of-a-Feather: ParaView and Catalyst Session
June 12, 2024

U.S. ARMY
US Army Corps of Engineers®
ERDC
ENGINEER RESEARCH & DEVELOPMENT CENTER
DAAC
Data Analysis and Assessment Center

DOD HPCMP CREATE™-AV HELIOS

- Parallel code focused on hi-fidelity rotorcraft simulations
- Typically runs for a week or two for a single simulation (20k time steps is typical)
- User-base size
  - On the order of 1000 licenses (valid for all CREATE™-AV products)
  - On the order of 100 users

CREATE-AV

Helios

s CFD Solution by Navair, Sikorsky, CCDC-AvMC
Visualization by DoD HPC/DA

https://centers.hpc.mil/CREATE/CREATE-AV.html

COMPUTATIONAL AEROMECHANICS "HELIOS" SOFTWARE DEVELOPMENT

- **Technical goals/objectives**
  - HPCMP CREATE^{TM}-AV program has overseen the development of high fidelity fixed and rotary wing full vehicle prediction tools
  - Accurately predict aeromechanics of full-vehicle rotorcraft, reducing reliance on physical test and accelerating the speed of acquisition

H-47 tandem simulation
Courtesy Boeing

AH-64 Apache
Courtesy Boeing

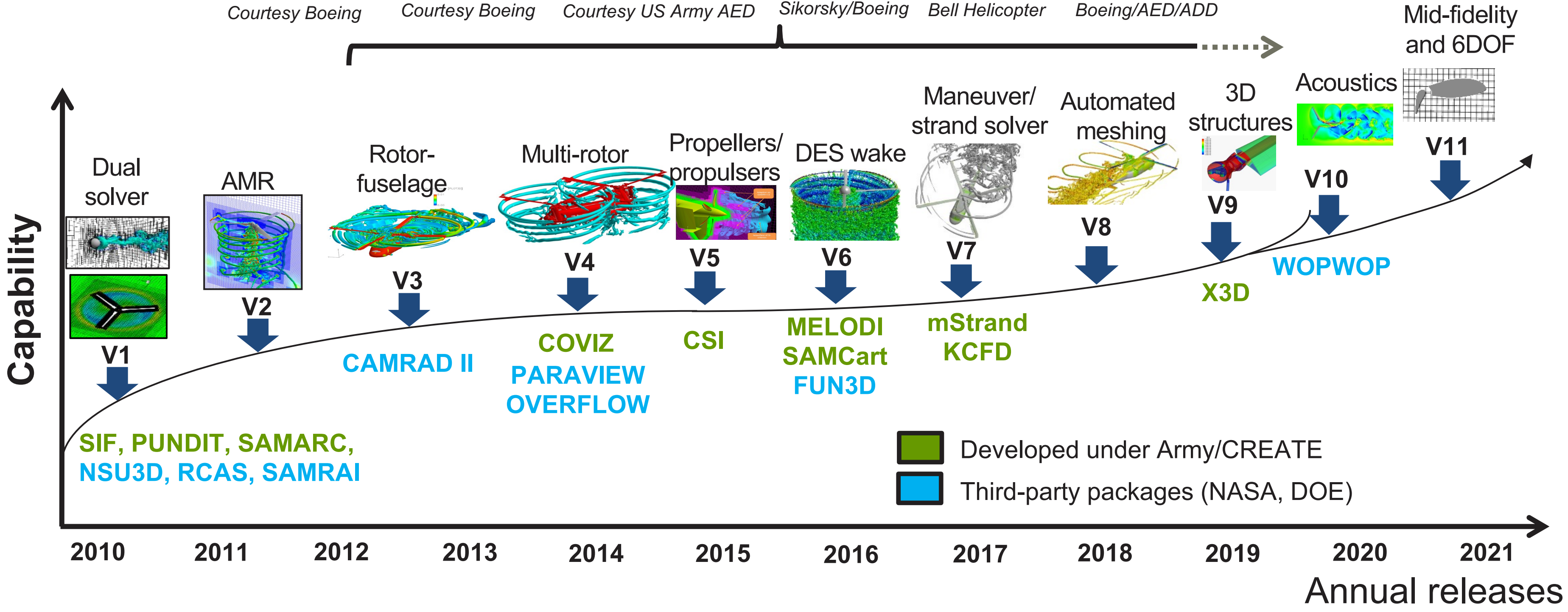
UH-60 main/tail simulation
Courtesy US Army AED

JMR "Defiant"
Sikorsky/Boeing

JMR "Valor"
Bell Helicopter

CH47 – Block II
Boeing/AED/ADD

Army programs using Helios

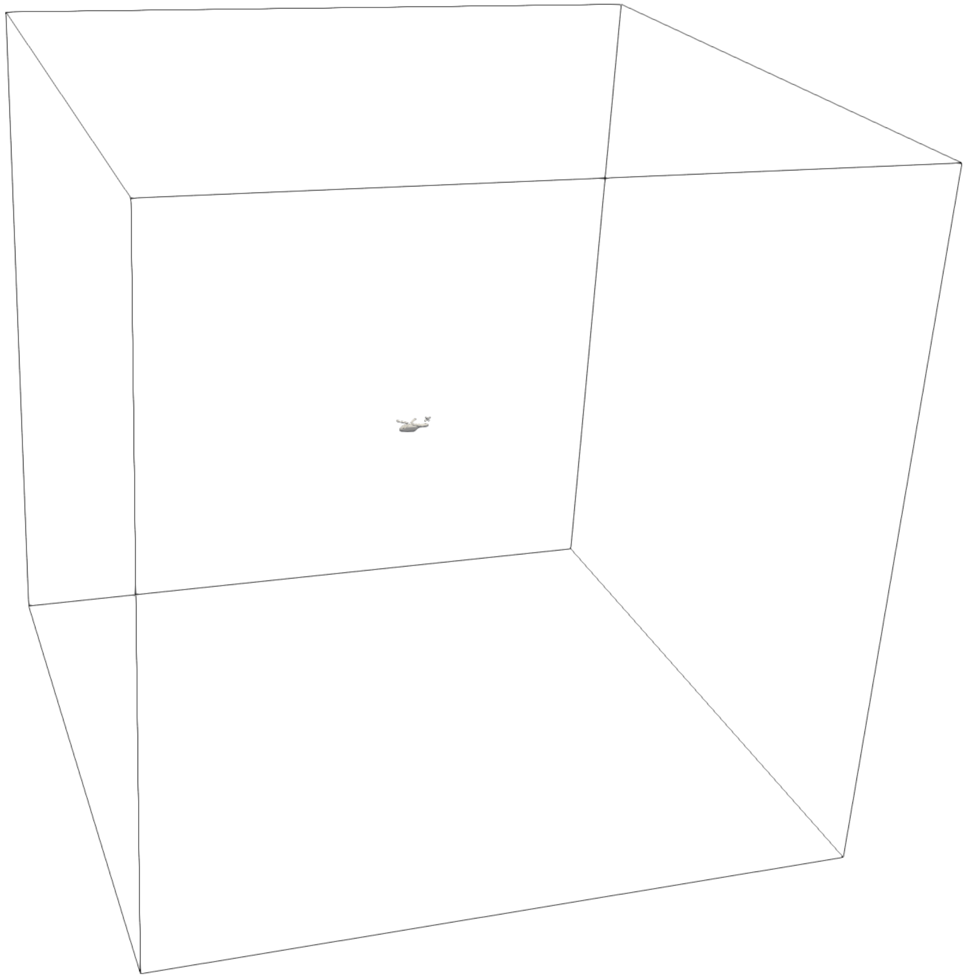


🟩 Developed under Army/CREATE
🟦 Third-party packages (NASA, DOE)

IN SITU FUNCTIONALITY

- Motivation
  - Full data dump was infeasible
    - File size was too big
    - Temporal resolution was too low
  - Region of interest is much smaller than the compute domain

- Solution
  - Production in situ use – Kitware SBIR on in situ processing
  - Output data extracts focused on information of interest



PRE-PROCESSING GUI

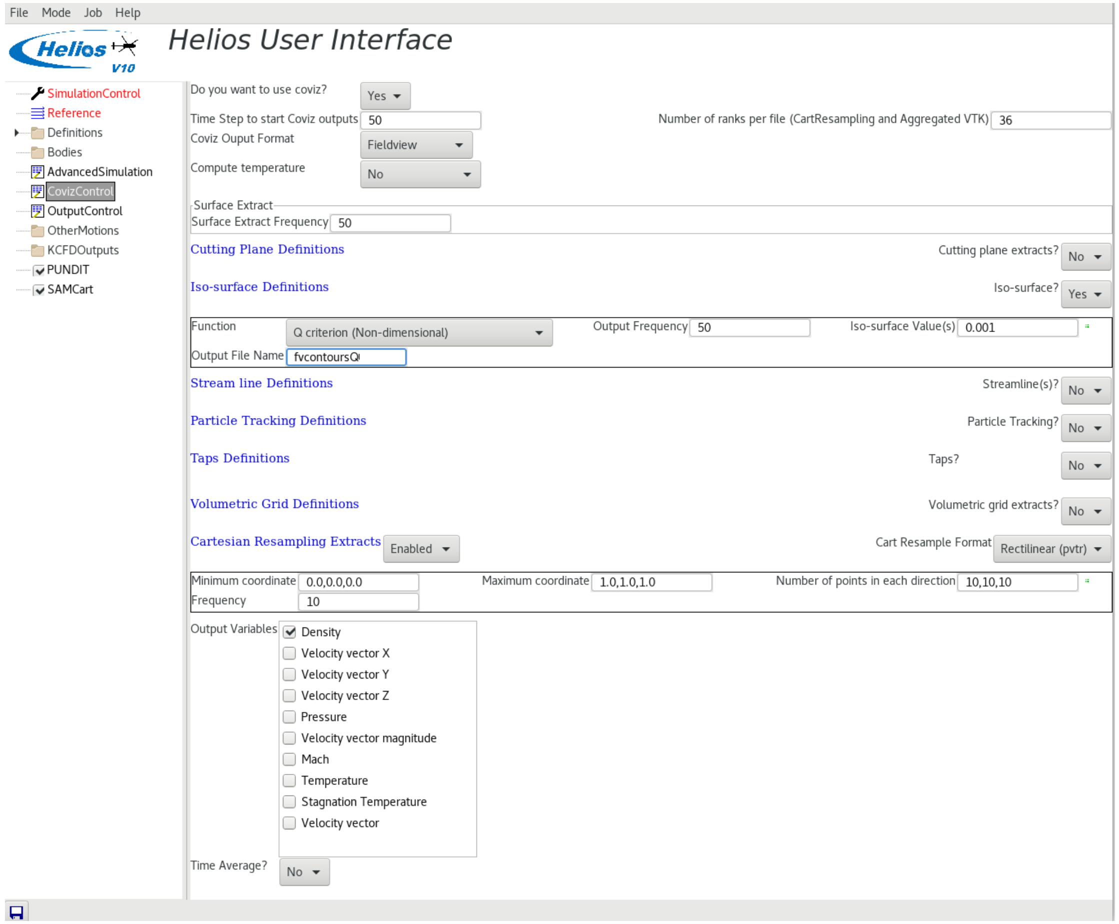
Tab for specifying in situ outputs

Global options

- When to start outputting data
- Output file format and settings
- Derived fields

Specific extracts

- Particle paths requires seed information, output frequency, file name
- Slices requires location and direction, offsets, output frequency, file name
- Contours requires field name, iso-surface values, offsets, output frequency, file name
- Surface extract requires output frequency, file name
- Streamlines requires seed information, output frequency, file name
- Taps requires seed information, output frequency, file name
- Cartesian extract requires geometry information, option to include time average, output frequency, file name
- Full data dump requires grid information, output frequency, file name



Do you want to use coviz? Yes

Time Step to start Coviz outputs 50

Number of ranks per file (CartResampling and Aggregated VTK) 36

Coviz Ouput Format Fieldview

Compute temperature No

Surface Extract Frequency 50

Cutting plane extracts? No

Iso-surface? Yes

Function Q criterion (Non-dimensional)

Output Frequency 50

Iso-surface Value(s) 0.001

Output File Name fvcontoursQ

Streamline(s)? No

Particle Tracking? No

Taps? No

Volumetric grid extracts? No

Cartesian Resampling Extracts Enabled

Cart Resample Format Rectilinear (pvtr)

Minimum coordinate 0.0,0.0,0.0

Maximum coordinate 1.0,1.0,1.0

Number of points in each direction 10,10,10

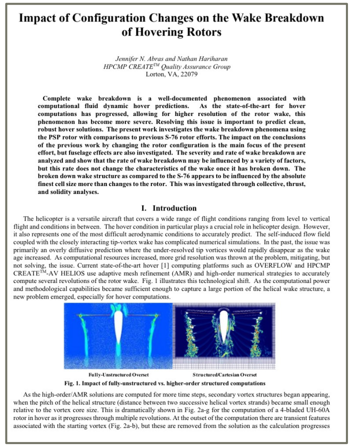
Frequency 10

Output Variables
- ☑ Density
- ☐ Velocity vector X
- ☐ Velocity vector Y
- ☐ Velocity vector Z
- ☐ Pressure
- ☐ Velocity vector magnitude
- ☐ Mach
- ☐ Temperature
- ☐ Stagnation Temperature
- ☐ Velocity vector

Time Average? No

DIVERSE USER BASE

- Users across academia, government and industry
- Can extract results in different formats
  - Helios workflow doesn't force users to learn a new post-processing tool
- Using coviz doesn't require detailed knowledge of the processing involved or knowledge of ParaView
- Coviz built into the normal workflow
- Work goals are research & development, design, safety analysis



Impact of Configuration Changes on the Wake Breakdown of Hovering Rotors

Jennifer N. Abras and Nathan Hariharan
HPCMP CREATE^{TM} Quality Assurance Group
Lorton, VA, 22079

Complete wake breakdown is a well-documented phenomenon associated with computational fluid dynamic hover predictions. As the state-of-the-art for hover computations has progressed, allowing for higher resolution of the rotor wake, this phenomenon has become more severe. Resolving this issue is important to predict clean, robust hover solutions. The present work investigates the wake breakdown phenomena using the PSP rotor with comparisons to previous S-76 rotor efforts. The impact on the conclusions of the previous work by changing the rotor configuration is the main focus of the present effort, but fuselage effects are also investigated. The severity and rate of wake breakdown are analyzed and show that the rate of wake breakdown may be influenced by a variety of factors, but this rate does not change the characteristics of the wake once it has broken down. The broken down wake structure as compared to the S-76 appears to be influenced by the absolute finest cell size more than changes to the rotor. This was investigated through collective, thrust, and solidity analyses.

I. Introduction

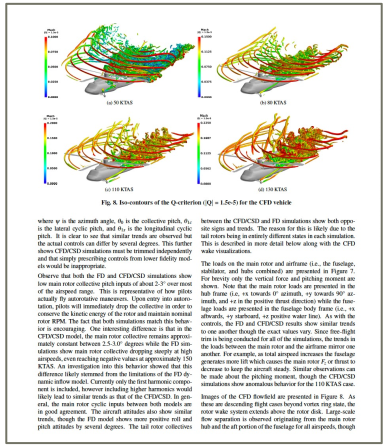
The helicopter is a versatile aircraft that covers a wide range of flight conditions ranging from level to vertical flight and conditions in between. The hover condition in particular plays a crucial role in helicopter design. However, it also represents one of the most difficult aerodynamic conditions to accurately predict. The self-induced flow field coupled with the closely interacting tip-vortex wake has complicated numerical simulations. In the past, the issue was primarily an overly diffusive prediction where the under-resolved tip vortices would rapidly disappear as the wake age increased. As computational resources increased, more grid resolution was thrown at the problem, mitigating, but not solving, the issue. Current state-of-the-art hover [1] computing platforms such as OVERFLOW and HPCMP CREATE^{TM}-AV HELIOS use adaptive mesh refinement (AMR) and high-order numerical strategies to accurately compute several revolutions of the rotor wake. Fig. 1 illustrates this technological shift. As the computational power and methodological capabilities became sufficient enough to capture a large portion of the helical wake structure, a new problem emerged, especially for hover computations.

Fully-Unstructured Overset | Structured/Cartesian Overset

Fig. 1. Impact of fully-unstructured vs. higher-order structured computations

As the high-order/AMR solutions are computed for more time steps, secondary vortex structures began appearing, when the pitch of the helical structure (distance between two successive helical vortex strands) became small enough relative to the vortex core size. This is dramatically shown in Fig. 2a-g for the computation of a 4-bladed UH-60A rotor in hover as it progresses through multiple revolutions. At the outset of the computation there are transient features associated with the starting vortex (Fig. 2a-b), but these are removed from the solution as the calculation progresses

Abras J., and Hariharan N., "Impact of Configuration Changes on the Wake Breakdown of Hovering Rotors," VFS Aeromechanics for Advanced Vertical Flight Technical Meeting, San Jose, CA, January 21-23, 2020



(a) 50 KTAS (b) 80 KTAS
(c) 110 KTAS (d) 130 KTAS

Fig. 8. Iso-contours of the Q-criterion ($|Q| = 1.5e\text{-}5$) for the CFD vehicle

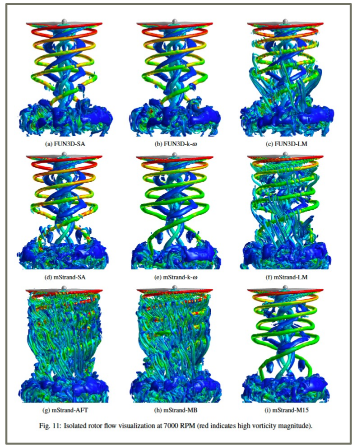
where ψ is the azimuth angle, θ_0 is the collective pitch, θ_{1c} is the lateral cyclic pitch, and θ_{1s} is the longitudinal cyclic pitch. It is clear to see that similar trends are observed but the actual controls can differ by several degrees. This further shows CFD/CSD simulations must be trimmed independently and that simply prescribing controls from lower fidelity models would be inappropriate.

Observe that both the FD and CFD/CSD simulations show low main rotor collective pitch inputs of about 2-3° over most of the airspeed range. This is representative of how pilots actually fly autorotative maneuvers. Upon entry into autorotation, pilots will immediately drop the collective in order to conserve the kinetic energy of the rotor and maintain nominal rotor RPM. The fact that both simulations match this behavior is encouraging. One interesting difference is that in the CFD/CSD model, the main rotor collective remains approximately constant between 2.5-3.0° degrees while the FD simulations show main rotor collective dropping steeply at high airspeeds, even reaching negative values at approximately 150 KTAS. An investigation into this behavior showed that this difference likely stemmed from the limitations of the FD dynamic inflow model. Currently only the first harmonic component is included, however including higher harmonics would likely lead to similar trends as that of the CFD/CSD. In general, the main rotor cyclic inputs between both models are in good agreement. The aircraft attitudes also show similar trends, though the FD model shows more positive roll and pitch attitudes by several degrees. The tail rotor collectives between the CFD/CSD and FD simulations show both opposite signs and trends. The reason for this is likely due to the tail rotors being in entirely different states in each simulation. This is described in more detail below along with the CFD wake visualizations.

The loads on the main rotor and airframe (i.e., the fuselage, stabilator, and hubs combined) are presented in Figure 7. For brevity only the vertical force and pitching moment are shown. Note that the main rotor loads are presented in the hub frame (i.e, +x towards 0° azimuth, +y towards 90° azimuth, and +z in the positive thrust direction) while the fuselage loads are presented in the fuselage body frame (i.e., +x aftwards, +y starboard, +z positive water line). As with the controls, the FD and CFD/CSD results show similar trends to one another though the exact values vary. Since free-flight trim is being conducted for all of the simulations, the trends in the loads between the main rotor and the airframe mirror one another. For example, as total airspeed increases the fuselage generates more lift which causes the main rotor F_z or thrust to decrease to keep the aircraft steady. Similar observations can be made about the pitching moment, though the CFD/CSD simulations show anomalous behavior for the 110 KTAS case.

Images of the CFD flowfield are presented in Figure 8. As these are descending flight cases beyond vortex ring state, the rotor wake system extends above the rotor disk. Large-scale flow separation is observed originating from the main rotor hub and the aft portion of the fuselage for all airspeeds, though

Tran S., S and Lopez M., "Full Vehicle Simulations of a Helicopter in Steady Autorotation", VFS Forum 80, Montreal, Canada, May 7-9, 2024.



(a) FUN3D-SA (b) FUN3D-k-ω (c) FUN3D-LM
(d) mStrand-SA (e) mStrand-k-ω (f) mStrand-LM
(g) mStrand-AFT (h) mStrand-MB (i) mStrand-M15

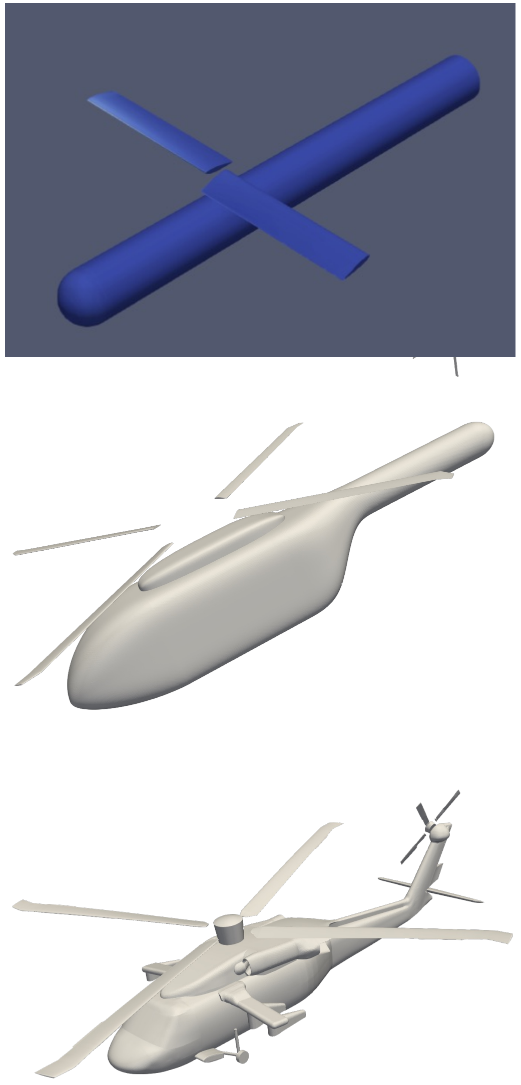
Fig. 11: Isolated rotor flow visualization at 7000 RPM (red indicates high vorticity magnitude).

Thai, A., Jain, R., and Grace, S., "CFD Validation of Small Quadrotor Performance using CREATETM-AV Helios," VFS Forum 75 Philadelphia, Pennsylvania, May 13–16, 2019.

ENGINEERING WORKFLOWS

In situ capabilities a natural fit for rotorcraft engineering workflows

- Users are running Helios many times on very similar problems
  - Parameter sweeps of functionality options (e.g. speed, turbulence model, angle of attack)
  - One-off geometries
- Physics is similar (CFD, CFD/CSD)
- Commonality of geometries
- Users know what they hope to get out of a simulation



IN SITU PARADIGM ADOPTION

In situ workflow provides a better option for the user base

- Augments full dump data output
- Production needs – it just works
- Requires restart capability (particle pathlines and Cartesian grid extract with time averaging)
- Requires stability
- Requires ease of use
  - In situ processing is regularly used by a significant portion of the user-base
- Treated as first class citizen
  - Included as part of the full workflow
  - Code and user documentation
  - Automated and QA testing

CONNECT WITH US

Andrew C. Bauer
Research Mechanical Engineer
DAAC/Information Technology Lab
U.S. Army Engineer Research and Development Center
U.S. Army Corps of Engineers
andrew.c.bauer@usace.army.mil
daac.hpc.mil

LinkedIn

YouTube

DISCOVER
DEVELOP
DELIVER
new ways to make the world safer and better every day
POWER OF ERDC PODCAST

BEYOND the GATES
How ERDC makes the difference

Scan this QR code with your phone for instant access

U.S. ARMY

US Army Corps of Engineers®

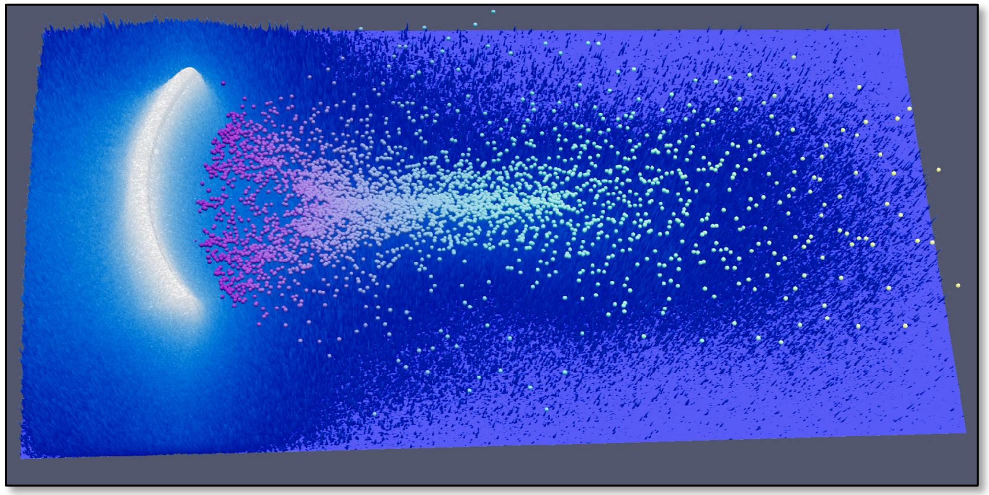
ERDC
ENGINEER RESEARCH & DEVELOPMENT CENTER

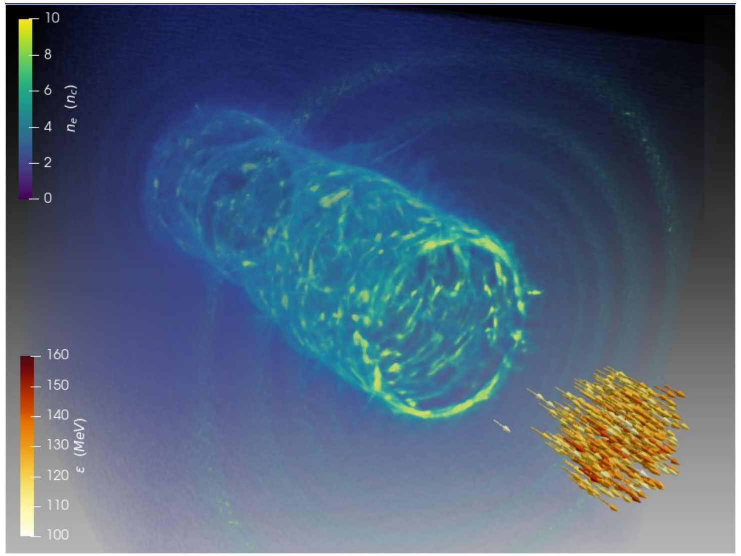
DAAC
Data Analysis and Assessment Center

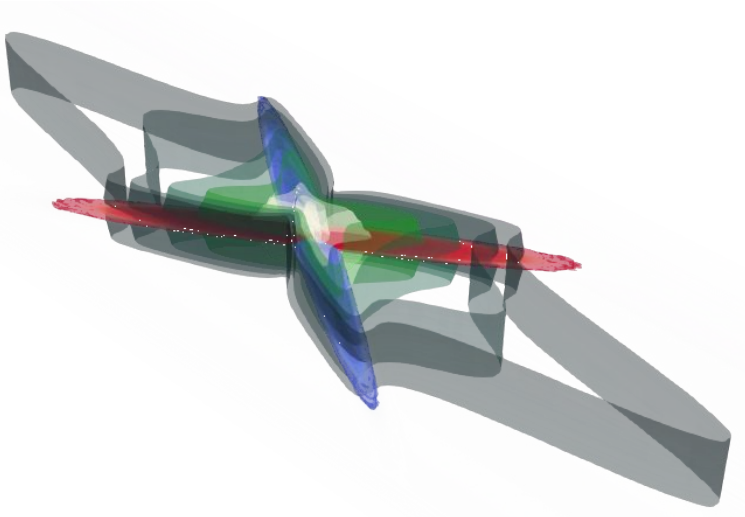
ParaView in High Energy Physics and Fusion Applications

Axel Huebl, Arianna Formenti, Marco Garten, Remi Lehe, Luca Fedeli, Jean-Luc Vay et al.
On behalf of the WarpX Collaboration at LBNL, LLNL, SLAC, CEA, DESY, TAE, CERN and beyond

Axel Huebl, Arianna Formenti, Marco Garten, Remi Lehe, Luca Fedeli, Jean-Luc Vay et al.
On behalf of the WarpX Collaboration at LBNL, LLNL, SLAC, CEA, DESY, TAE, CERN and beyond







Wednesday June 12, 2024
CASS Community BOF Days

Special thanks to Ascent teams
Cyrus Harrison, Matt Larsen, Nicole Marsaglia et al.
and ParaView teams
Berk Geveci, Abhishek Yenpure et al.

BERKELEY LAB

Advanced Modeling Program
ACCELERATOR TECHNOLOGY & APPLIED PHYSICS DIVISION ATAP

U.S. DEPARTMENT OF ENERGY | Office of Science

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

Developed by an international, multidisciplinary team

forming open governance

open source initiative®

THE LINUX FOUNDATION

HIGH PERFORMANCE SOFTWARE FOUNDATION

BERKELEY LAB

ATAP
ACCELERATOR TECHNOLOGY & APPLIED PHYSICS DIVISION

AMCRD

NERSC

Jean-Luc Vay

Ji Qiang

Arianna Forment

Marco Garten

Axel Huebl

Rémi Lehe

Chad Mitchell

Ryan Sandberg

Olga Shapoval

Edoardo Zoni

UR LLE

Kale Weichmann

Ann Almgren

Kevin Gott

Junmin Gu

Revathi Jambunathan

Andrew Myers

Weiqun Zhang

David Grote

Justin Angus

HELION

Eric Clark

Germany

DESY.

Maxence Thévenet

Severin Diederichs

Alexander Sinn

Ángel Ferran Pousa

Rob Shalloo

France

loa

Igor Andriyash

Switzerland

CERN

Lorenzo Giacomel

SLAC

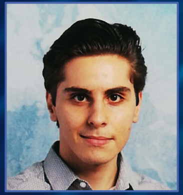
Lixin Ge

France

DE LA RECHERCHE À L'INDUSTRIE
cea
PARIS-SACLAY

Henri Vincenti

Luca Fedeli


Thomas Clark

Pierre Bartoli

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

Franz Poeschel

tae TECHNOLOGIES

Roelof Groenewald

AVALANCHE

MODERN ELECTRON

***over 80* contributors, incl. from the private sector**

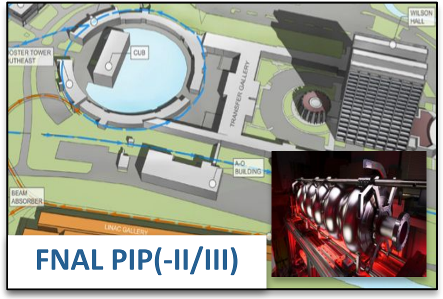
Science Drivers & Applications

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

High Energy Physics: CAMPA

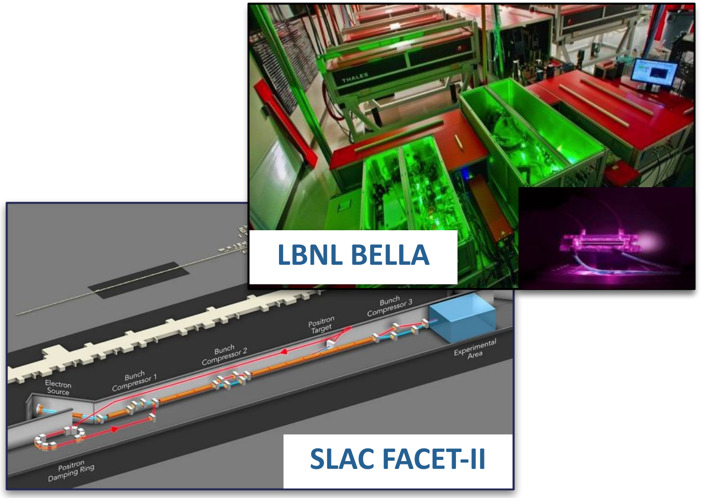
Collaboration for Advanced Modeling of Particle Accelerators

Radio-Frequency Accelerators



FNAL PIP(-II/III)

Plasma Accelerators: Laser- or Beam-Driven



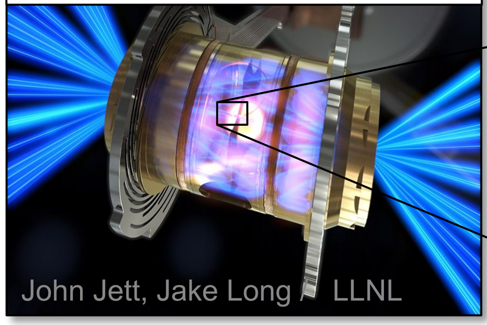
LBNL BELLA

SLAC FACET-II

Fusion Energy Sciences: KISMET

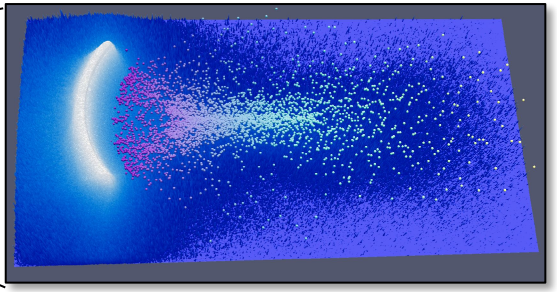
Kinetic IFE Simulations at Multiscale with Exascale Technologies

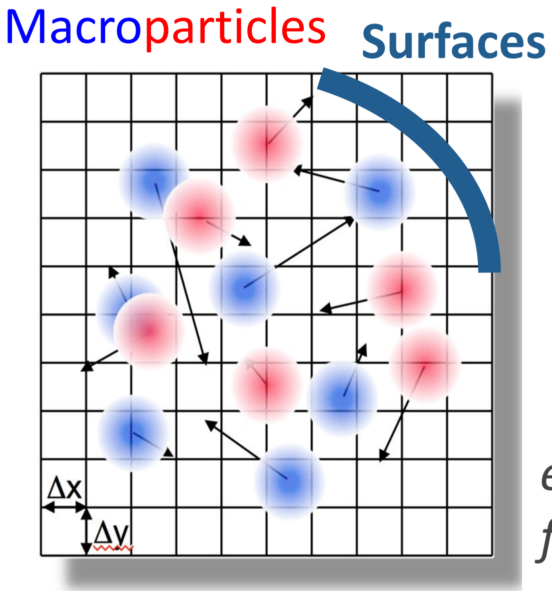
Kinetic effects in Inertial Confinement Fusion



John Jett, Jake Long / LLNL

WarpX on Frontier - *ParaView*
A Huebl, M Garten, J-L Vay, J Ludwig, S Wilks, A Kemp





electromagnetic (EM) fields on a grid

Kinetic effects on many space- and time scales to cover:

- From **<μm** (e.g., plasma structures) **to km** (e.g., accelerator)
- from **fs** (laser-plasma interaction) **to >seconds** (lifetime)

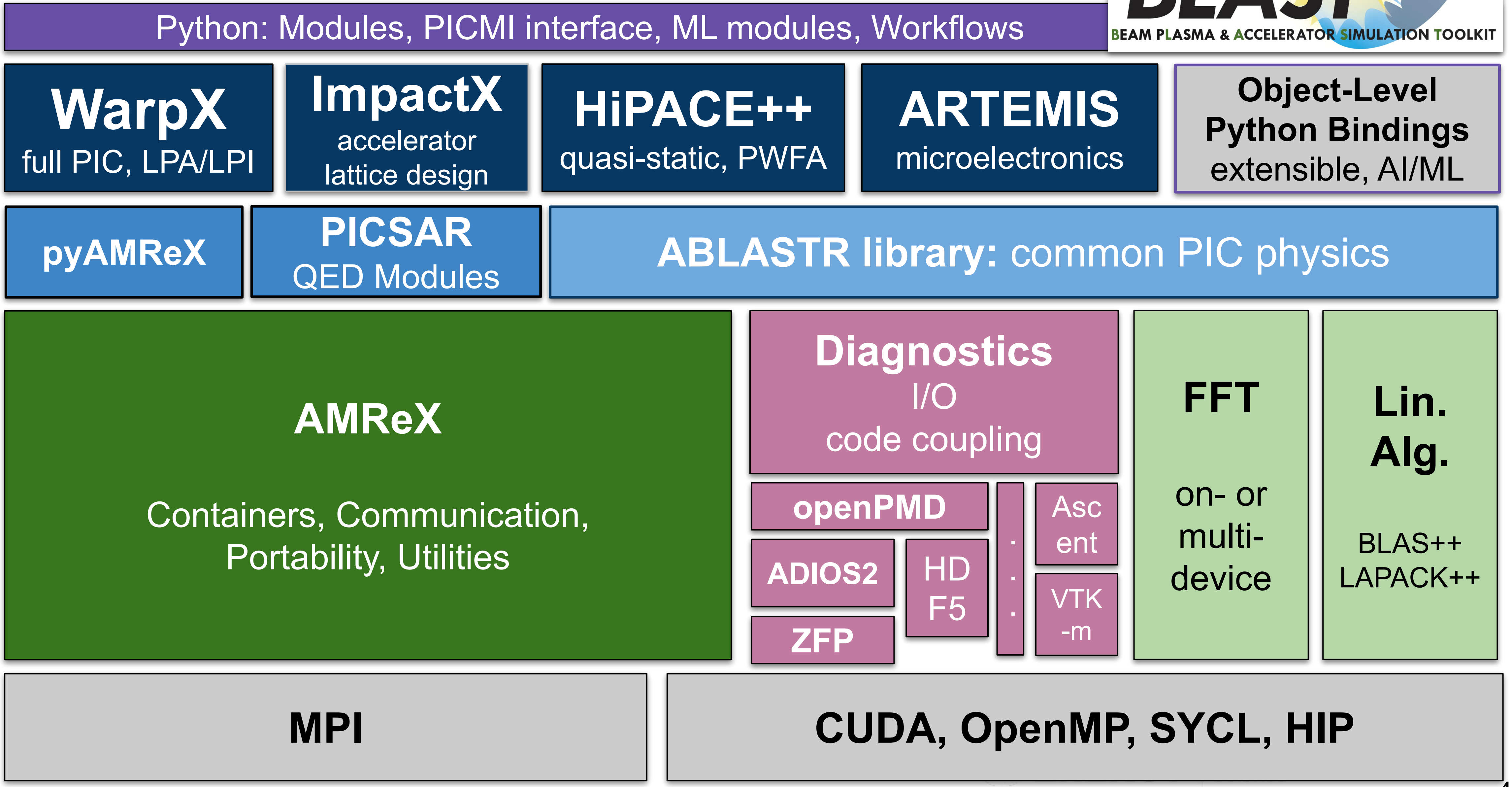
⇒ needs best algorithms on largest & fastest supercomputers

U.S. DEPARTMENT OF ENERGY
Office of Science

SciDAC
Scientific Discovery through Advanced Computing

Partnerships are Essential for Us

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT



Python: Modules, PICMI interface, ML modules, Workflows

WarpX
full PIC, LPA/LPI

ImpactX
accelerator lattice design

HiPACE++
quasi-static, PWFA

ARTEMIS
microelectronics

Object-Level Python Bindings
extensible, AI/ML

pyAMReX

PICSAR
QED Modules

ABLASTR library: common PIC physics

AMReX
Containers, Communication, Portability, Utilities

Diagnostics
I/O
code coupling

openPMD
ADIOS2
HDF5
ZFP
...
Ascent
VTK-m

FFT
on- or multi-device

Lin. Alg.
BLAS++
LAPACK++

MPI

CUDA, OpenMP, SYCL, HIP

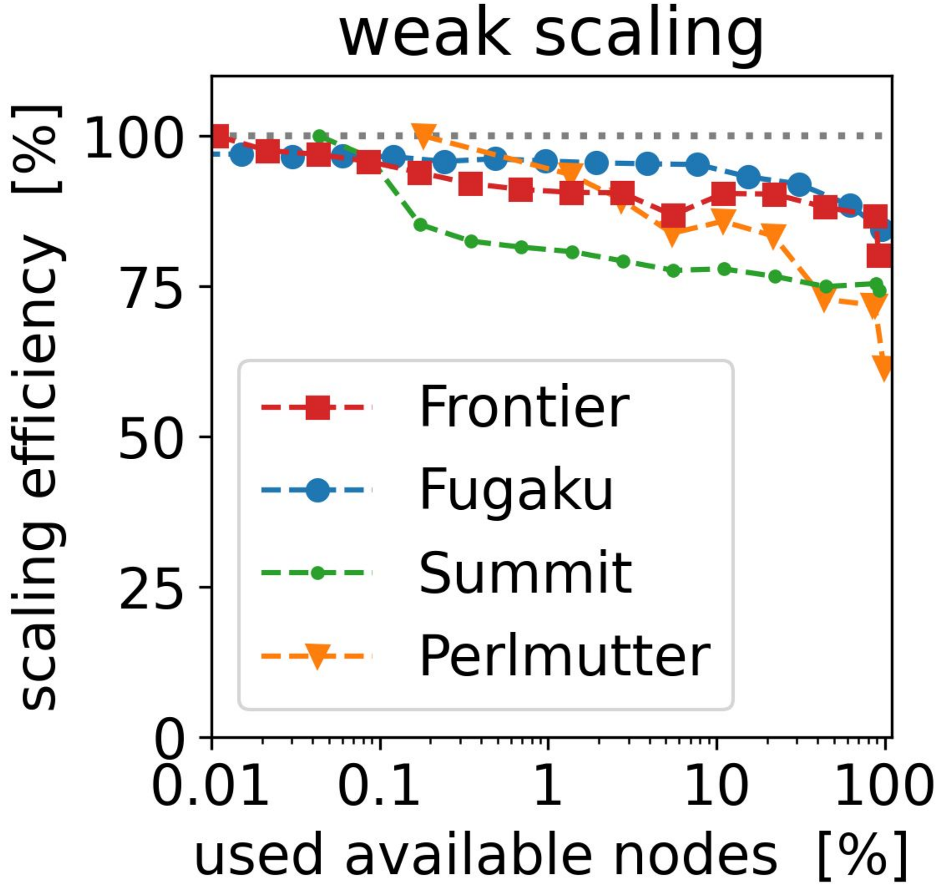
mac OS

Desktop to HPC

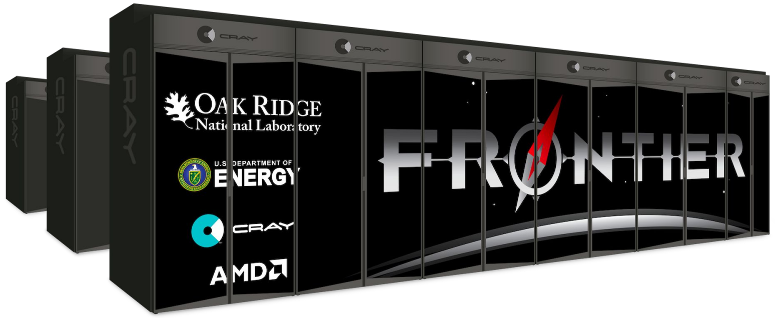
WarpX is the first Particle-in-Cell Code to Win the ACM Gordon Bell

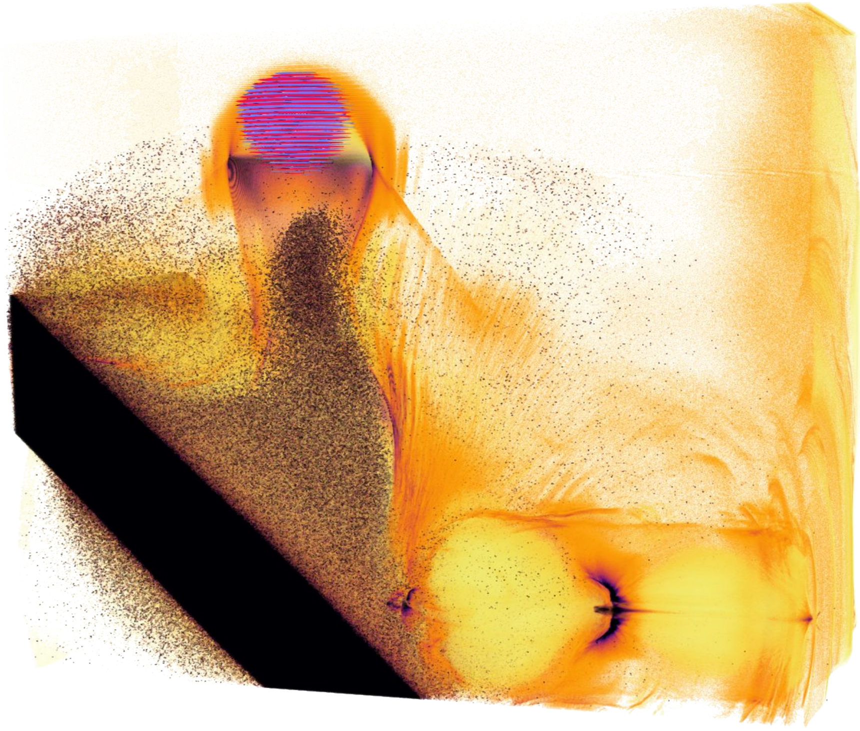
April-July 2022: WarpX on **world's largest HPCs**

L. Fedeli, A. Huebl et al., *Gordon Bell Prize Winner* in SC'22, 2022



Note: Perlmutter & Frontier in pre-acceptance at the time!





WarpX - *ParaView*

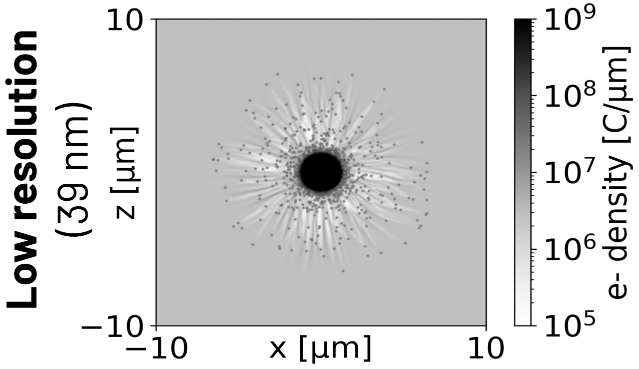
L. Fedeli (2022)

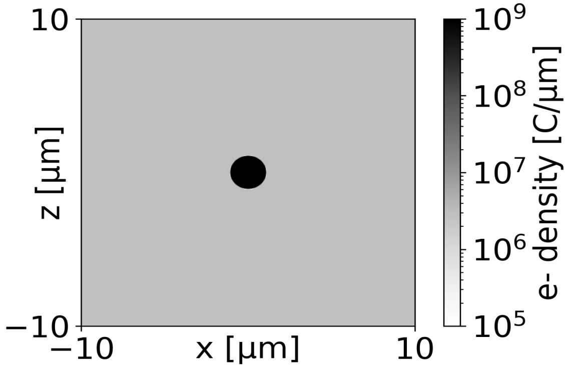
High resolution
(26 nm)

Through Visualization, we want to Stimulate & Develop Insight

Selected *visual* questions we want to address *in situ*

- **Physics:** Which effects of scale & dimensionality are overlooked in lower fidelity?
- **Dynamics:** Is a (costly) simulation evolving as anticipated?
- **Analytics:** What is the response on a (virtual) detector?
- **Correctness:** Are numerical options and resolution sufficient & stable?





L. Fedeli, A. Huebl et al., SC’22, 2022

  - ○ Are any hardware or software issues/bugs appearing at scale?

Traditional Post-Processing: Laser-Ion Acceleration

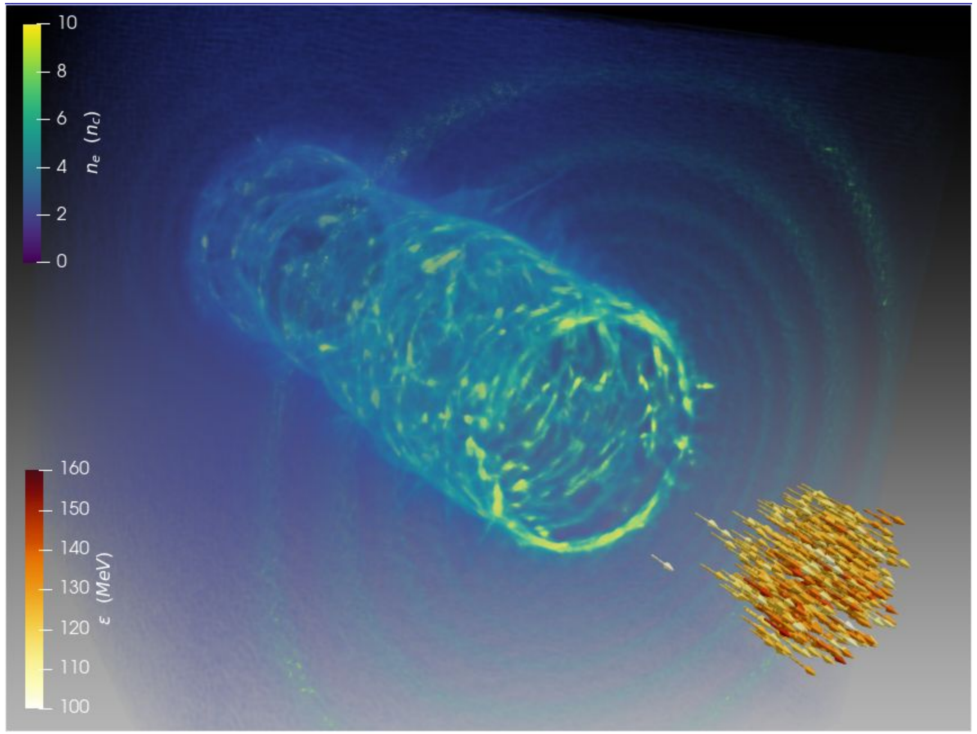
Workflow

- WarpX run with ***multiple outputs***
  - **openPMD schema**: ADIOS1→2, HDF5, …
- **"Vis data output"**: *down-sampled field* and particle data
- **Post-processing** with ParaView

www.openPMD.org

- Open Particle Mesh Data Standard[1]
- 18 simulation codes, 11 analysis frameworks, reference API, staging/streaming[2]

WarpX on Perlmutter - *ParaView*
A Huebl, M Garten (2023)



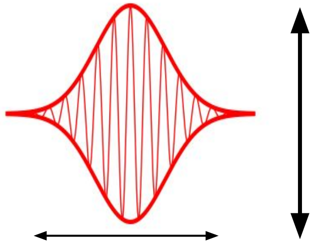
electron density of the plasma booster stage (left) and energy-boosted proton beam (right) from our 3D particle-in-cell simulations[3]

---

[1] A Huebl et al., "openPMD: A meta data standard for particle and mesh based data," DOI:10.5281/zenodo.591699 (2015)
[2] F Poeschel et al., "Transitioning from File-Based HPC Workflows to Streaming Data Pipelines with openPMD & ADIOS2", SMC 2021 (2022)
[3] M Garten et al., "Laser-Plasma Ion Beam Booster Based on Hollow-Channel Magnetic Vortex Acceleration", *accepted* (2024)

Traditional Post-Processing: Inertial Fusion Science

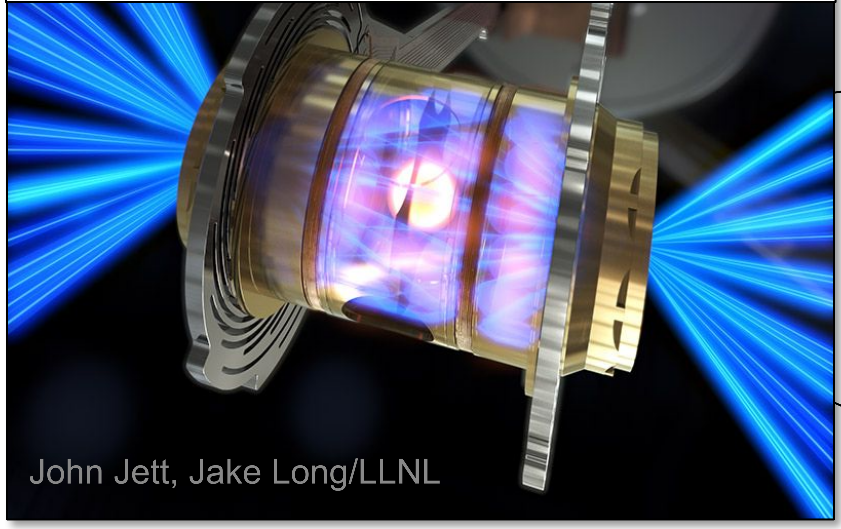
10^{14}W to 10^{16}W



τ=30fs

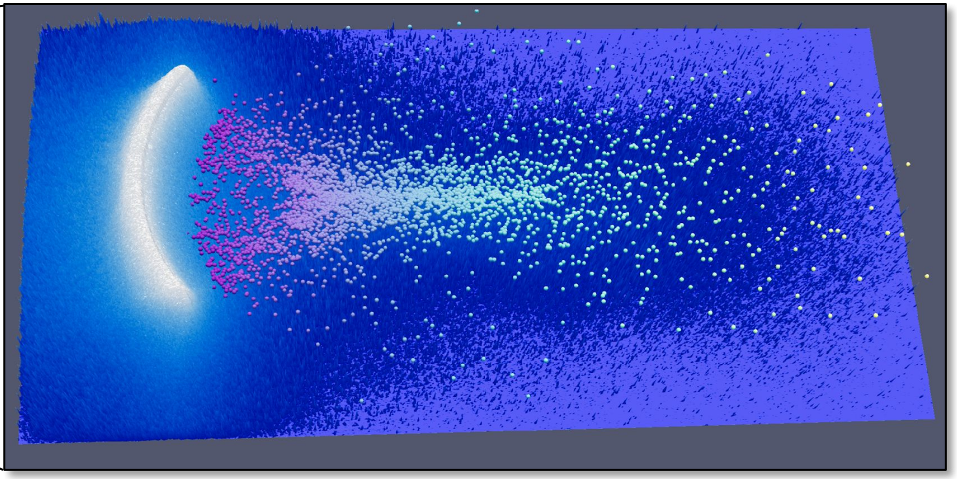
high power laser pulse

Kinetic effects in Inertial Confinement Fusion

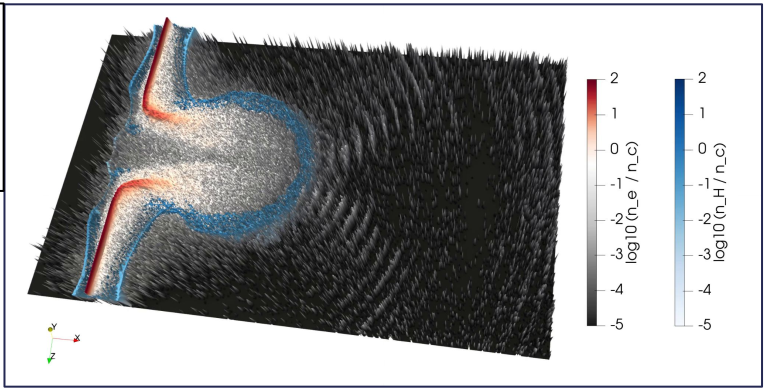

John Jett, Jake Long/LLNL

WarpX 2D on Frontier - *ParaView*

A Huebl, M Garten, J-L Vay, J Ludwig, S Wilks, A Kemp



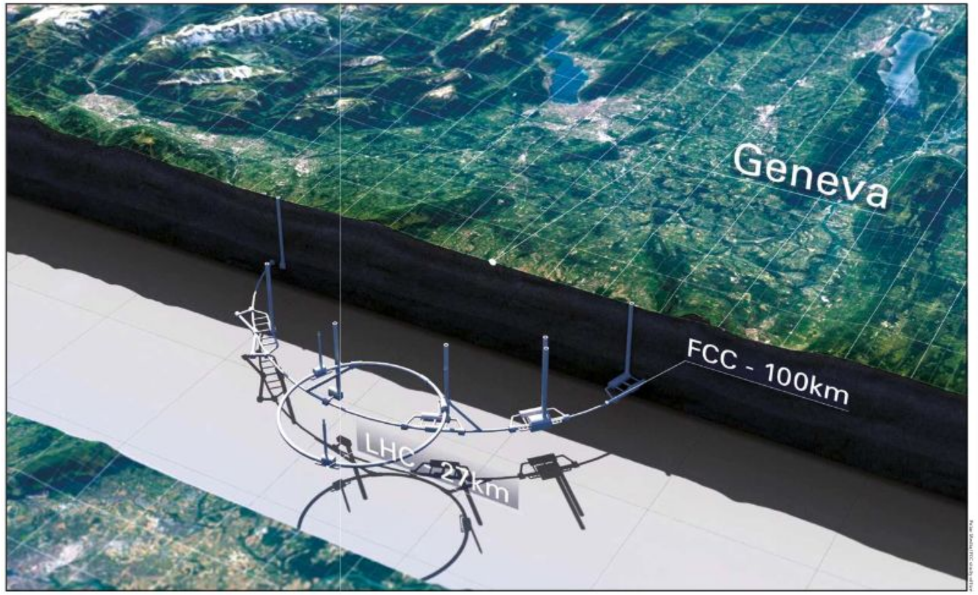
Laser-Ion Acceleration



WarpX 3D on Perlmutter - *visualized 2D with ParaView*

A Huebl, M Garten

Traditional Post-Processing: High-Energy Physics Collider



M Benedikt, F Zimmerman, A giant leap for Physics, Cern Courier 59(1), 2019

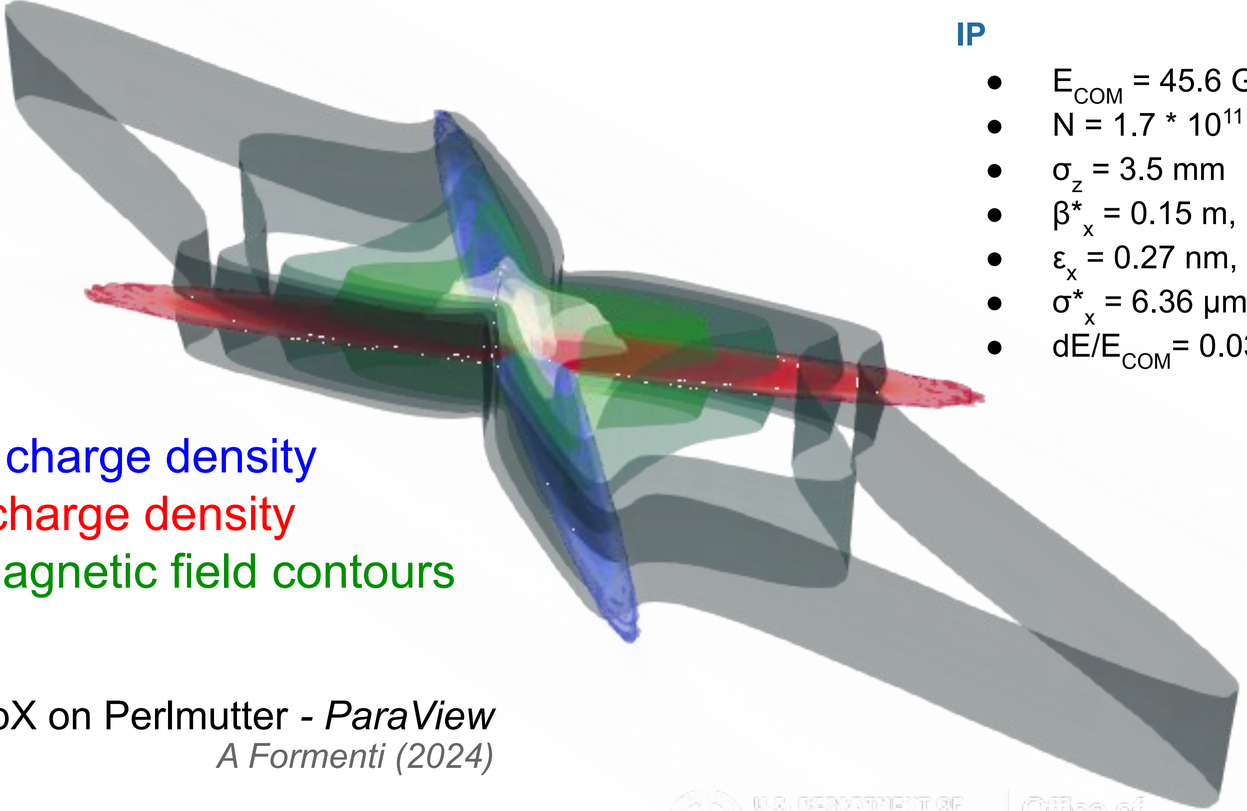
Interaction Point

Beam-Beam Crossing

FCC-ee = Future Circular Collider, electron-positron @ CERN

preliminary simulation under FCC-ee Z parameters in the collinear boosted frame at the 1st IP

- E_{COM} = 45.6 GeV
- $N = 1.7 * 10^{11}$
- σ_z = 3.5 mm
- β^*_x = 0.15 m, β^*_y = 0.8 mm
- ε_x = 0.27 nm, ε_y = 1 pm
- σ^*_x = 6.36 μm, σ^*_y = 28.3 nm
- dE/E_{COM} = 0.038%



blue: electron charge density

red: positron charge density

green-gray: magnetic field contours

WarpX on Perlmutter - *ParaView*

A Formenti (2024)

Open challenges

- large aspect ratio boxes
- streamlines and/or glyphs for the electromagnetic field
- particle trajectories

number of cells = 512 x 256 x 512

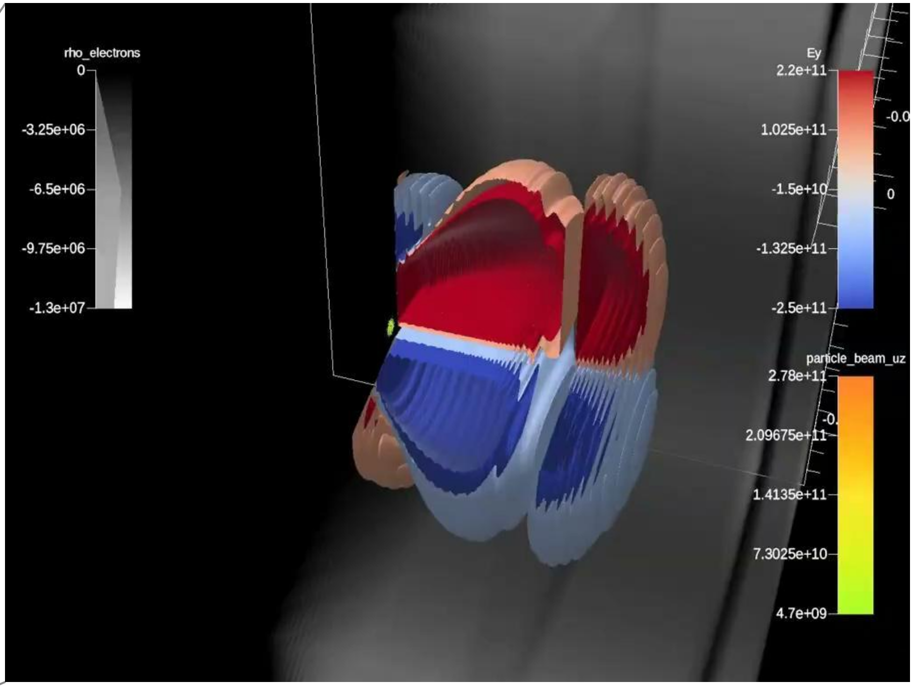
number of computational particles = 1e6 per beam

number of timesteps = 200

In Situ Processing: Laser-Electron acceleration

Integration

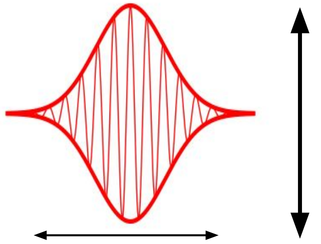
- **In situ backends** in WarpX for
  - Ascent / Conduit
  - Catalyst 2 (pending)
  - SENSEI
  - *New: Python GPU Data APIs*



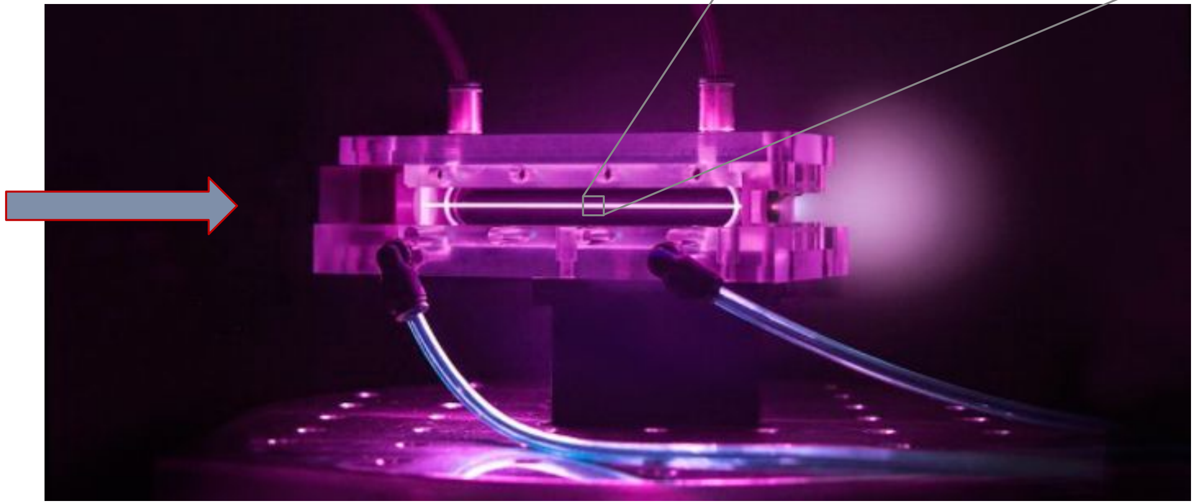
EXASCALE COMPUTING PROJECT *Staging of Laser-Wakefield Accelerators for Next-Gen Colliders*

WarpX on Frontier (552 GPUs/GCDs): transv. electric field in an LPA - *Ascent & VTK-m*

N Marsaglia, M Larsen, C Harrison, A Huebl, J-L Vay DOI:10.5281/zenodo.8226853



high power laser pulse

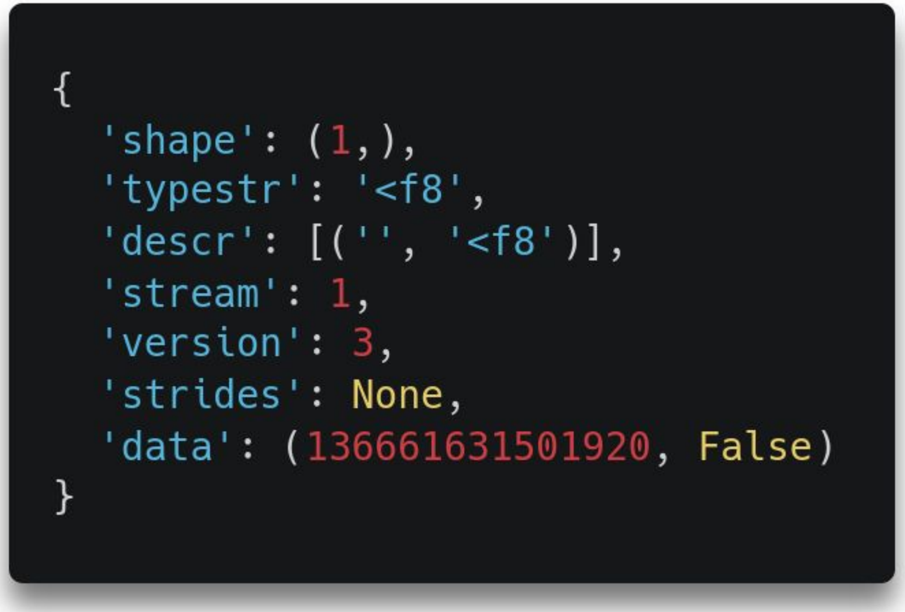


Plasma	←**Conventional**
100 GV / m	←20-200 MV/m

Augmenting & GPU-accelerating PIC Simulations: In Situ Analysis, ML Models, Vis, …

Embracing Emerging API Standards

- here: __cuda_array_interface__



```
{
  'shape': (1,),
  'typestr': '<f8',
  'descr': [('', '<f8')],
  'stream': 1,
  'version': 3,
  'strides': None,
  'data': (136661631501920, False)
}
```

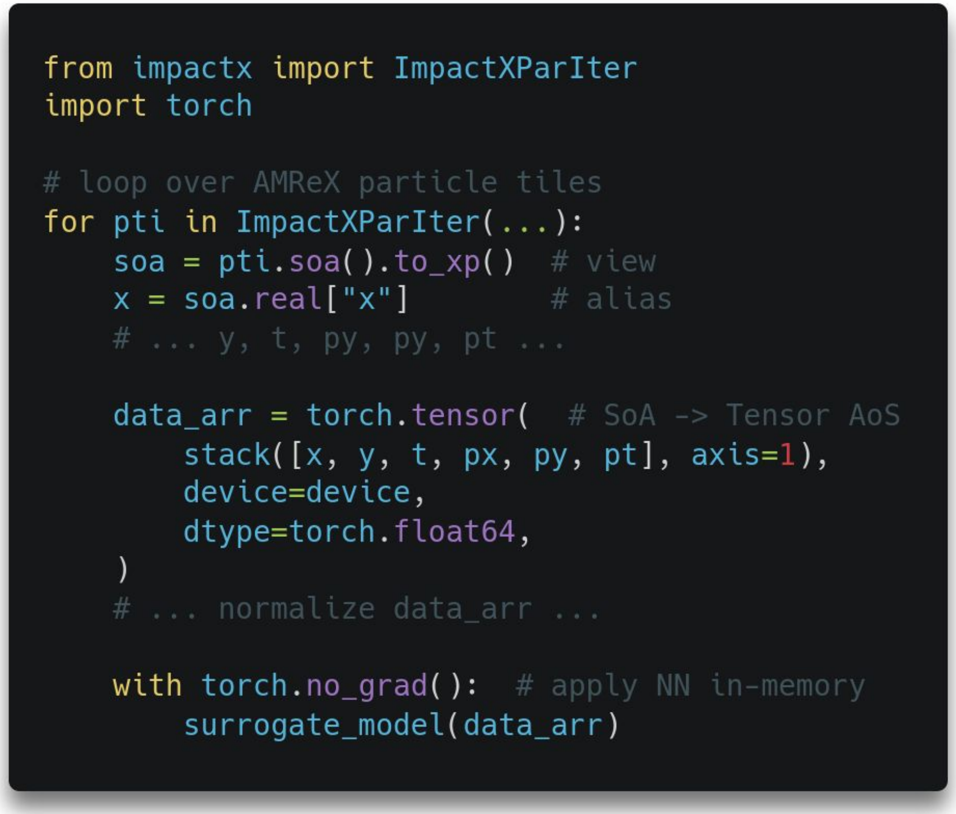
- more general: DLPack

Cross-Ecosystem, In Situ Coupling

Consortium for Python Data API Standards data-apis.org

Compute example

- data shared as views, stays on device
- enables in-memory updates



```
from impactx import ImpactXParIter
import torch

# loop over AMReX particle tiles
for pti in ImpactXParIter(...):
    soa = pti.soa().to_xp()  # view
    x = soa.real["x"]        # alias
    # ... y, t, py, py, pt ...

    data_arr = torch.tensor(  # SoA -> Tensor AoS
        stack([x, y, t, px, py, pt], axis=1),
        device=device,
        dtype=torch.float64,
    )
    # ... normalize data_arr ...

    with torch.no_grad():  # apply NN in-memory
        surrogate_model(data_arr)
```

A Huebl et al., "pyAMReX: GPU-Enabled, Zero-Copy AMReX Python Bindings including AI/ML" (2023)
A Myers et al., "AMReX and pyAMReX: Looking Beyond ECP, under review, arXiv:2403.12179" (2024)
R Sandberg et al., "Synthesizing Particle-in-Cell Simulations Through Learning and GPU Computing for Hybrid Particle Accelerator Beamlines"(2024)

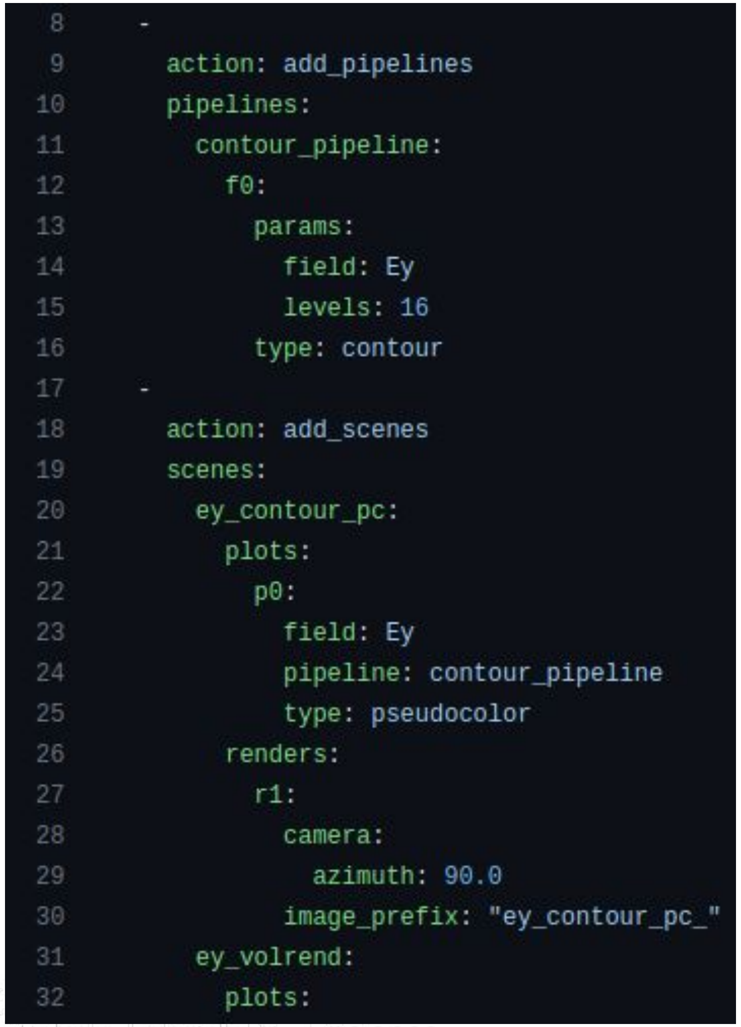
Data Understanding: Opportunities in Visualization

Selected Research Opportunities

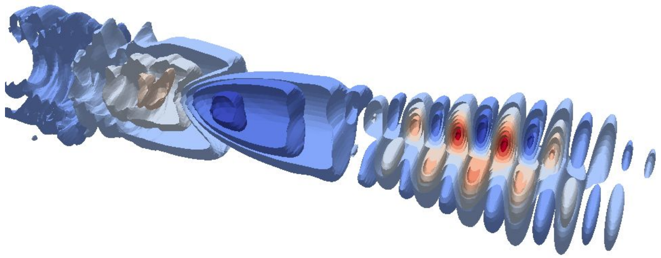
- **Rapid scene design - standardization**[1]
  - exchange scenes: post-processing GUIs ⇆ in situ
  - ambience: load external/STL geometries
  - domain-specific GUIs, e.g., *trame*

- ***In Situ* Animations**
  - flicker: iso-contours, glyphs, streamlines
  - reason: roughness of simulation data & steps
  - challenge: smooth transitions as in web/CSS?

- **Quality of *In Situ* Generated Vis**
  - casting soft+hard shadows, tracing reflections, semi-transparent iso-contours, smoothing, volume-rendering >1 overlapping source



```
  -
    action: add_pipelines
    pipelines:
      contour_pipeline:
        f0:
          params:
            field: Ey
            levels: 16
          type: contour
  -
    action: add_scenes
    scenes:
      ey_contour_pc:
        plots:
          p0:
            field: Ey
            pipeline: contour_pipeline
            type: pseudocolor
        renders:
          r1:
            camera:
              azimuth: 90.0
            image_prefix: "ey_contour_pc_"
      ey_volrend:
        plots:
```



WarpX on Summit - *Ascent*

A Huebl et al. (2020)

---

[1] Conduit, OpenUSD.org, ANARI

Supplementary Materials

GPU-accelerated Synthesis:
PIC Simulations & ML Models

Demonstrated profits from GPUs

- *first-principle models:*
  Particle-in-Cell simulations
- *data-driven models:*
  neural network training & inference

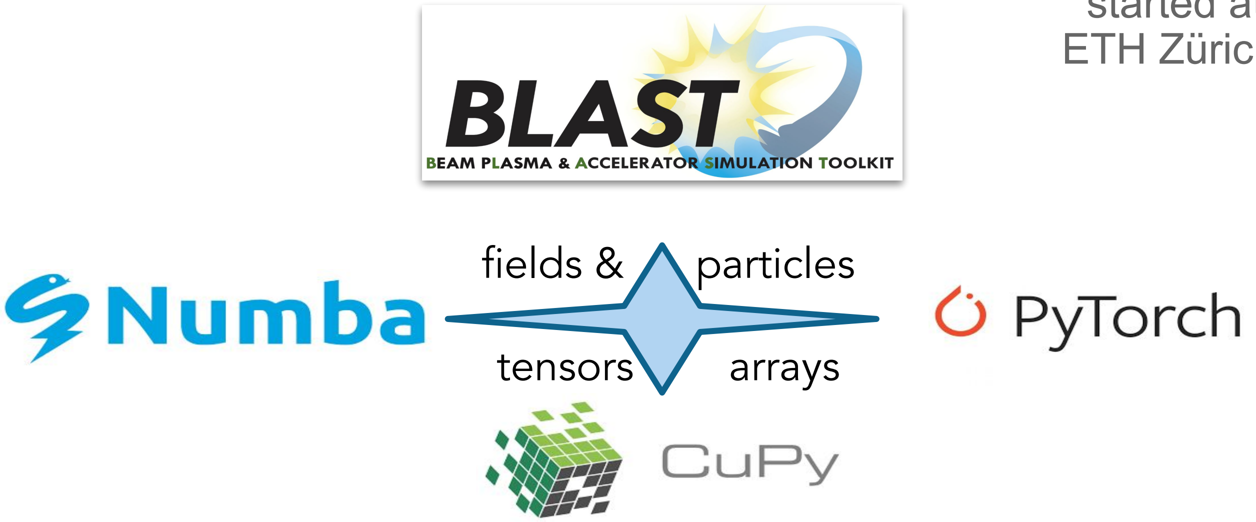
Approach

- Creation of a *compatible ecosystem*
- C++ core, Python control/glue
- pure C++ Python bindings: pybind11

started at
ETH Zürich!

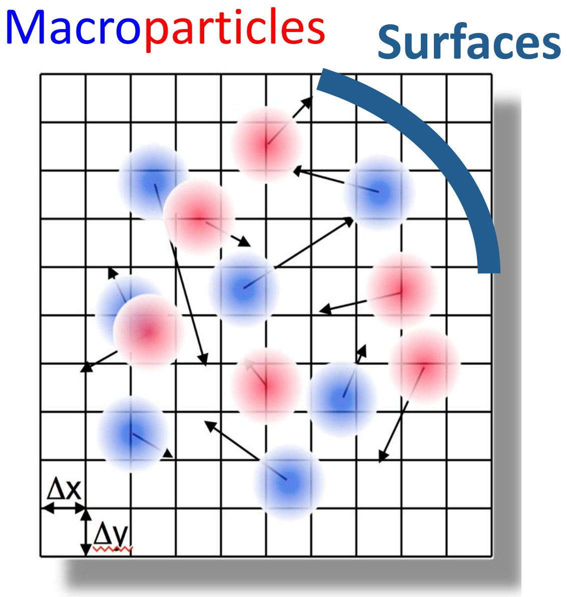
Implementation Goals

- **augment & accelerate** *on-GPU* PIC simulations with *on-GPU* ML models
- support many **HPC C++ compilers**
- **rapid ML model design** "plug-and-play"



W Jakob et al., pybind11 – Seamless operability between C++11 and Python (2017)
A Huebl et al., pyAMReX: GPU-Enabled, Zero-Copy AMReX Python Bindings including AI/ML (2023)
A Myers et al., AMReX and pyAMReX: Looking Beyond ECP, under review, arXiv:2403.12179 (2024)

First Principle Particle-in-Cell Modeling of Particle Accelerators requires the best algorithms and fastest computers



electromagnetic (EM) fields on a grid

Involves the modeling of the intricate interactions of

- **relativistic particles**: beams, plasmas
- **EM fields**: accelerating/focusing fields, beam self-fields, laser/plasma fields
- **structures**: metals, dielectrics.

Typical computer representations:

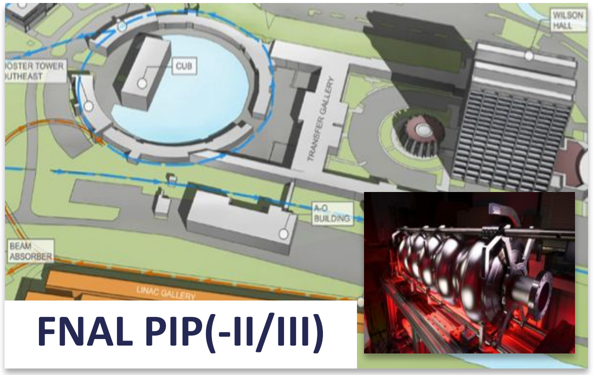
- **particles:** macro particles representing each 1-10^6 particles
- **fields:** electromagnetic, on a grid
- **structures:** surfaces interacting with grid and macroparticles

Many space- and time scales to cover:

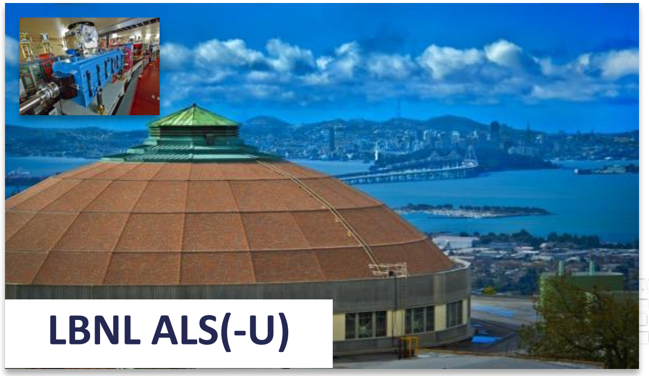
- from **µm** (e.g., plasma structures, e^--surface interactions) **to km** (e.g., LHC)
- from **ns** (beam passing one element) **to seconds or more** (beam lifetime)

⇒ needs best algorithms on largest & fastest computers

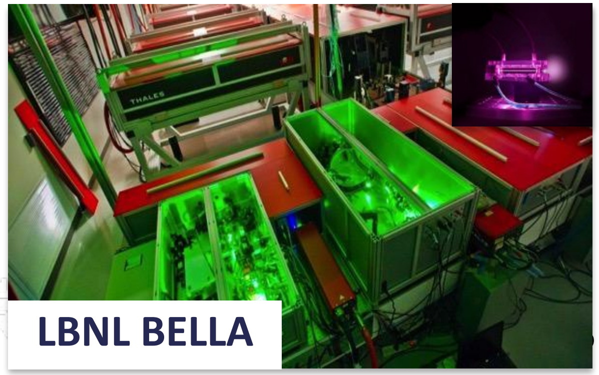
CERN (HL-)LHC



FNAL PIP(-II/III)



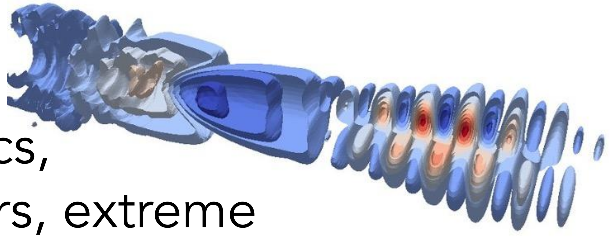
LBNL ALS(-U)



LBNL BELLA

WarpX is a Community Exascale Particle-in-Cell Code

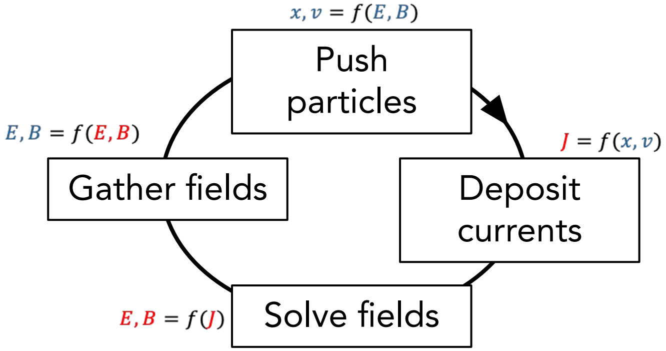
Applications



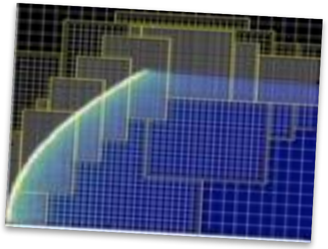
laser-plasma physics,
particle accelerators, extreme
light sources, fusion devices & plasmas, …

Exascale Particle-in-Cell Code

- electromagnetic or electro/magnetostatic
- PIC-fluid hybrid
- time integration: explicit, implicit



$x, v = f(E, B)$
Push particles
$J = f(x, v)$
Deposit currents
$E, B = f(J)$
Solve fields
$E, B = f(E, B)$
Gather fields



International Contributors incl. private sector

Award–Winning Code & Science

PLASMA SIMULATION CODE WINS 2022 ACM GORDON BELL PRIZE

18 NOV 2022

Portable, Multi-Level Parallelization

- MPI: 3D MR decomposition
  - dynamic load balancing
- GPU: CUDA, HIP and SYCL
- CPU: OpenMP

Desktop to HPC

Scalable & Standardized

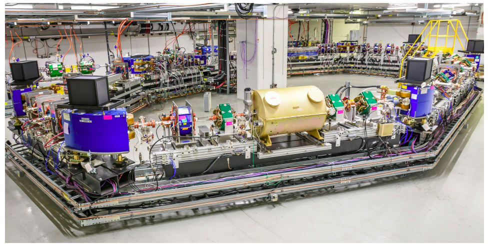
- Python APIs, openPMD data
- In situ processing
- Open community ecosystem

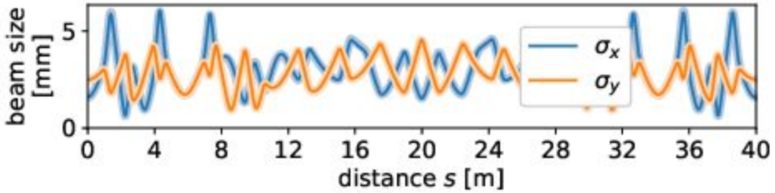
J-L Vay et al., NIMA 909.12 (2018)
L Fedeli, A Huebl et al., SC22, DOI:10.1109/SC41404.2022.00008 (2022)

ImpactX Leverages WarpX Technology to Model *Whole Beamlines*

Applications

Beam-Dynamics in
Linacs, Rings, Colliders

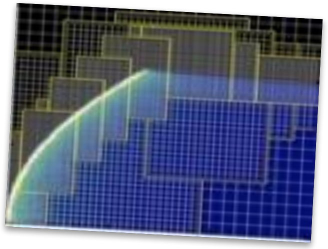




Electrostatic Particle-in-Cell Code

- *integrate* relative to a *reference particle*
- particle advance: symplectic maps
- collective effects: space charge,

  soon: radiative effects



efficient modeling of large scales (e.g. km) for full beamlines

Portable, Multi-Level Parallelization

- MPI: 3D MR decomposition
- GPU: CUDA, HIP and SYCL
- CPU: OpenMP

Desktop to HPC

Scalable & Standardized

- Python APIs, openPMD data
- In situ processing
- Open community ecosystem

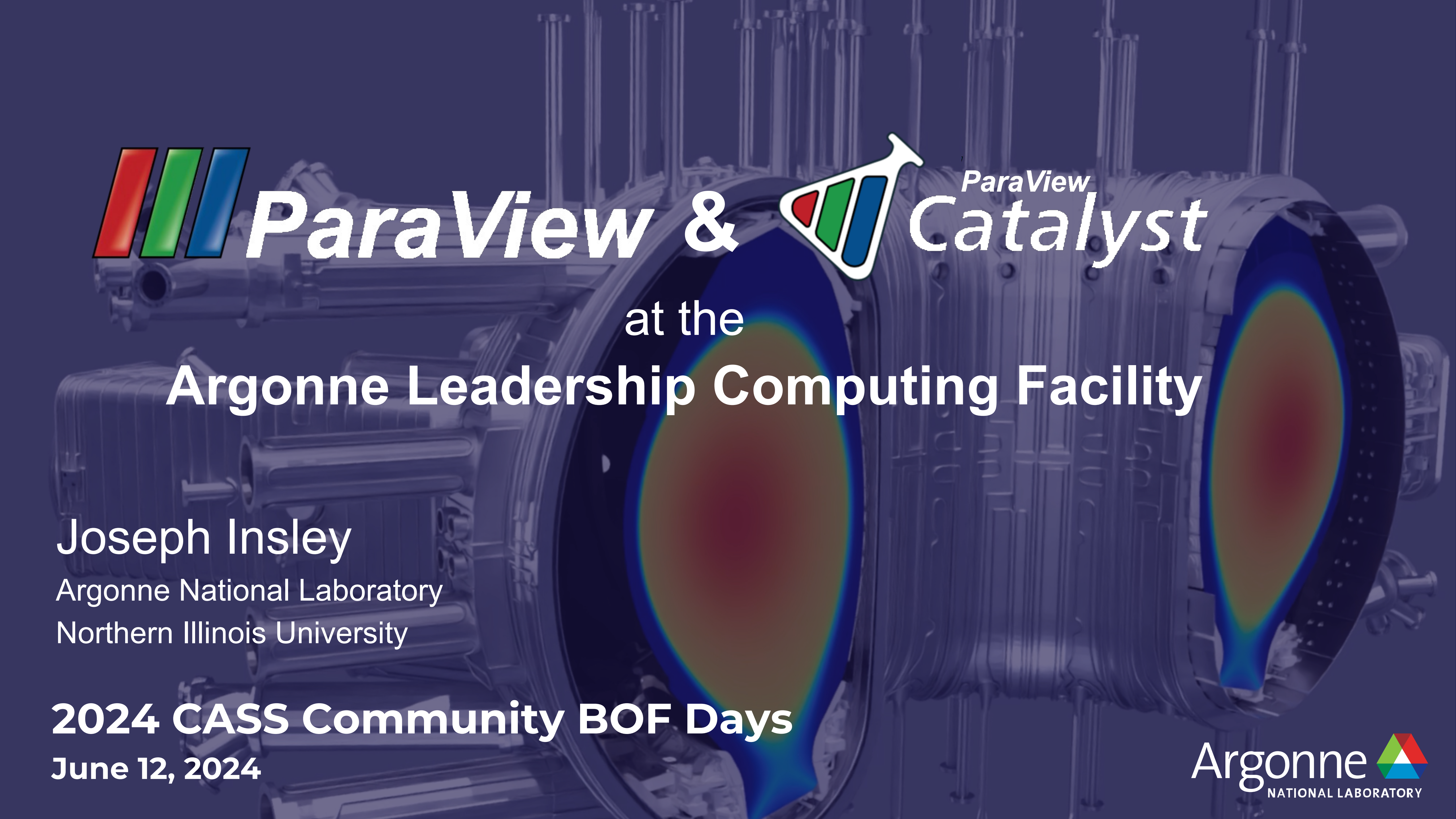
openPMD

AMReX

BLAST
BEAM PLASMA & ACCELERATOR SIMULATION TOOLKIT

U.S. DEPARTMENT OF ENERGY
Office of Science

A Huebl et al., NAPAC22 and AAC22 (2022)
C Mitchell et al., HB2023, THBP44 and TUA2I2 (2023)
J Qiang et al., PRSTAB (2006); RD Ryne et al., ICAP2006 ICAP2006 (2006)



ParaView & ParaView Catalyst at the Argonne Leadership Computing Facility

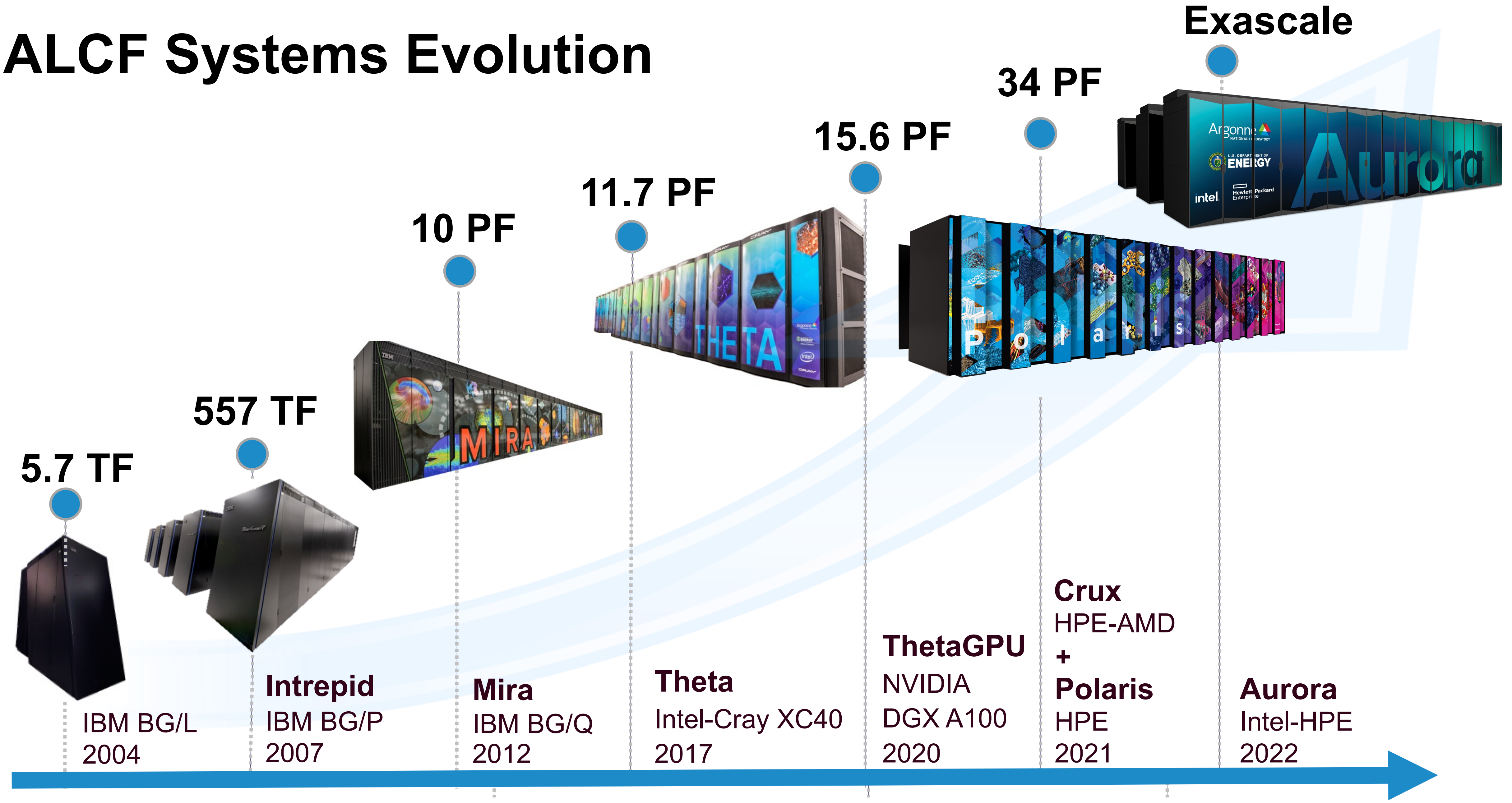
Joseph Insley
Argonne National Laboratory
Northern Illinois University

2024 CASS Community BOF Days

June 12, 2024

Argonne NATIONAL LABORATORY

ALCF Systems Evolution



Exascale

34 PF

15.6 PF

11.7 PF

10 PF

557 TF

5.7 TF

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)

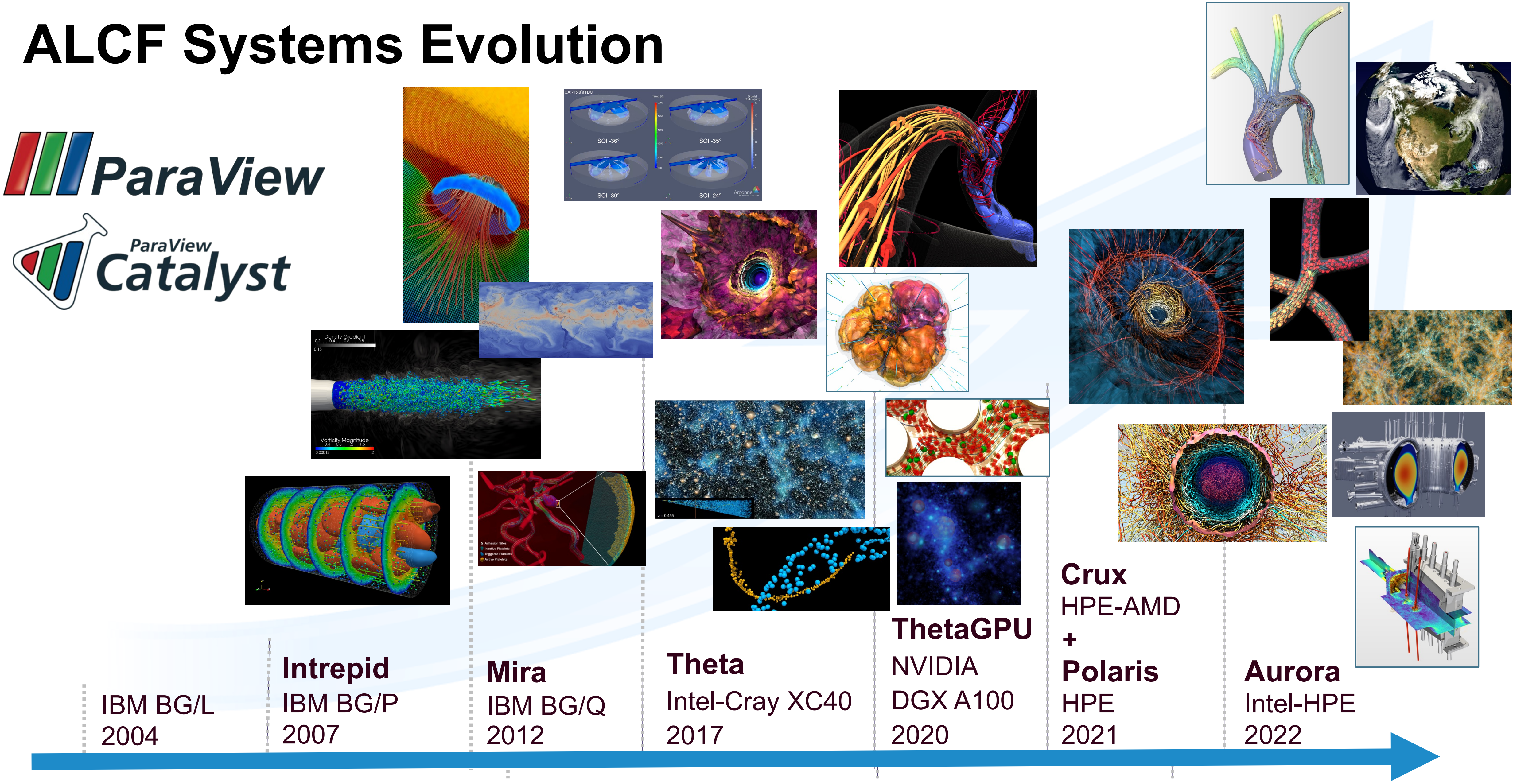
AI Testbed (2020)

Edge Testbed (2021)

ALCF Systems Evolution

ParaView

ParaView Catalyst



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

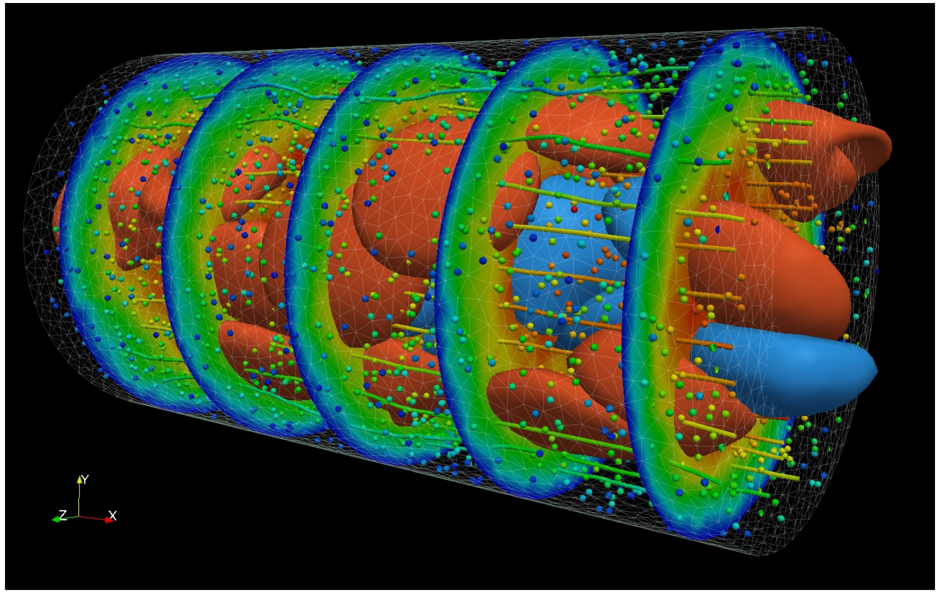
Aurora
Intel-HPE
2022

JLSE (2013) **AI Testbed** (2020) **Edge Testbed** (2021)

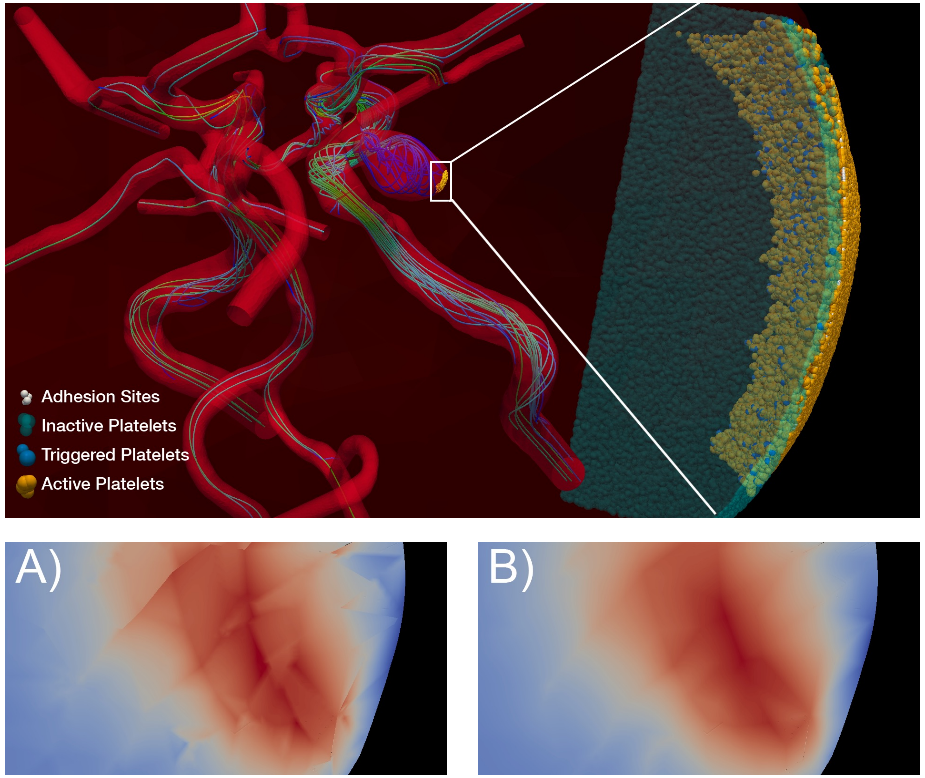
PI: George Karniadakis, Brown University

SC11: Gordon Bell Honorable Mention
NekTar Reader Plug-in

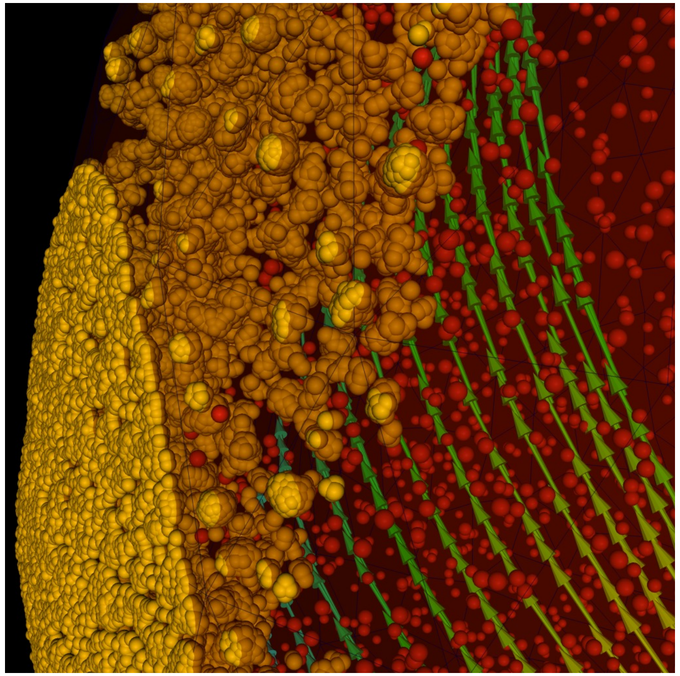
2010



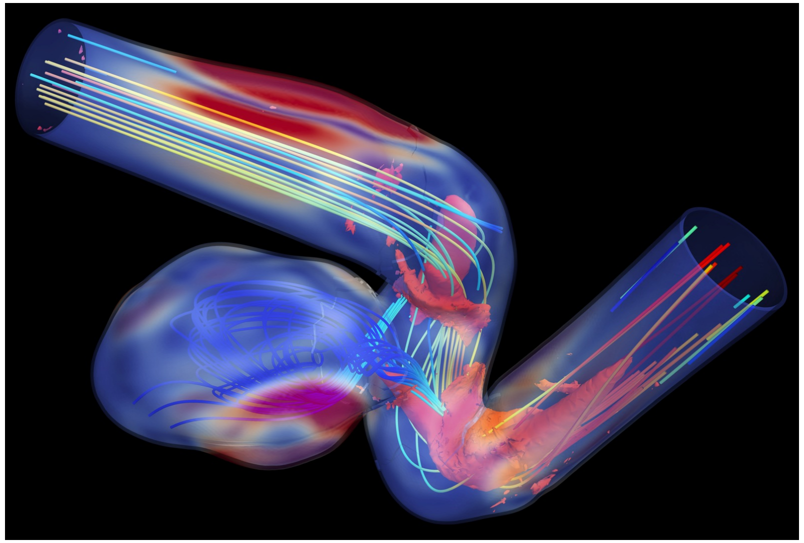
2011



2012

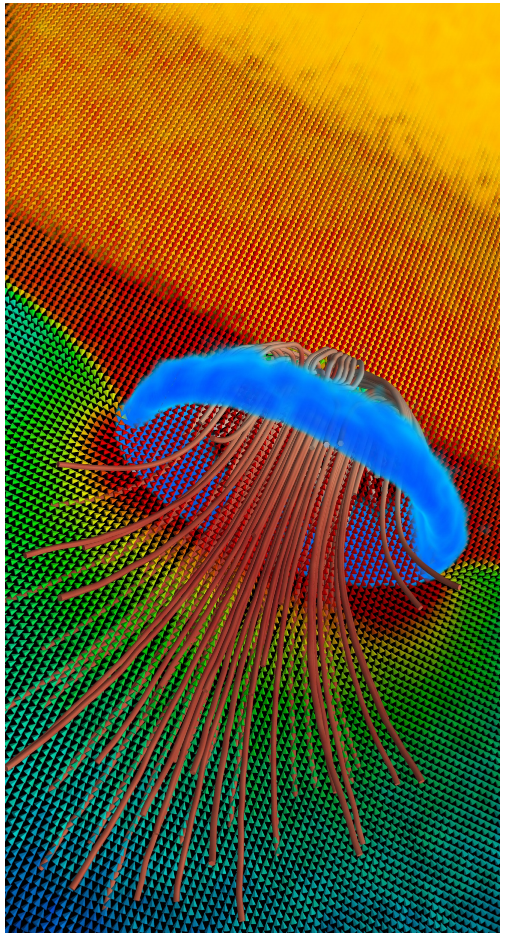


2014



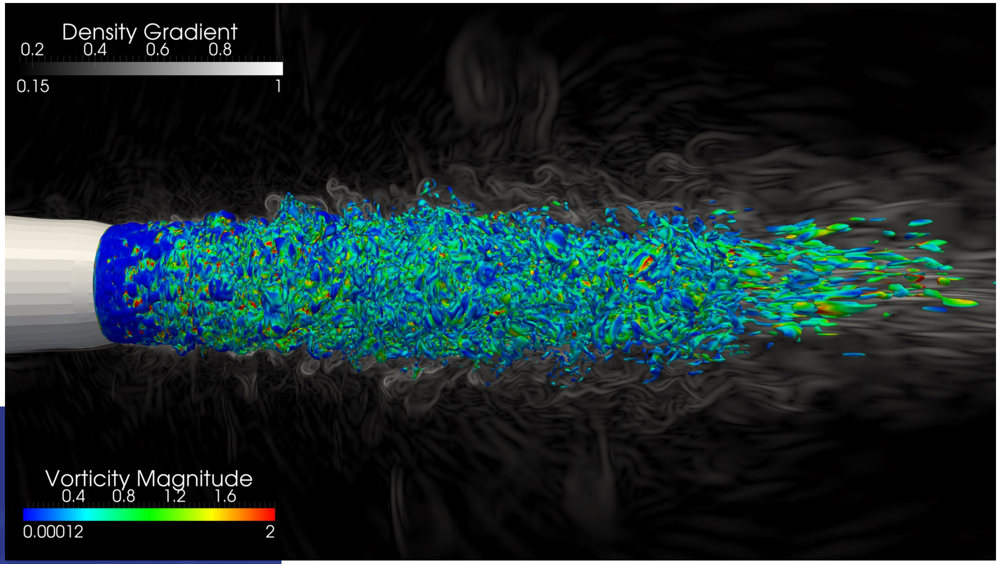
PI: Priya Vashishta, University of Southern California

2011

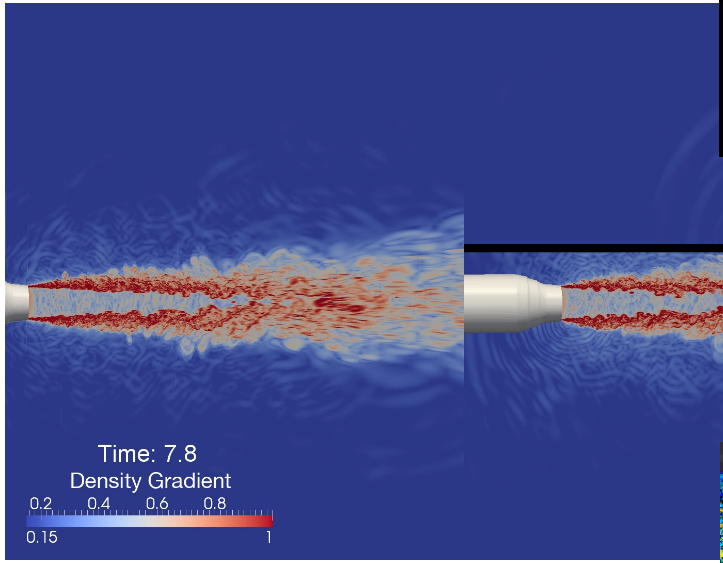


PI: Umesh Paliath, GE Global Research

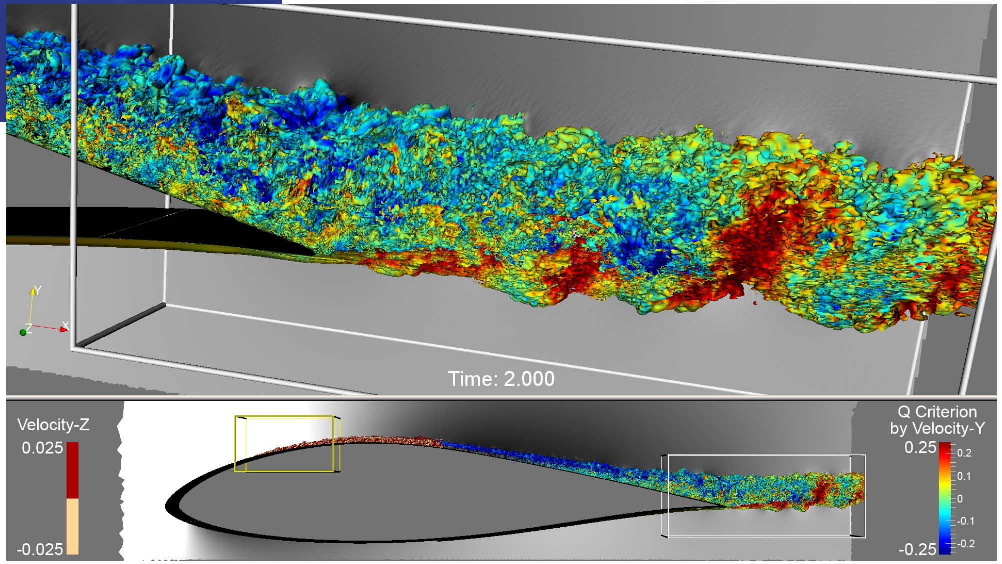
2010



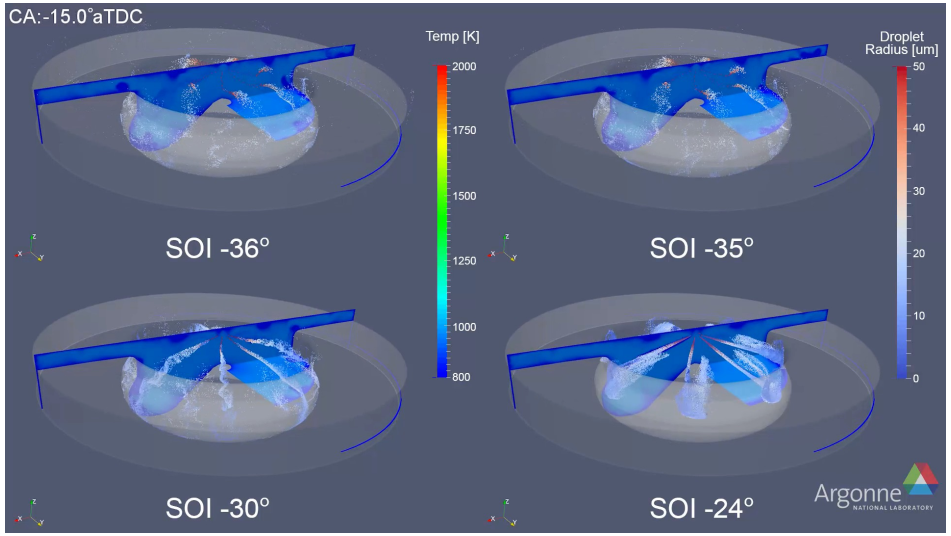
2011



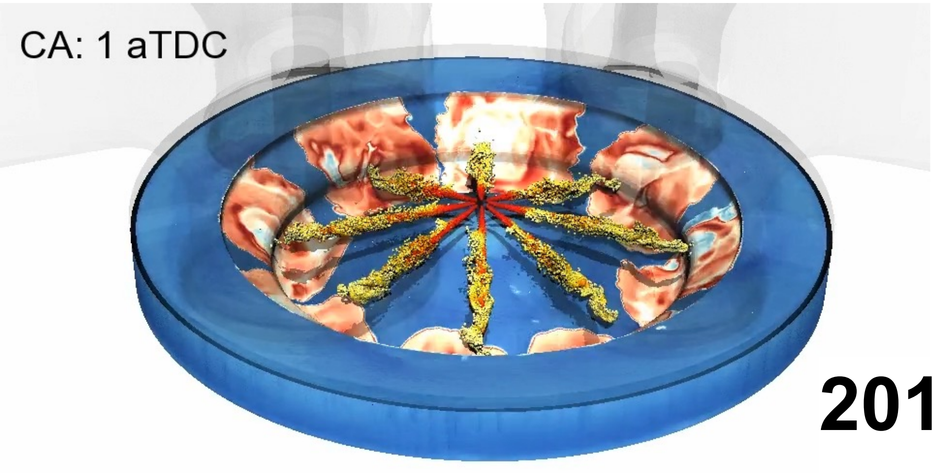
2013



PI: Sibendu Som, Argonne National Laboratory

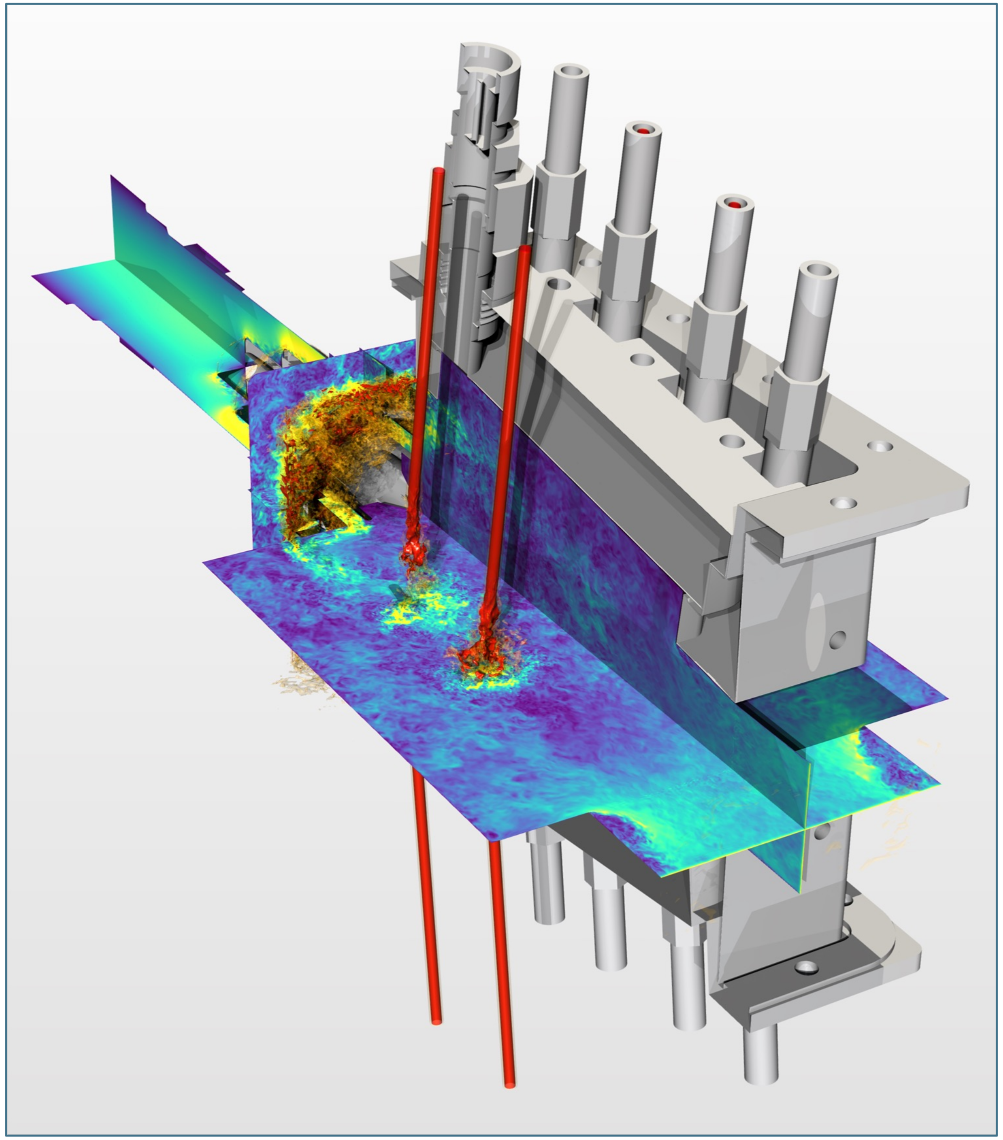


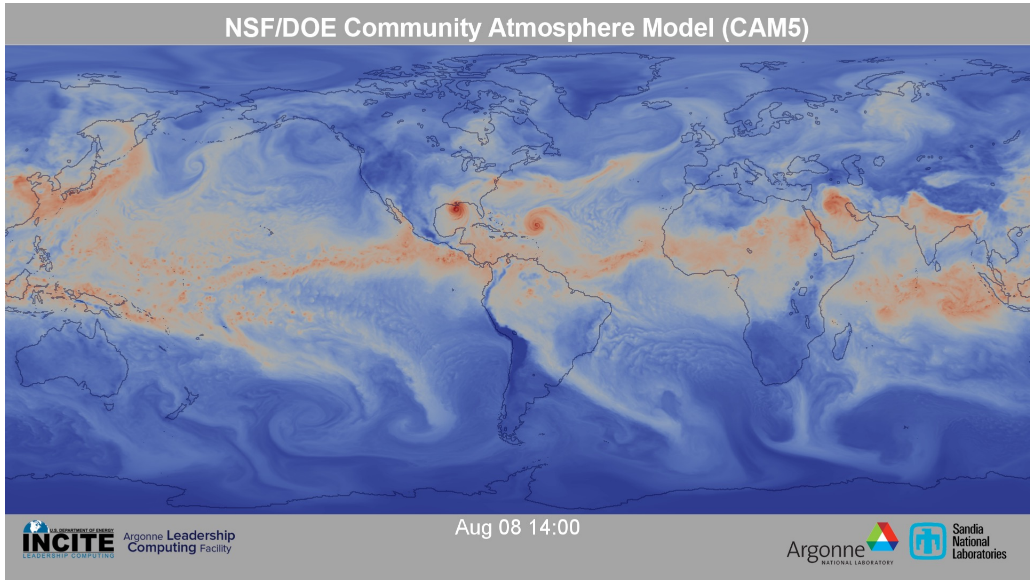
2015



2017

2023




NSF/DOE Community Atmosphere Model (CAM5)
Aug 08 14:00
INCITE
Argonne Leadership Computing Facility
Argonne
Sandia National Laboratories

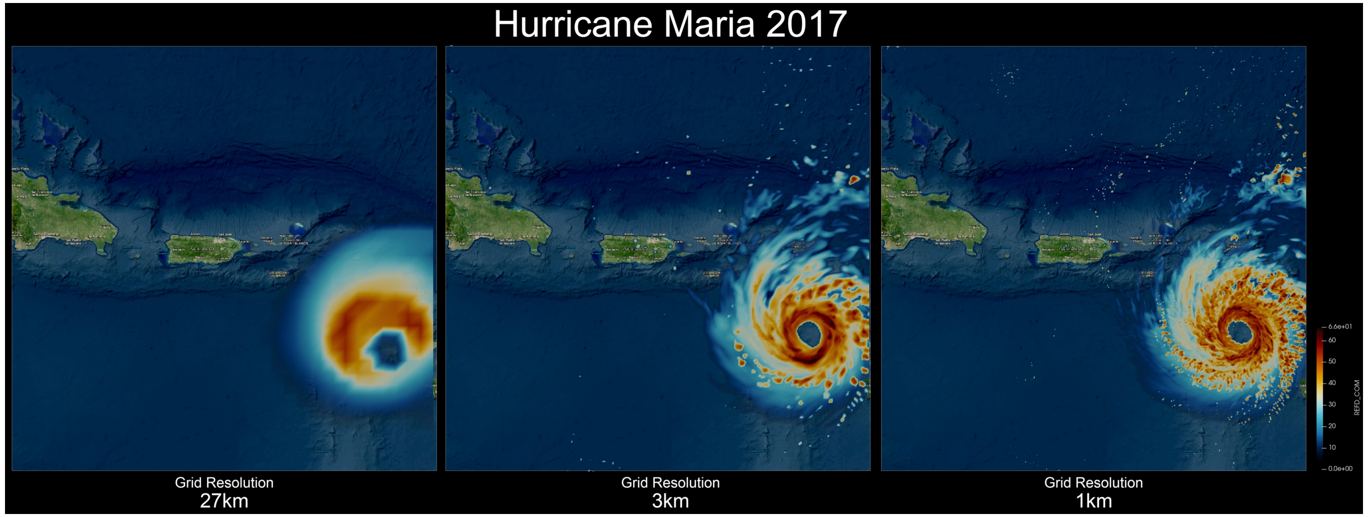
PI: Warren Washington, National Center for Atmospheric Research

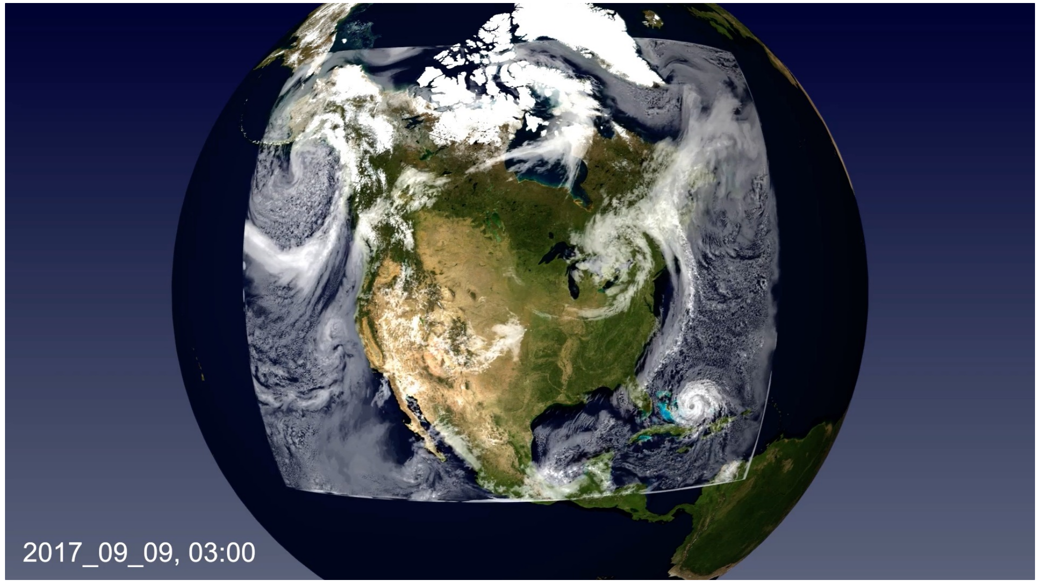
**PI: Rao Kotamarthi,
Argonne National Laboratory**

2012

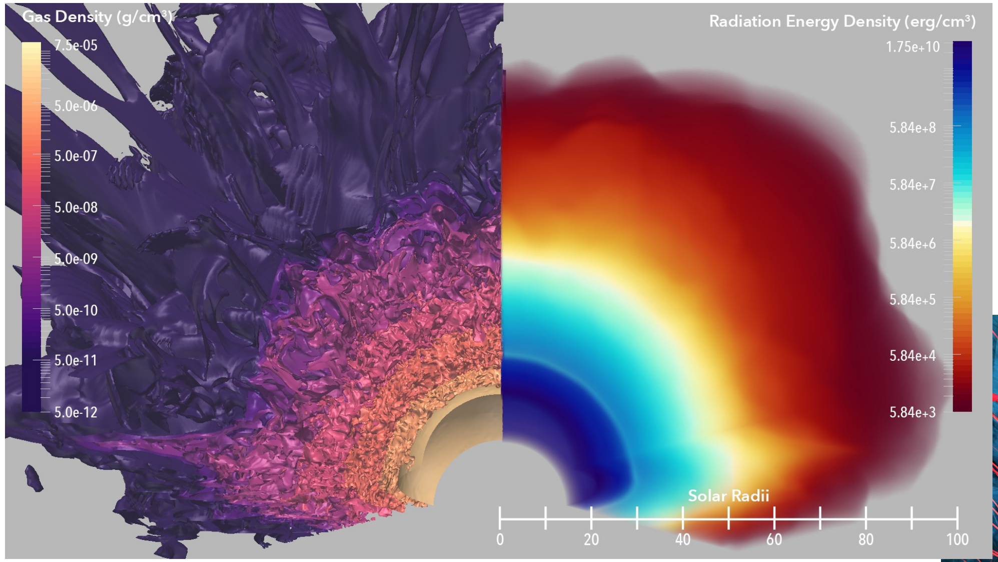
2024

2022





PI: Lars Bildsten, University of California, Santa Barbara



2017

2018

OUTLOOK
Brain cancer

nature

THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE

STELLAR OUTBURST

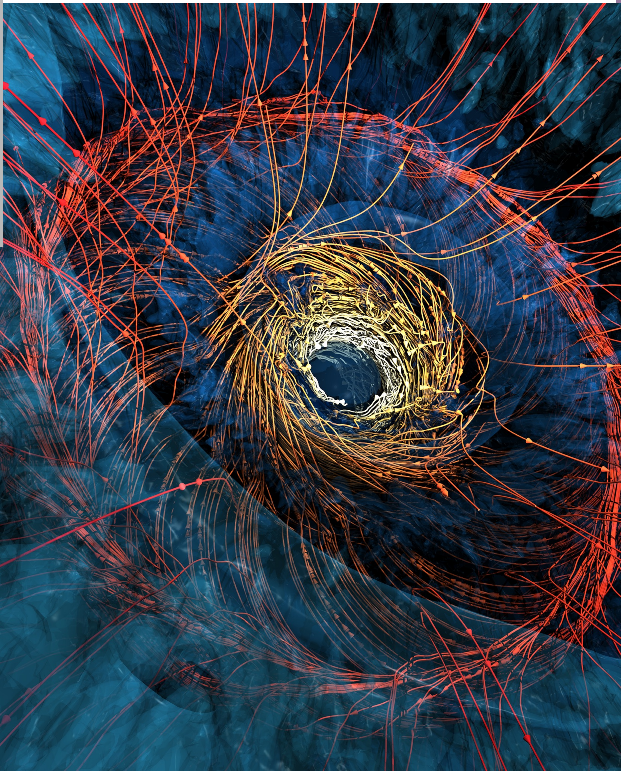
3D simulations reveal source of variability in massive blue stars
PAGE 498

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

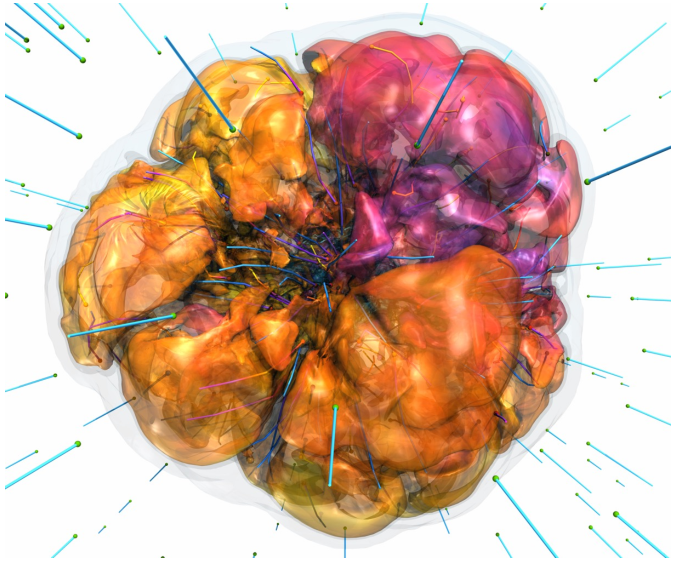
BIOCHEMISTRY
ROLL OUT THE β-BARREL
Targeted protein binding by computational design
PAGES 471 & 485

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



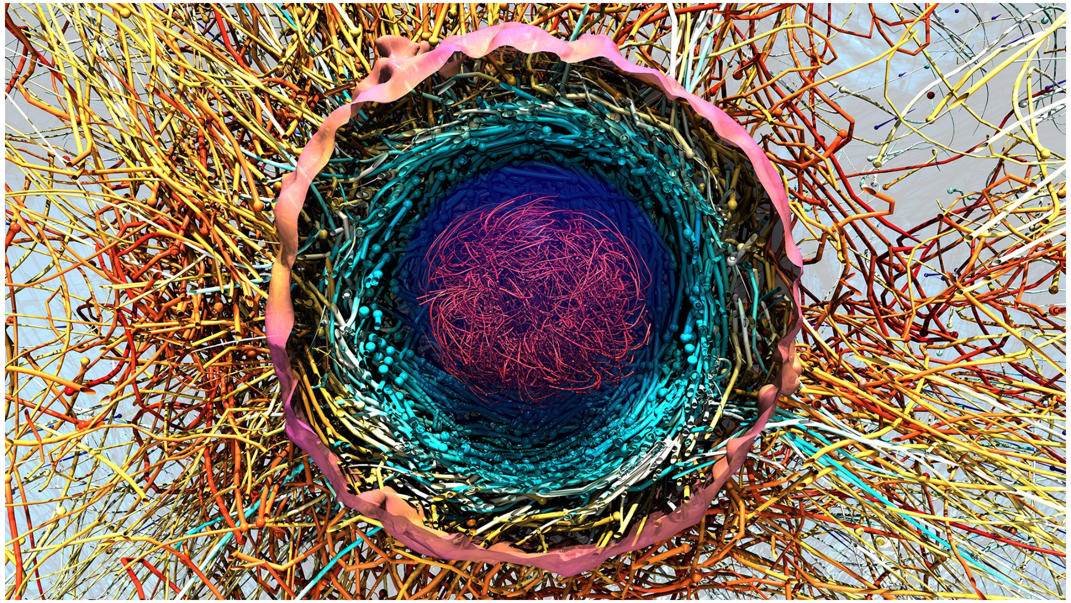
2021

PI: Adam Burrows, Princeton University

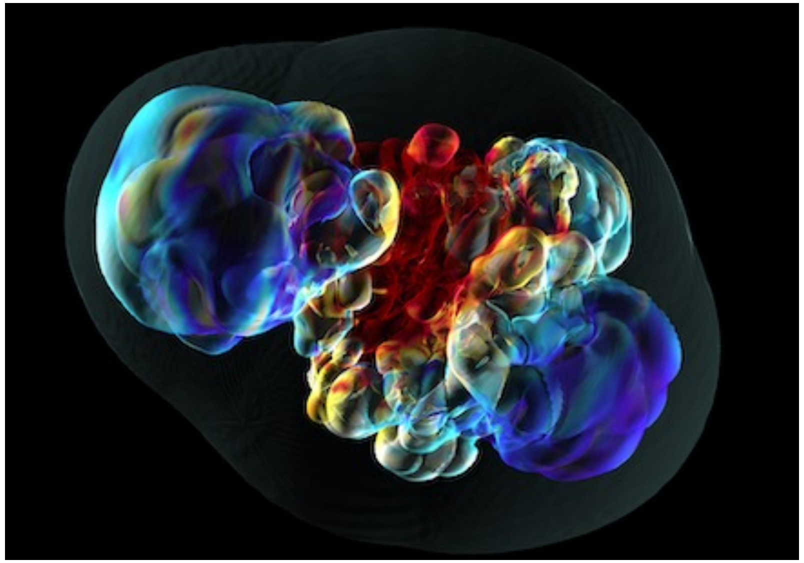


2019

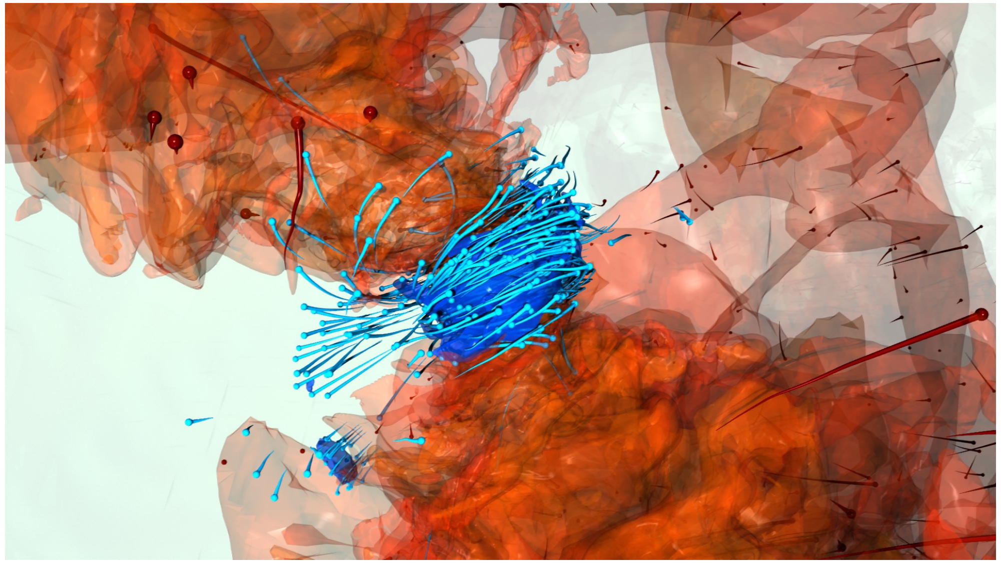
2021



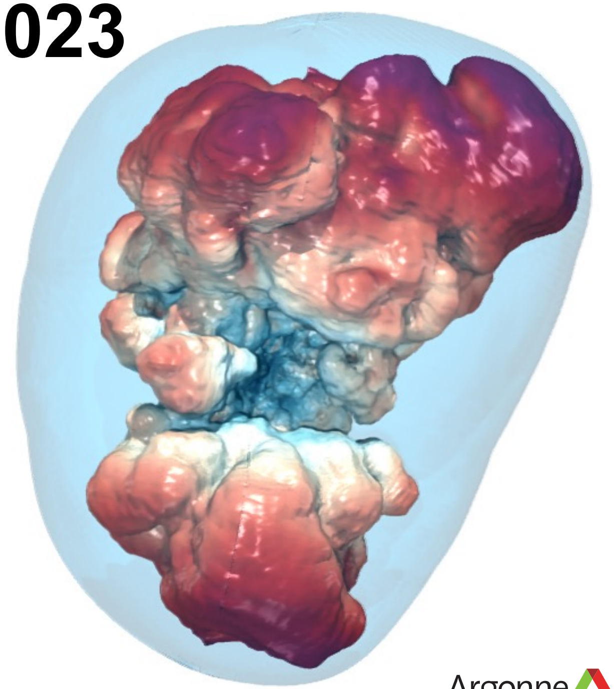
2022



2023

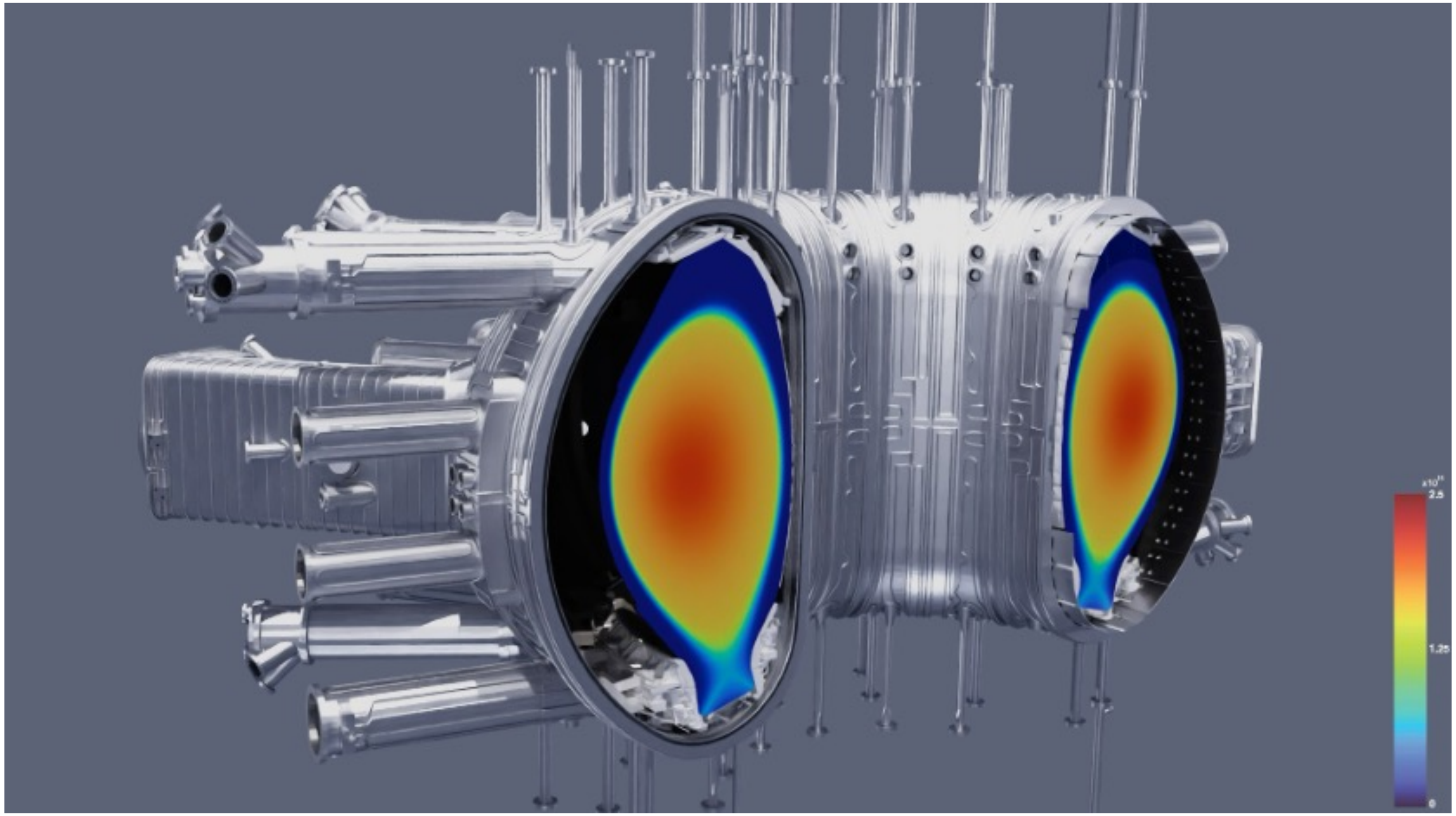


2023

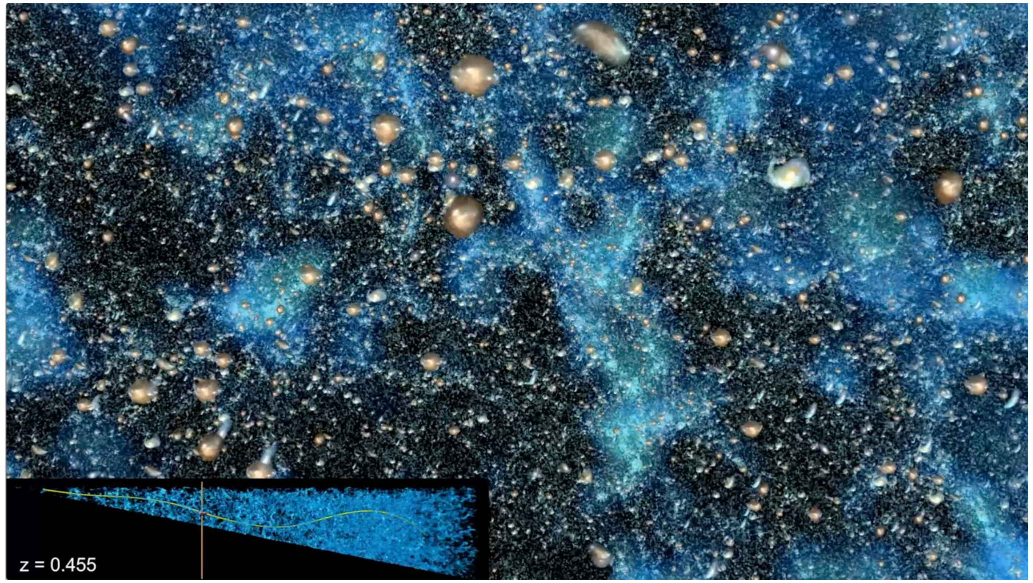


2024

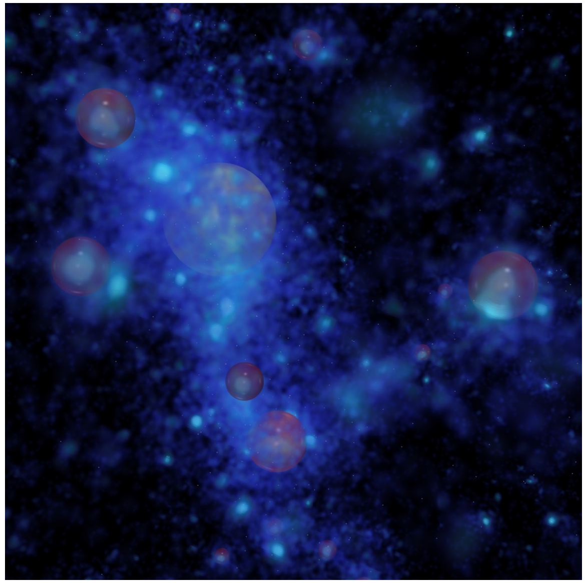
Science: Julien Dominski, Choong-Seock Chang



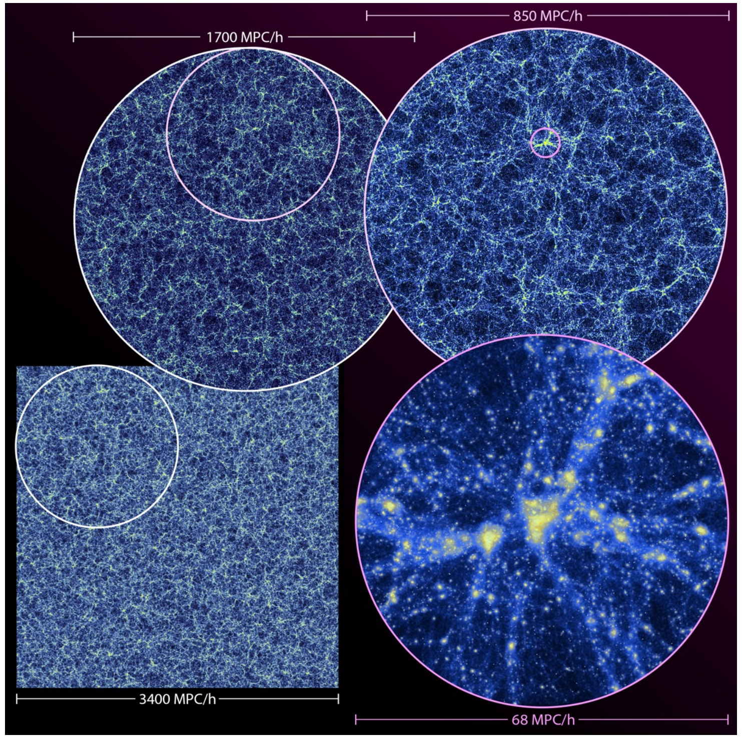
2018

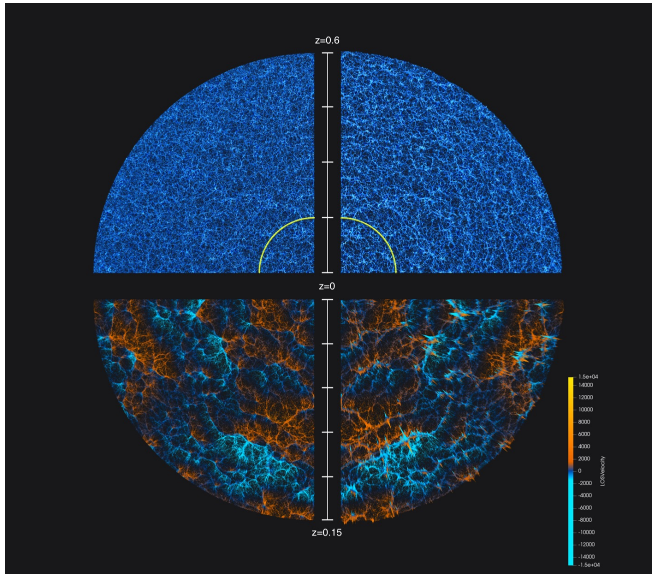


2020

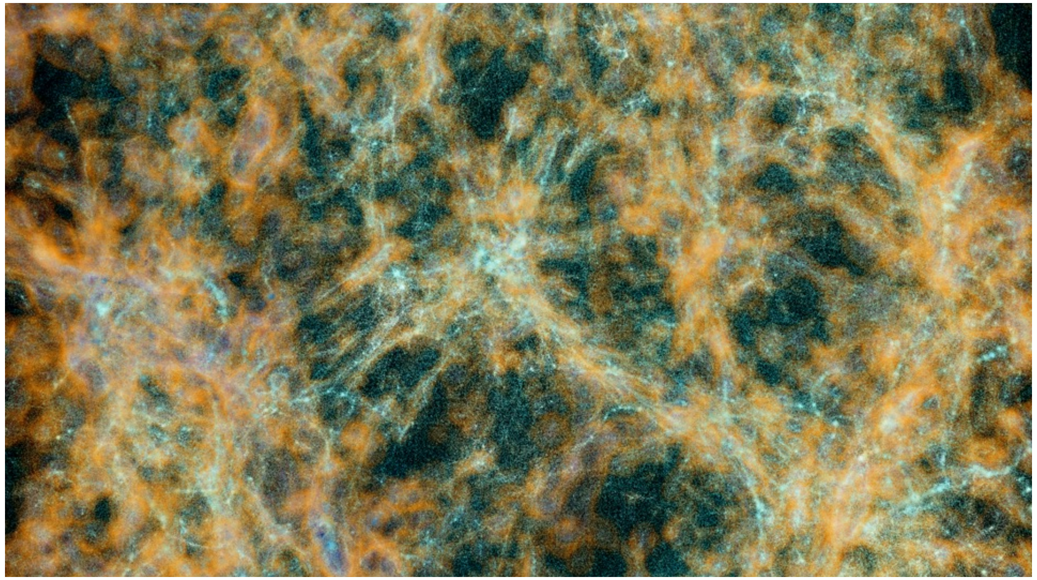


2020





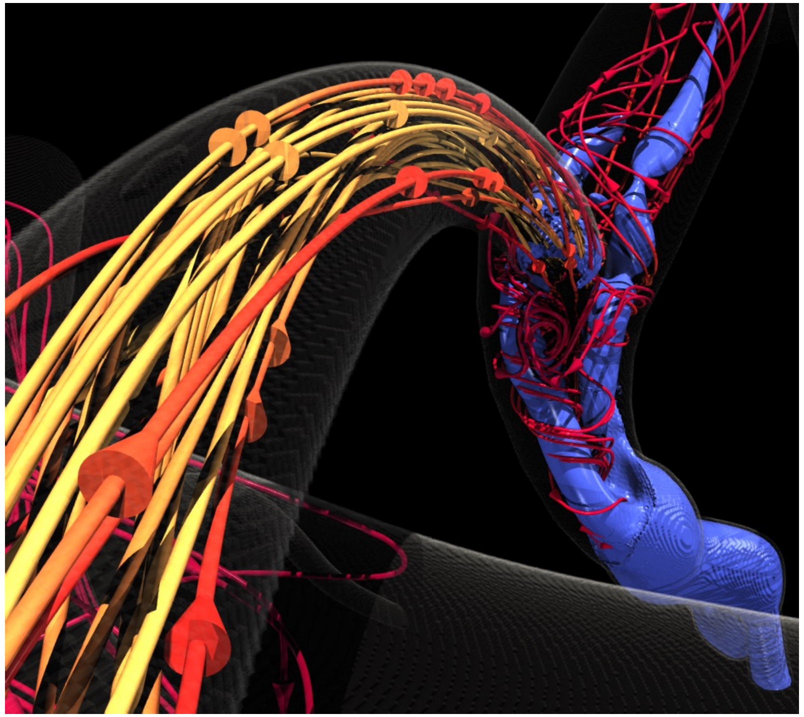
2021



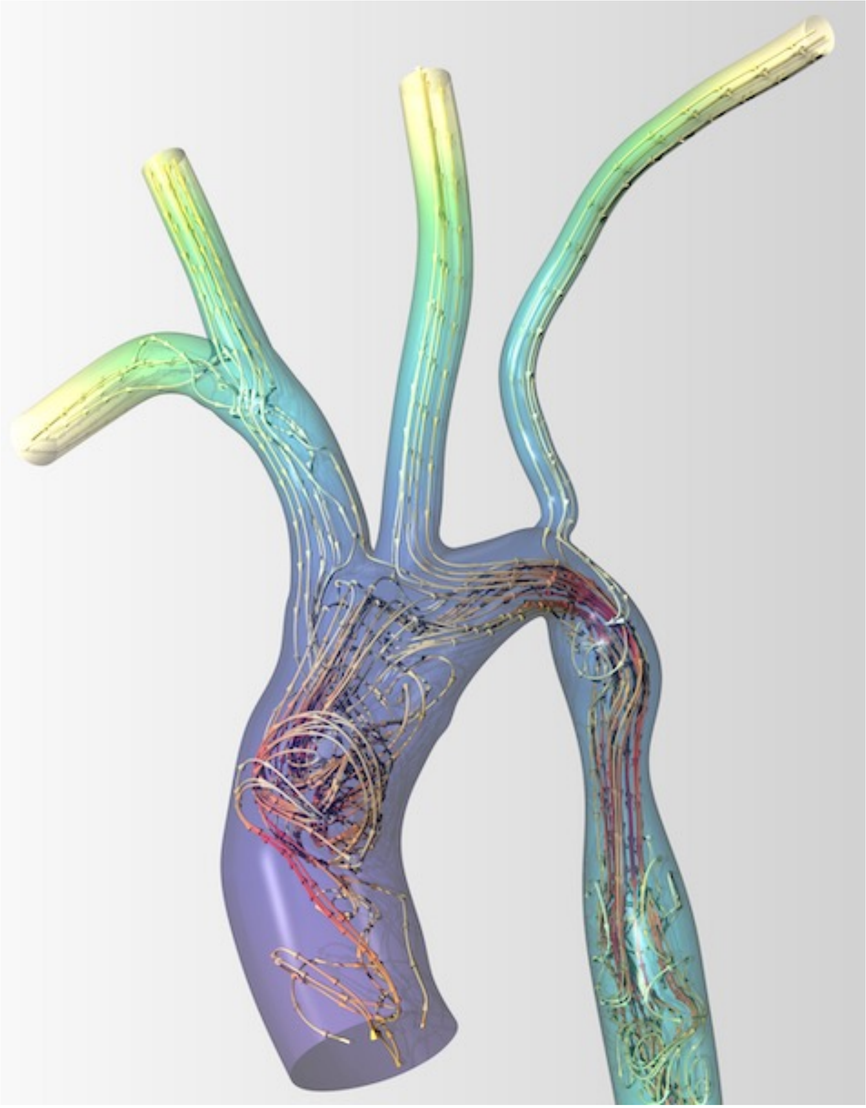
Computed and Rendered on Aurora 2023

PI: Salman Habib and HACC Team, Argonne National Laboratory

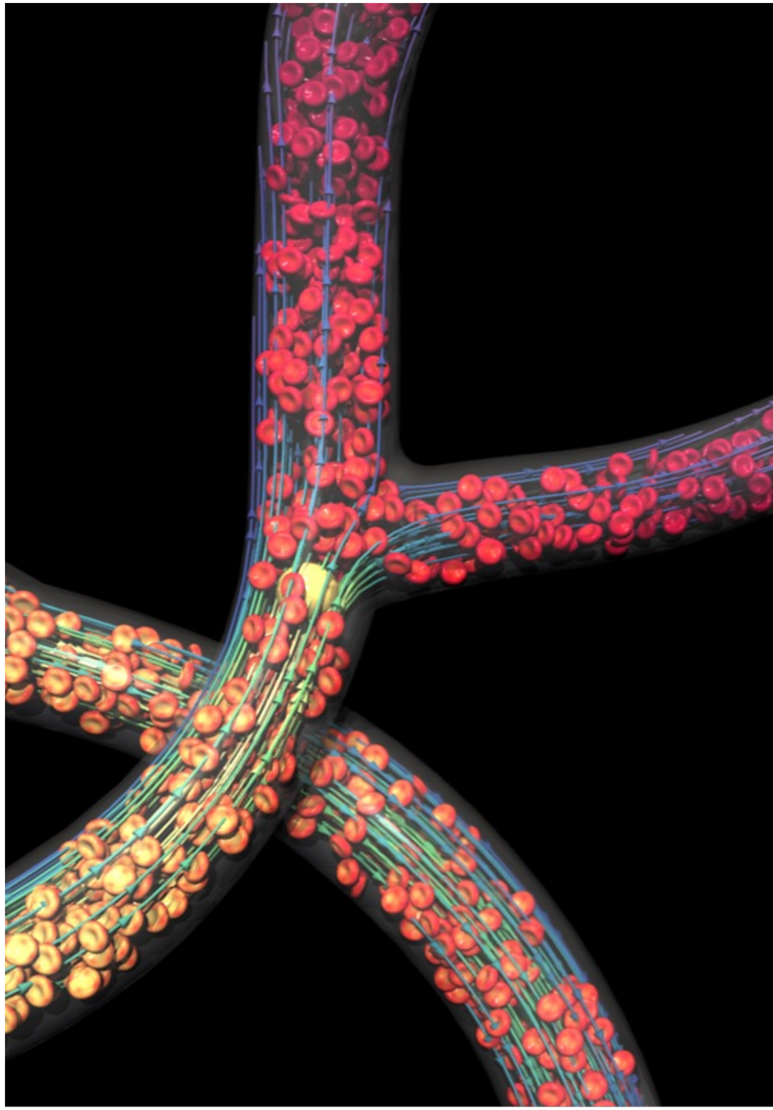
PI: Amanda Randles, Duke University



2020



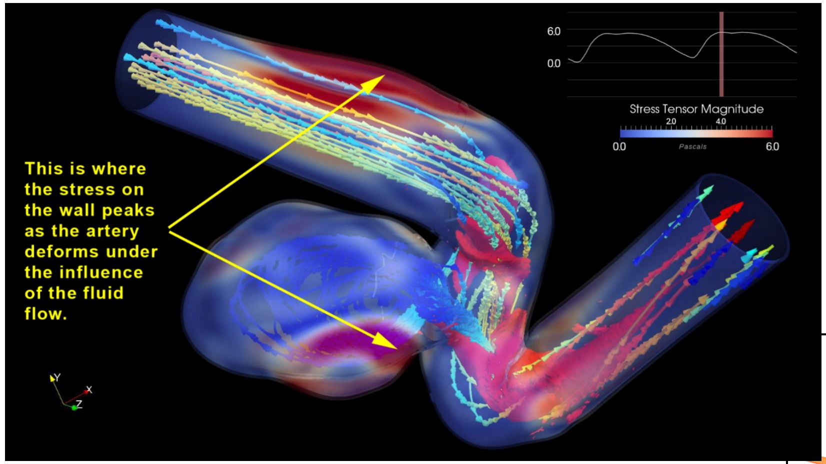
2023 Rendered on Aurora



2023 Rendered on Aurora

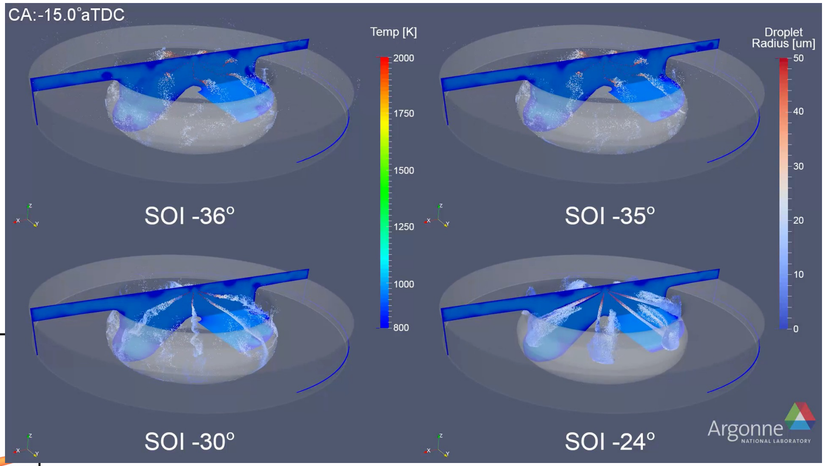
SCxx Visualization Showcase Finalists

2014: Blood Flow: Fluid Structure Interaction



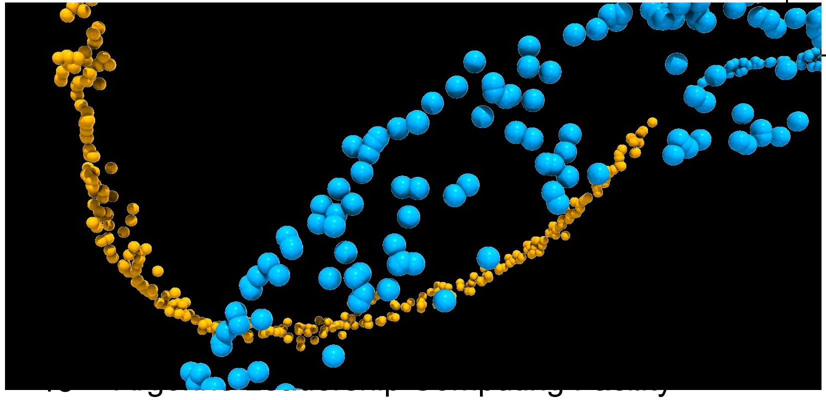
This is where the stress on the wall peaks as the artery deforms under the influence of the fluid flow.

2015: Gas Compression Ignition

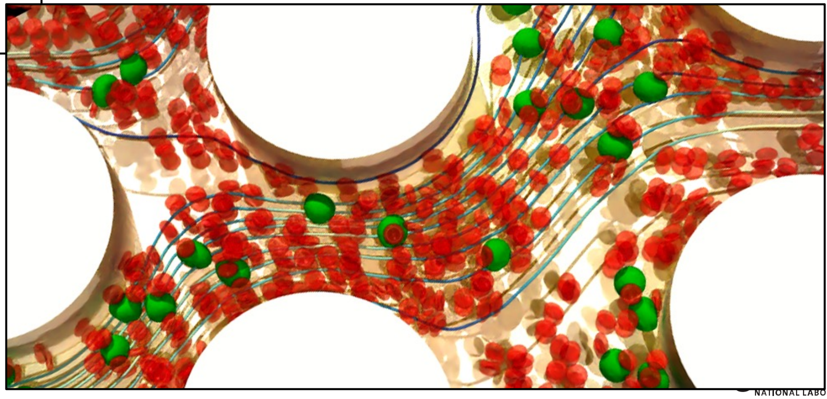


2018: Massive Stars

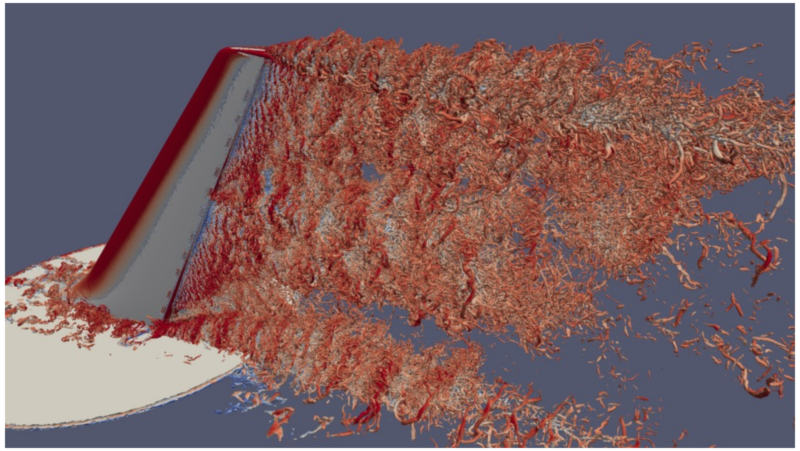
2019: Deep Learning



2020: Blood Flow: Circulating Tumor Cells



In Situ

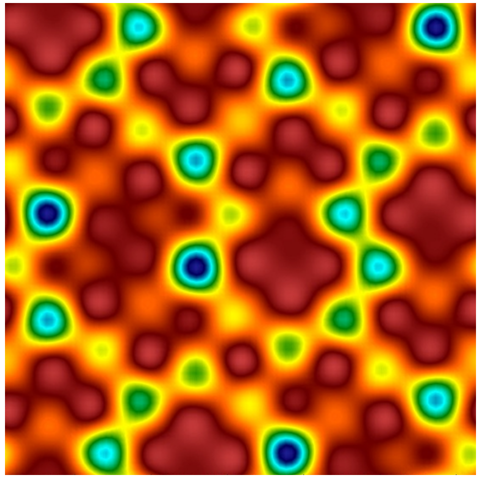


~2014

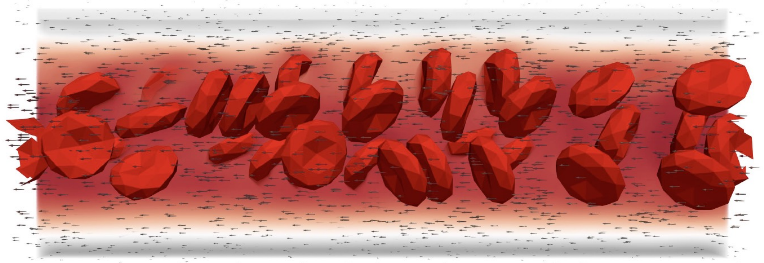
PHASTA, Catalyst,
Ken Jansen

2018

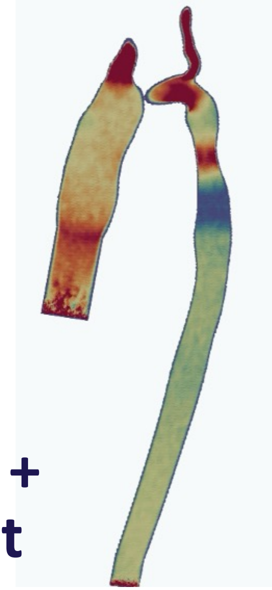
Nek5000,
SENSEI



2021 - 2024



Palabos+LAMMPS,
SENSEI + Catalyst,
bi-directional

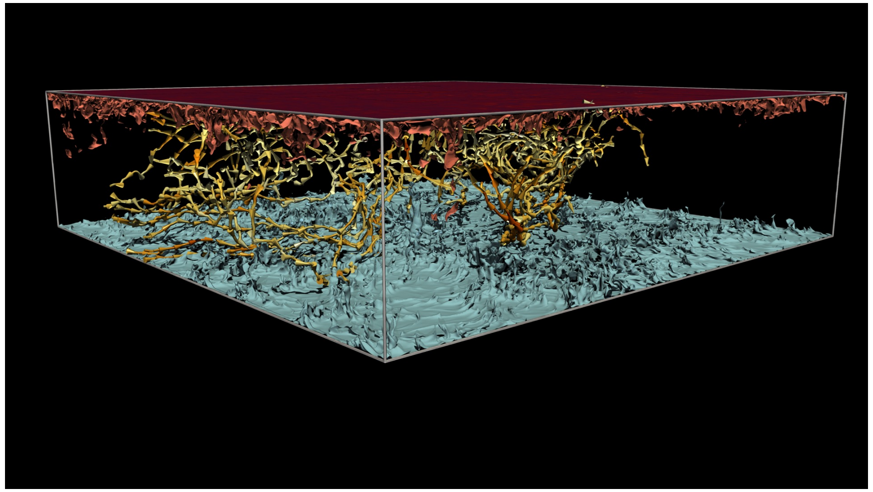


2019

SENSEI +
Catalyst

2024

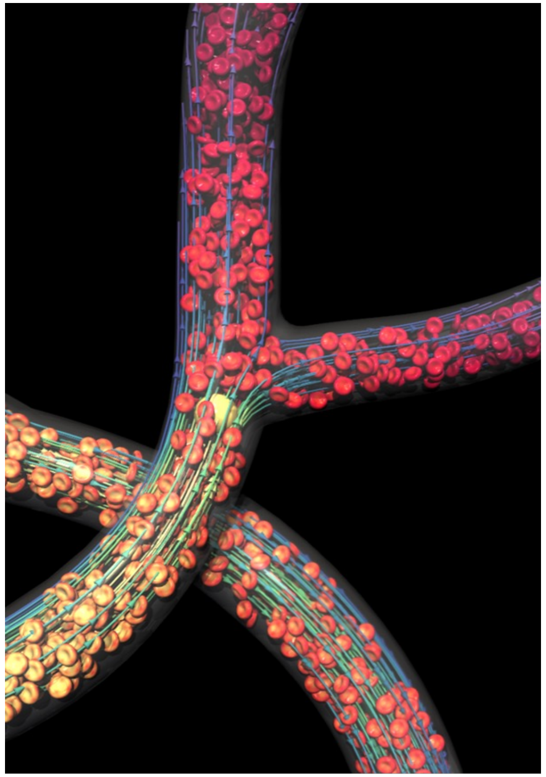
nekRS,
Ascent +
Catalyst



HARVEY

Ascent +
Catalyst

2024



THANKS

- ALCF Visualization and Data Analysis Team
  - Silvio Rizzi, Victor Mateevitsi, Janet Knowles, Geng Liu, Michael Papka

- Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357.

QUESTIONS?

Joe Insley
insley@anl.gov