Scientific Visualization at the Argonne Leadership Computing Facility

Joseph Insley
Argonne National Laboratory
Northern Illinois University

Silvio Rizzi
Argonne National Laboratory

Victor Mateevitsi
Argonne National Laboratory

EPICURE webinar **July 9, 2024**

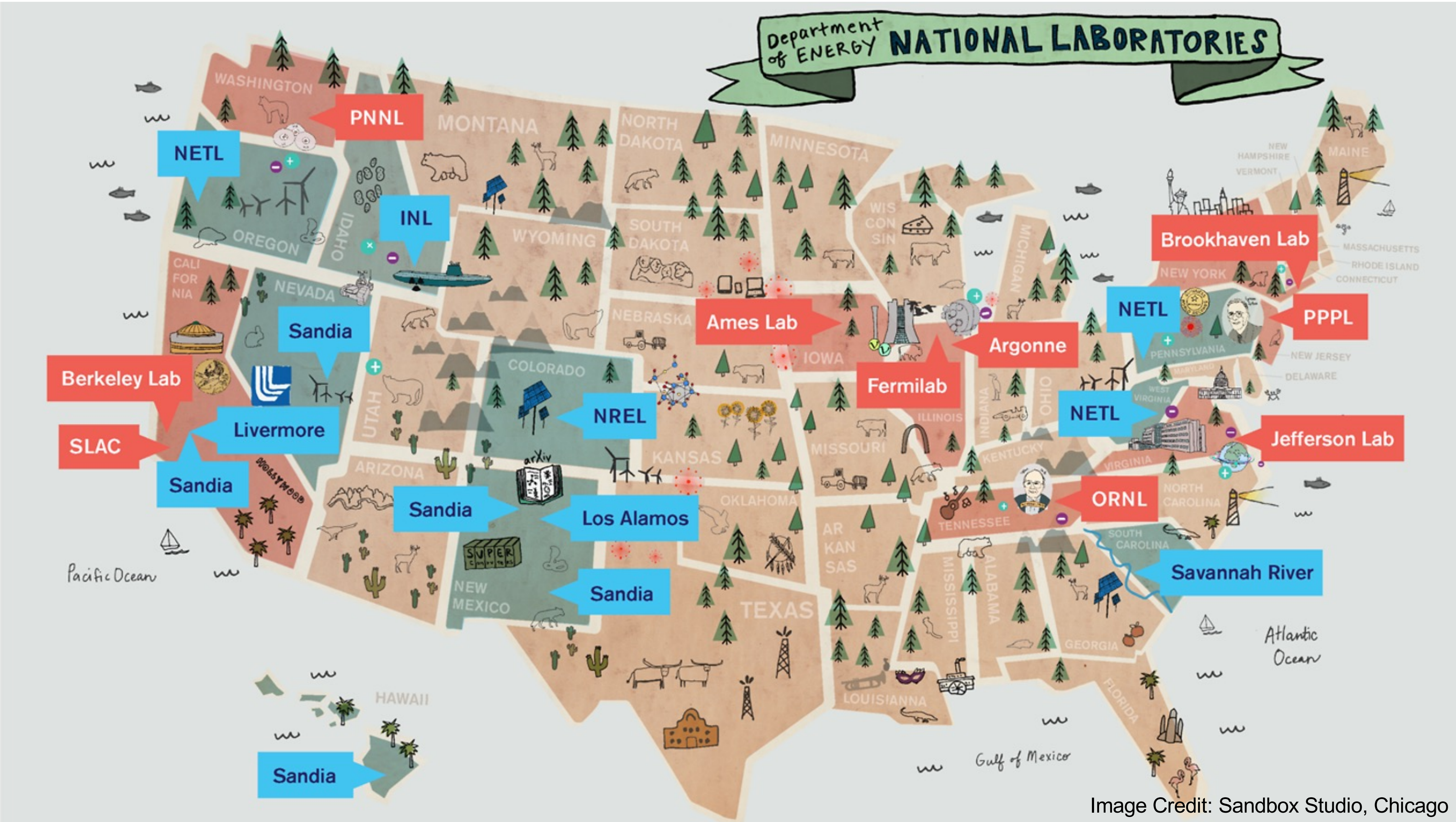
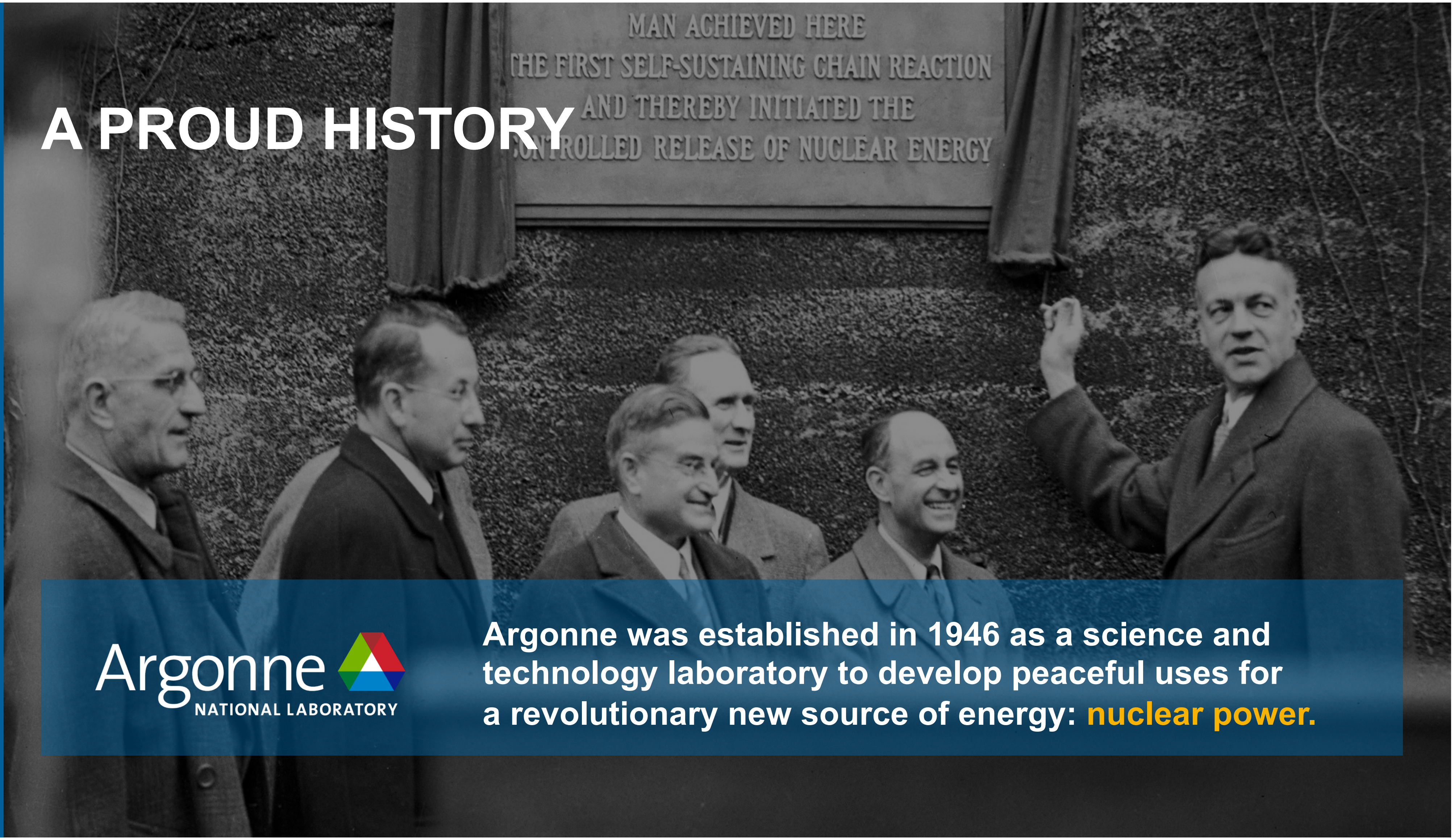


Image Credit: Sandbox Studio, Chicago

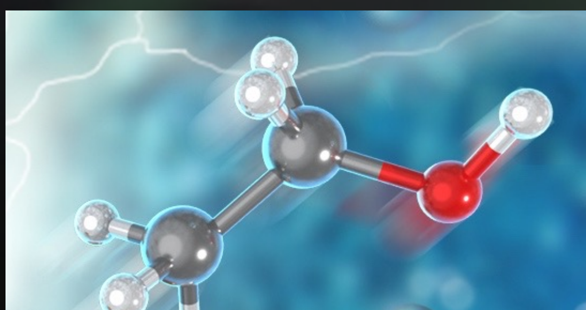
A PROUD HISTORY



Argonne NATIONAL LABORATORY

Argonne was established in 1946 as a science and technology laboratory to develop peaceful uses for a revolutionary new source of energy: nuclear power.

Argonne accelerates science and technology for U.S. prosperity and security.



Pivotal discoveries

Innovative collaborations

Unparalleled facilities

Our signature contributions

To science and society

Scientific discoveries that solve the deepest mysteries in the physical and life sciences

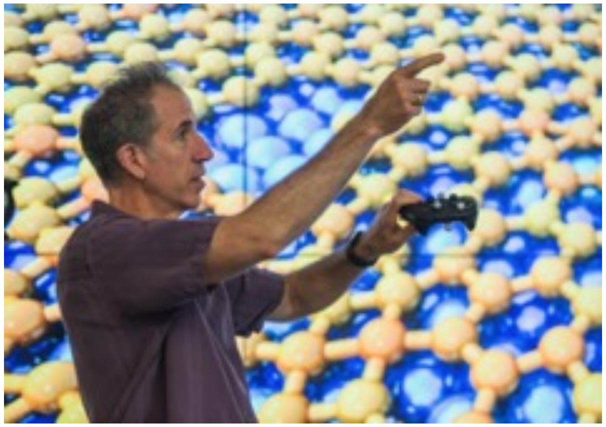
Energy and climate solutions that improve the quality of human life and preserve our planet

Global security advances that protect society from diverse natural and anthropogenic threats

Cutting-edge research facilities that support thousands of scientists and engineers from around the world

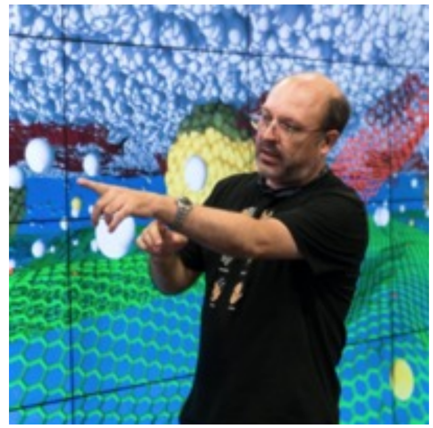
Development of the STEM workforce and its leaders

ALCF VISUALIZATION AND DATA ANALYTICS GROUP



Joe Insley
Team Lead

Janet Knowles
Principal Software Engineering Specialist

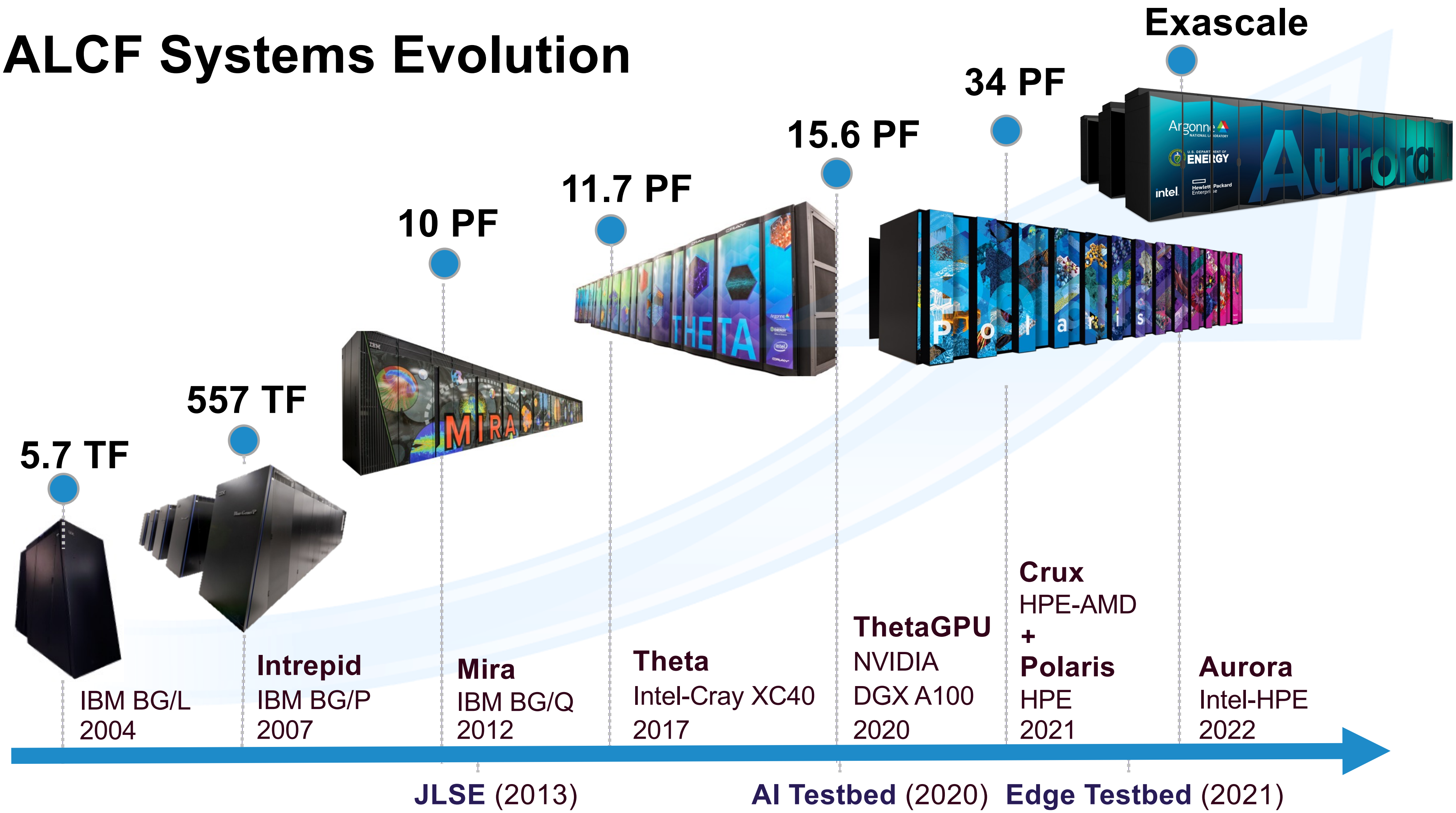


Silvio Rizzi
Computer Scientist

Victor Mateevitsi
Assistant Computer Scientist

Geng Liu
Postdoctoral Appointee

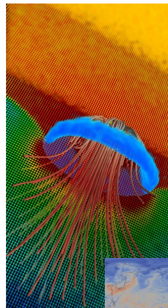
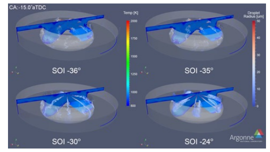
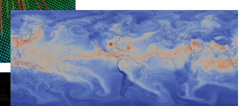
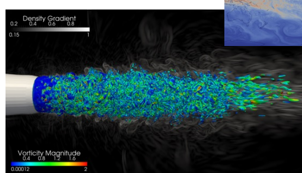
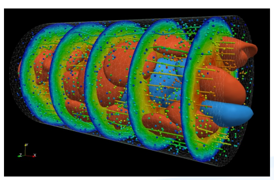
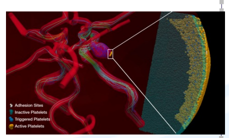
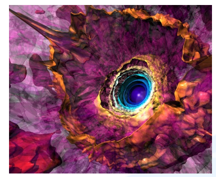
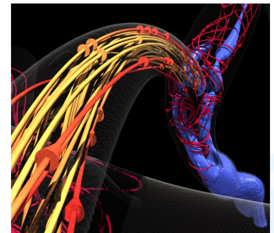
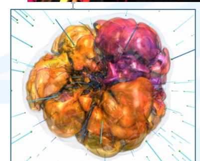
ALCF Systems Evolution

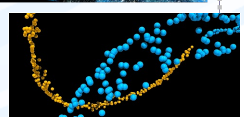
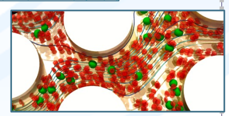
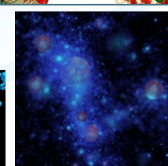
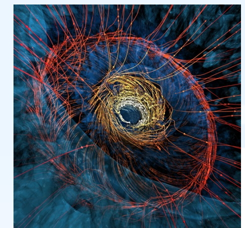
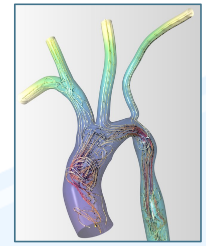
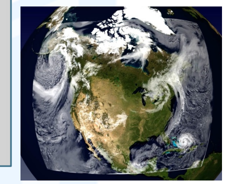
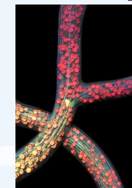
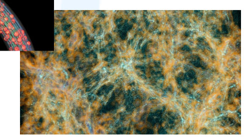
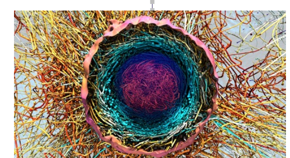
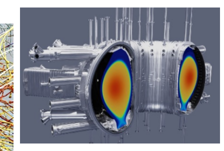
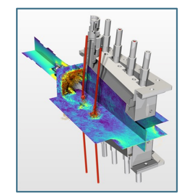


ALCF Systems Evolution

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

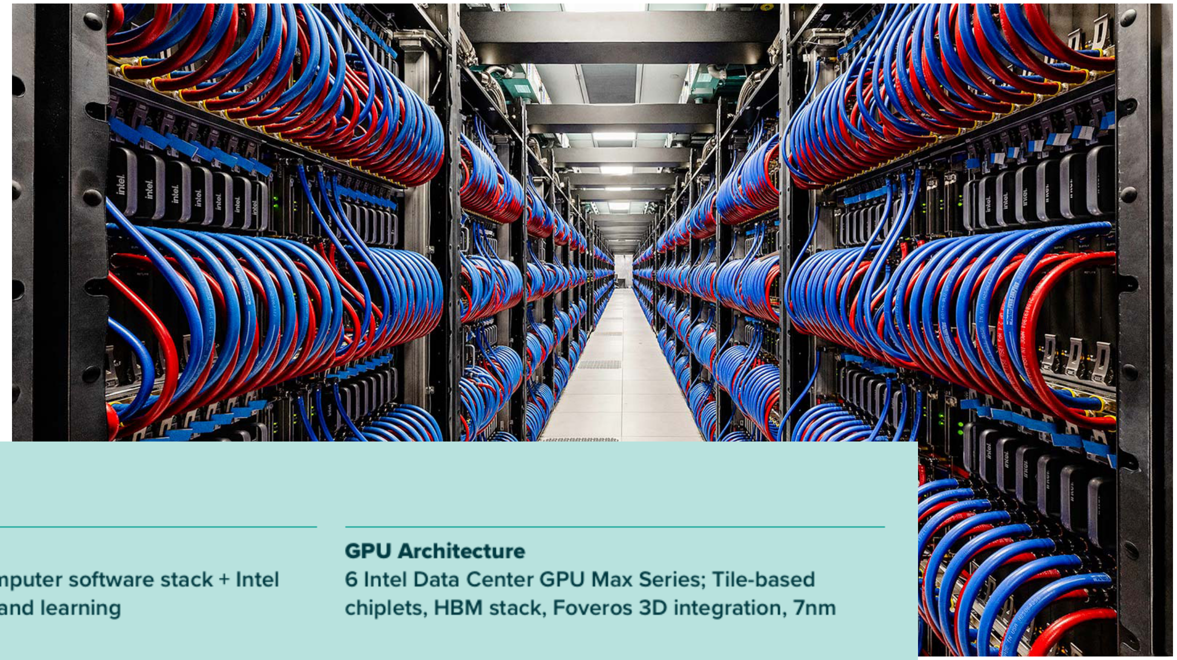
JLSE (2013) **AI Testbed** (2020) **Edge Testbed** (2021)

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

AURORA

1.012 EF
#1 HPL-MxP (10.6 EF)
#2 Top500



Aurora System Specifications

Compute Node
2 Intel Xeon CPU Max Series processors: 64GB HBM on each, 512GB DDR5 each; 6 Intel Data Center GPU Max Series, 128GB HBM on each, RAMBO cache on each; Unified Memory Architecture; 8 SlingShot 11 fabric endpoints

Software Stack
HPE Cray EX supercomputer software stack + Intel enhancements + data and learning

GPU Architecture
6 Intel Data Center GPU Max Series; Tile-based chiplets, HBM stack, Foveros 3D integration, 7nm

CPU-GPU Interconnect
CPU-GPU: PCIe; GPU-GPU: Xe Link

System Interconnect
Slingshot 11; Dragonfly topology with adaptive routing; Peak Injection bandwidth 2.12 PB/s; Peak Bisection bandwidth 0.69 PB/s

Network Switch
25.6 Tb/s per switch, from 64–200 Gbs ports (25 GB/s per direction)

System Performance
Exascale

High-Performance Storage
230 PB, 31 TB/s, 1024 Nodes (DAOS)

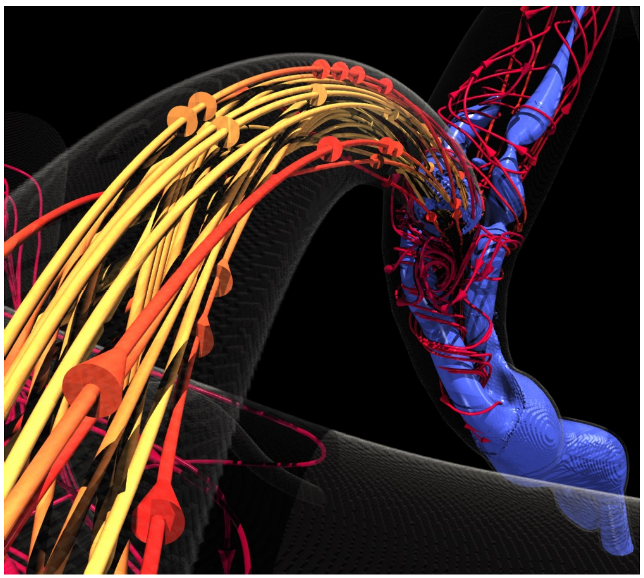
Programming Models
Intel oneAPI, MPI, OpenMP, C/C++, Fortran, SYCL/DPC++

Platform
HPE Cray EX supercomputer

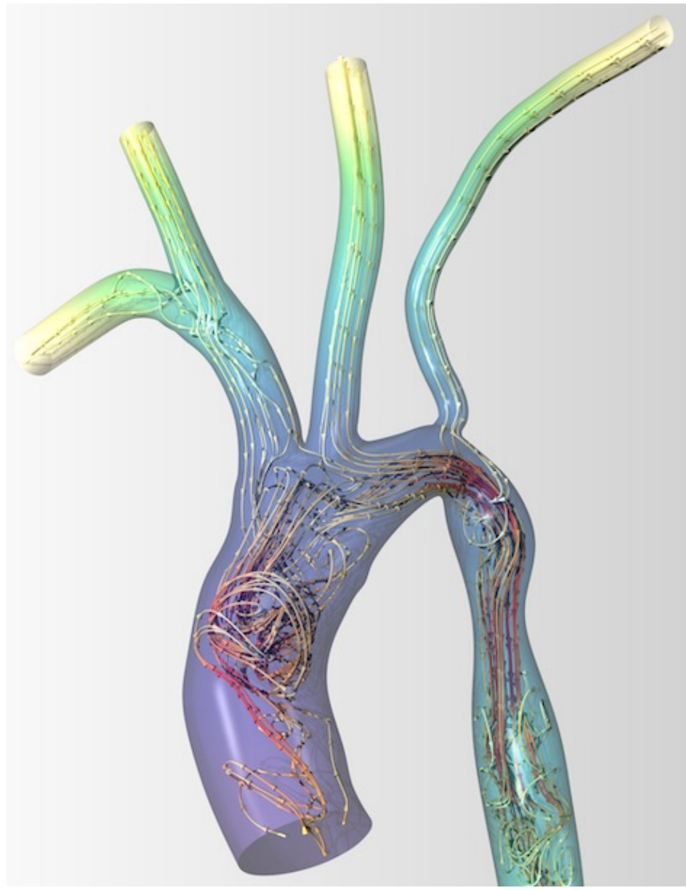
Aggregate System Memory
10.9 PB

System Size
10,624 nodes

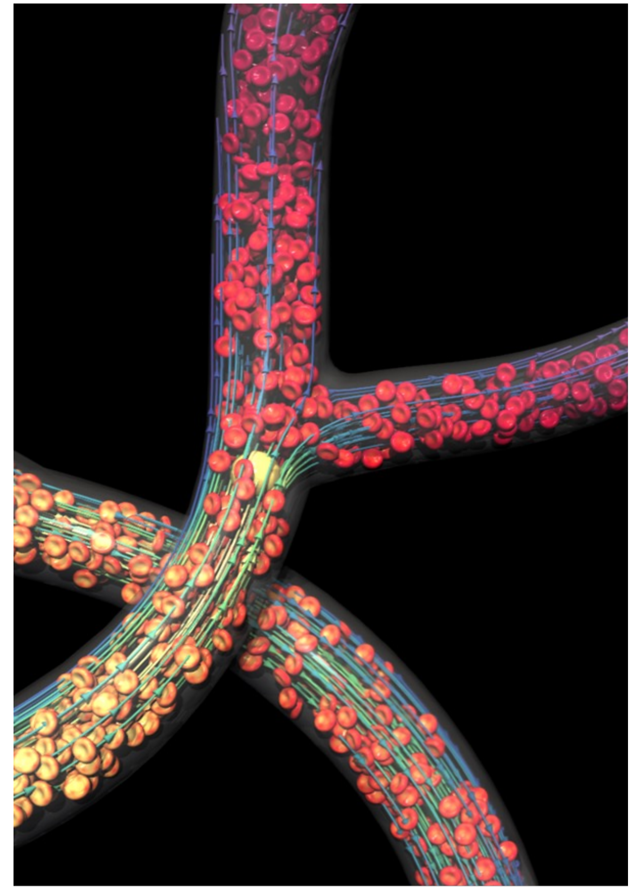
PI: Amanda Randles, Duke University



2020



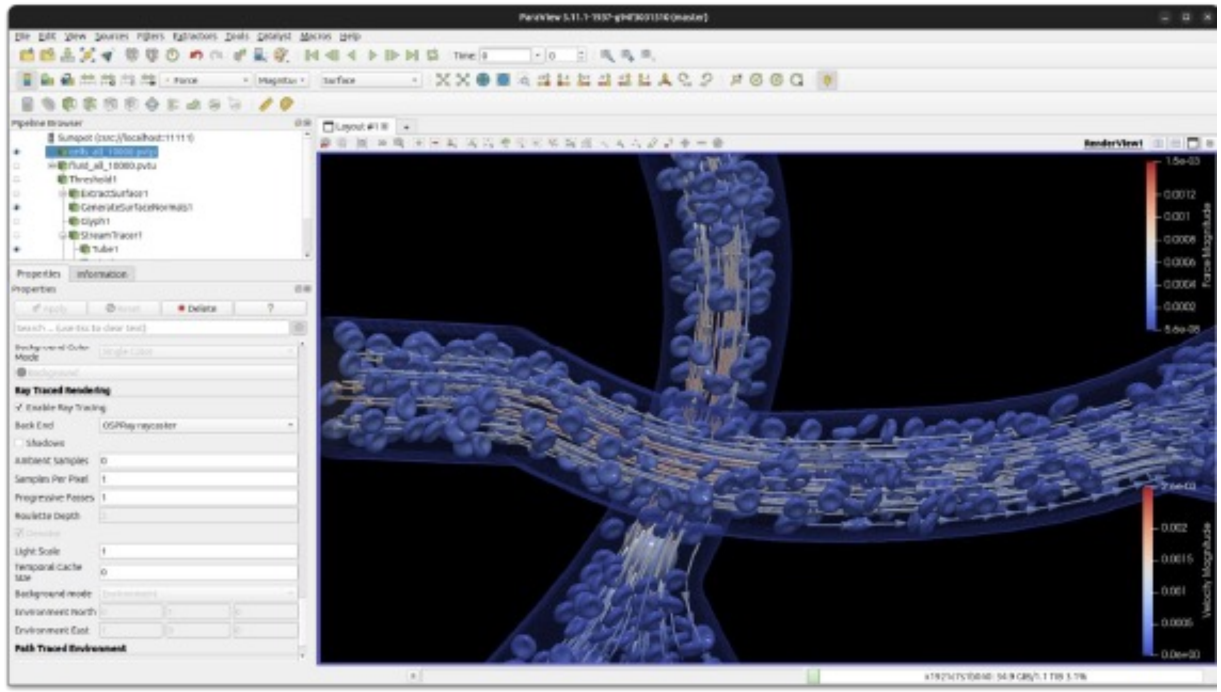
2023 Rendered on Aurora



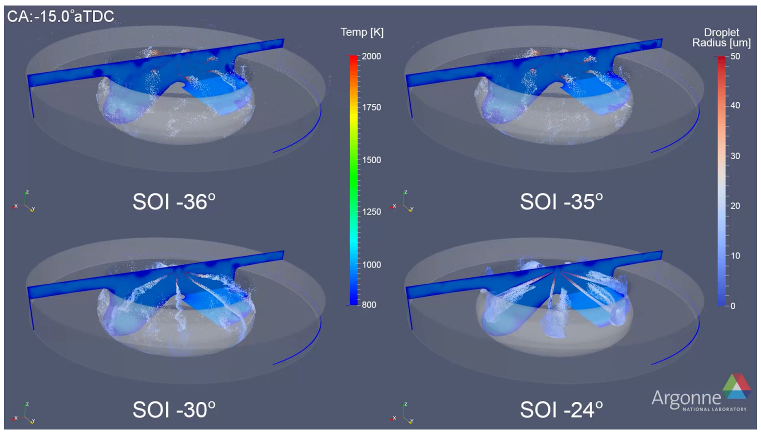
2023 Rendered on Aurora

SC23 Live Demo

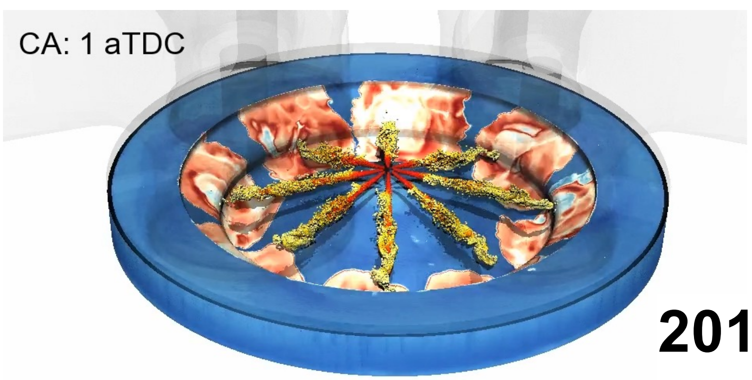
- ParaView server running on 16 nodes (96 GPUs) on Sunspot
- ParaView client running on SC23 show floor



PI: Sibendu Som, Argonne National Laboratory

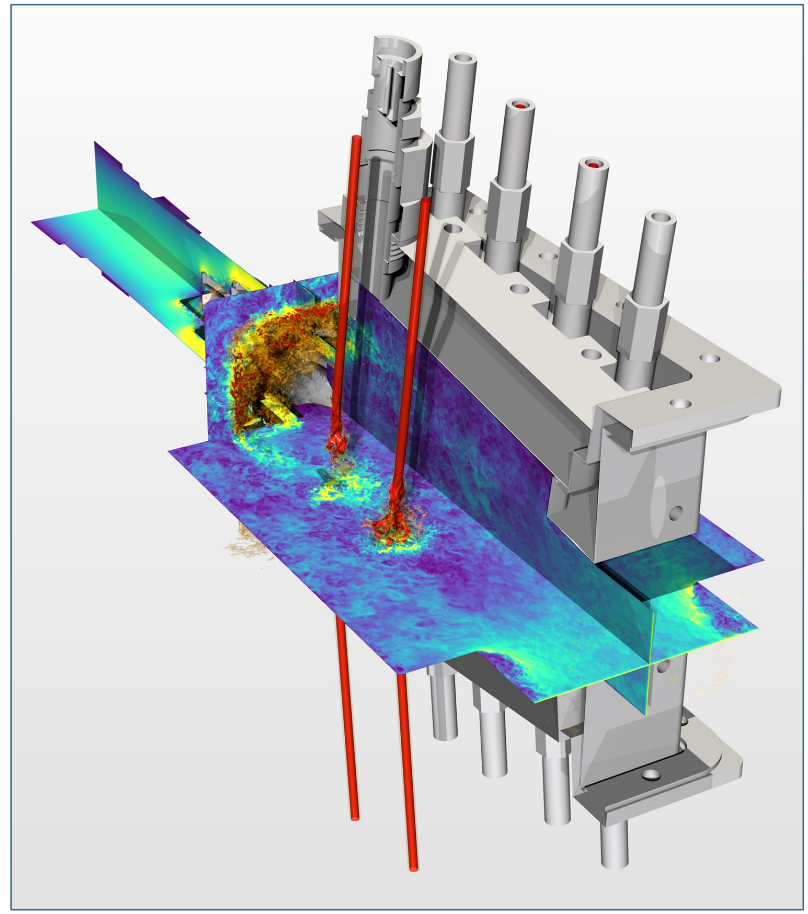


2015

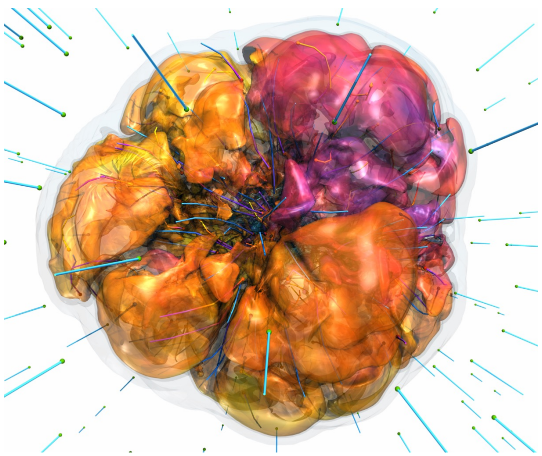


2017

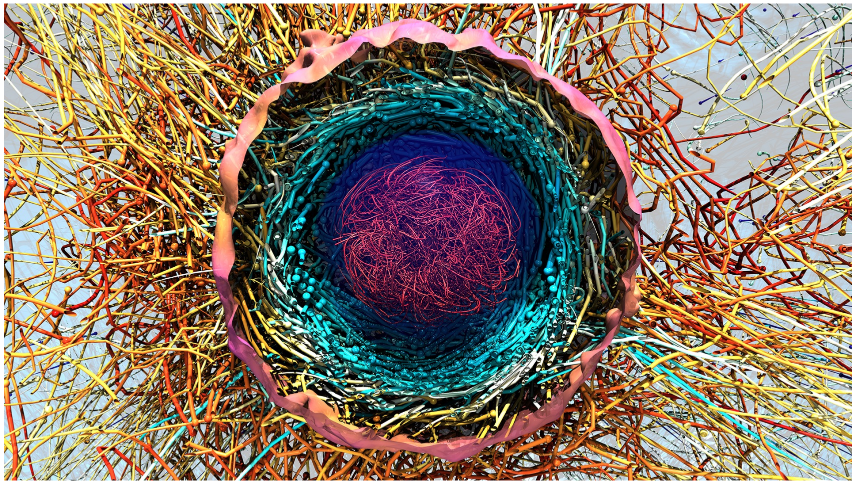
2023



PI: Adam Burrows, Princeton University

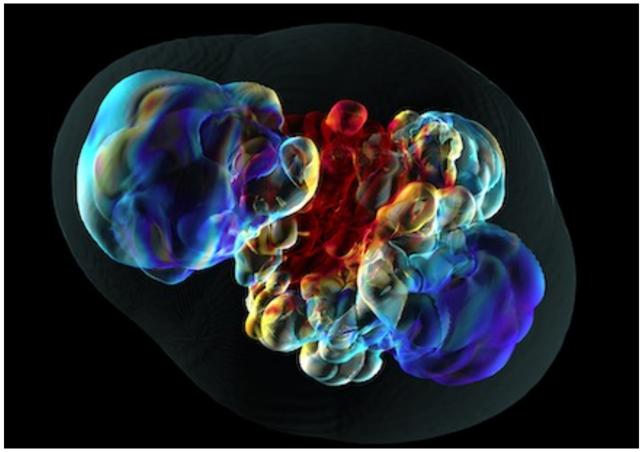


2019

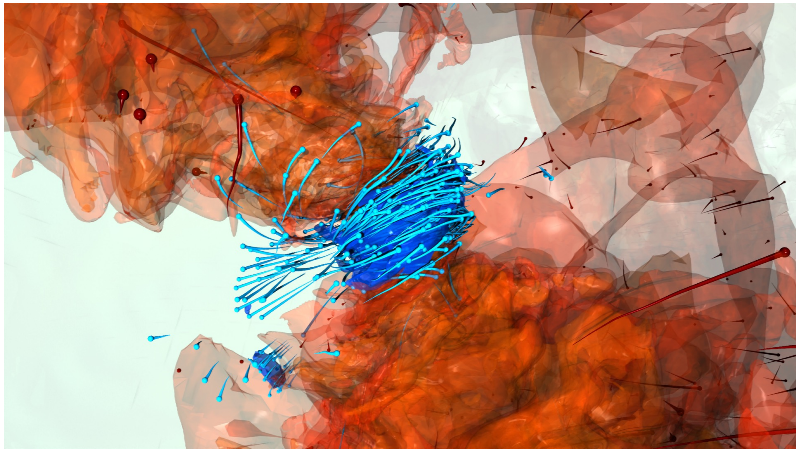


2021

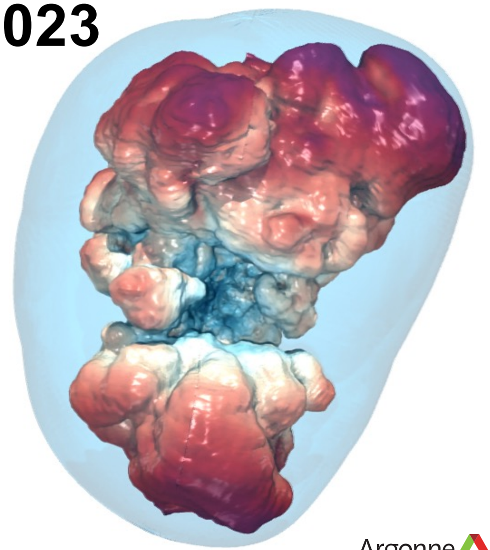
2022



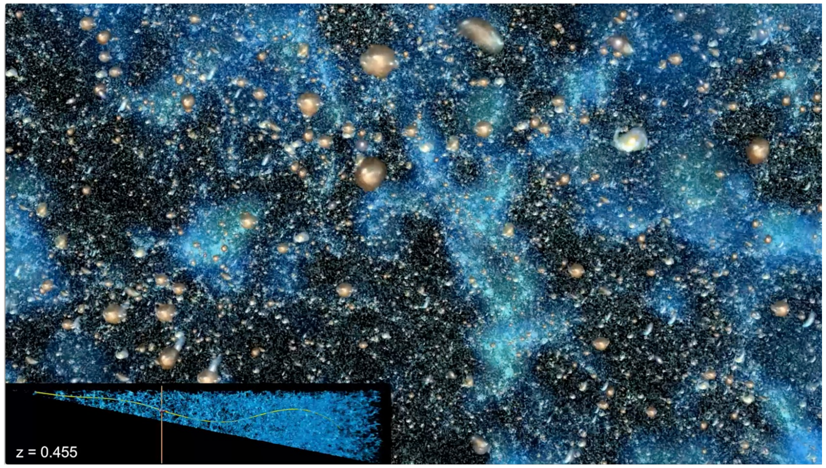
2023



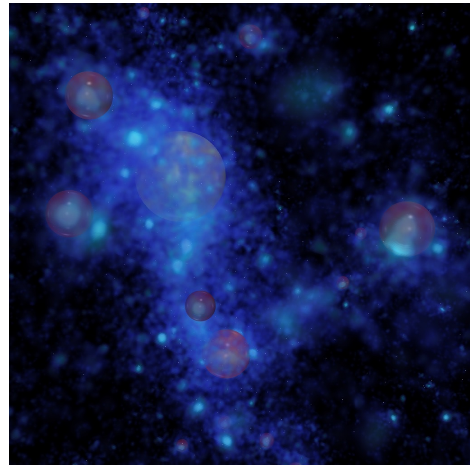
2023



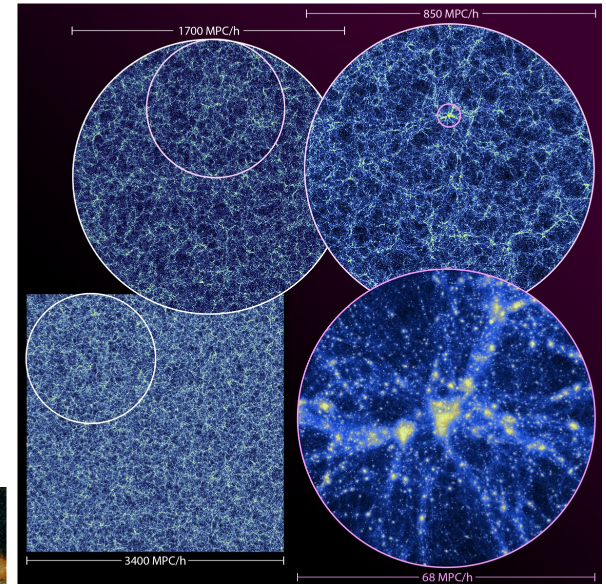
2018

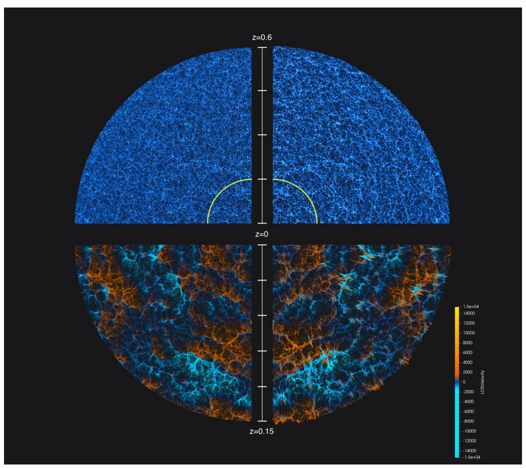


2020

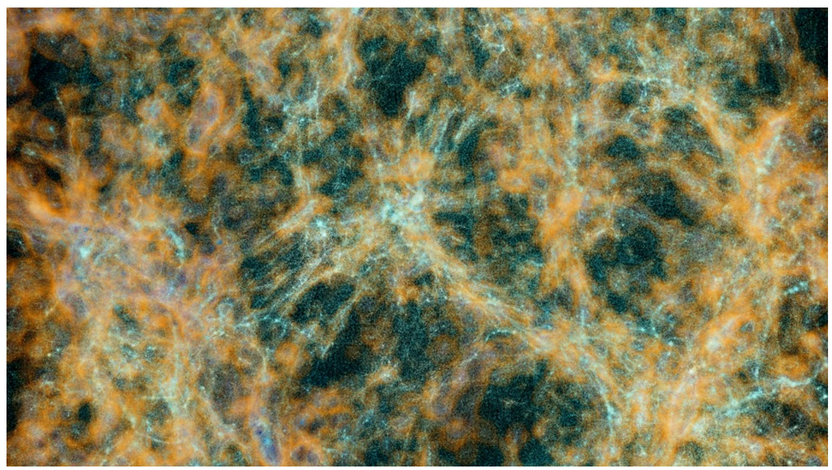


2020





2021



Computed and Rendered on Aurora **2023**

PI: Salman Habib and HACC Team, Argonne National Laboratory

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

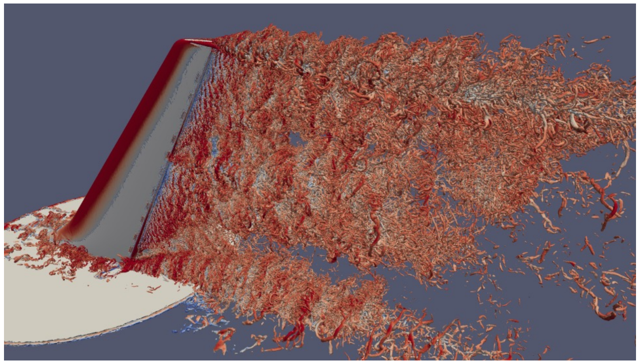
- I/O is too expensive
- Scientists cannot save every timestep, and/or resolution
- Lost cycles: simulation waits while I/O is happening
- Lost discoveries: scientists might miss discoveries

- Solution: *In situ* visualization and analysis

WHAT IS *IN SITU*

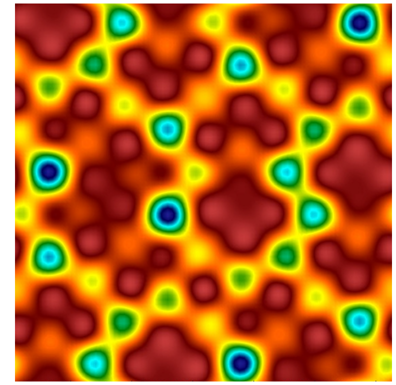
- Traditionally visualization and analysis happens post hoc
  - aka: Data gets saved to the disk, scientist opens it after the simulation has ended
- *In situ*
  - Data gets visualized/analyzed **while** in memory.
  - If zero-copy used, there is no data movement
  - Ideally the data is on the GPU and stays on the GPU

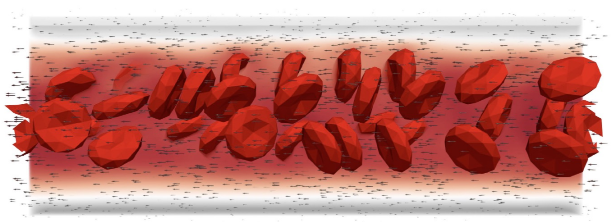
In Situ



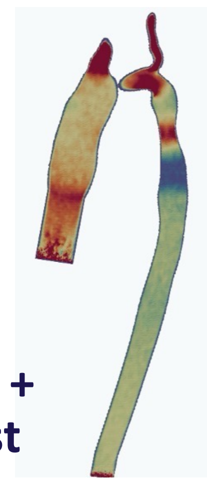
~2014
PHASTA, Catalyst, Ken Jansen

2018
Nek5000, SENSEI



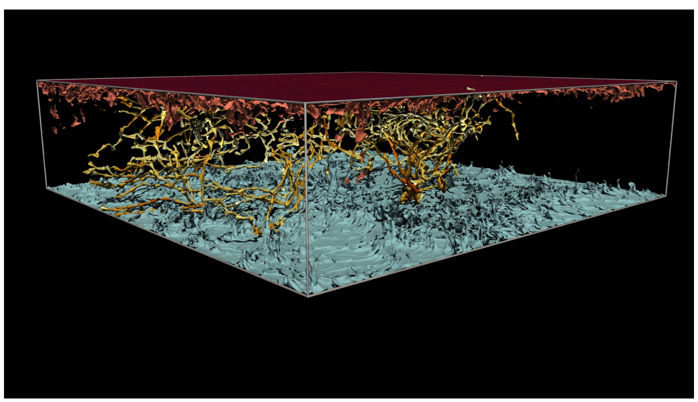


2021 - 2024
Palabos+LAMMPS, SENSEI + Catalyst, bi-directional



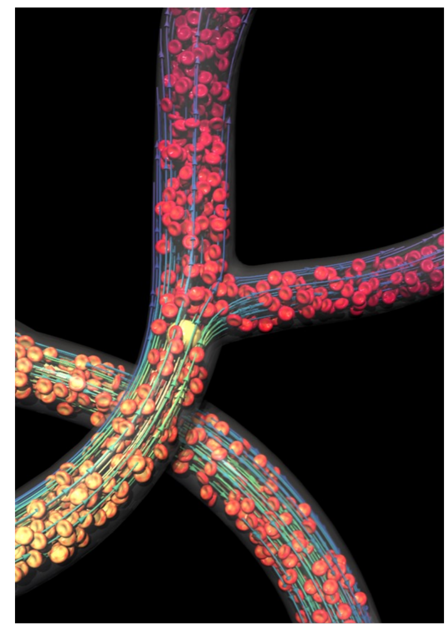
2019
SENSEI + Catalyst

2024
nekRS, Ascent + Catalyst



HARVEY

Ascent + Catalyst
2024



SCALING COMPUTATIONAL FLUID DYNAMICS: IN SITU VISUALIZATION OF NEKRS USING SENSEI

U.S. DEPARTMENT OF ENERGY
Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

Argonne NATIONAL LABORATORY

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

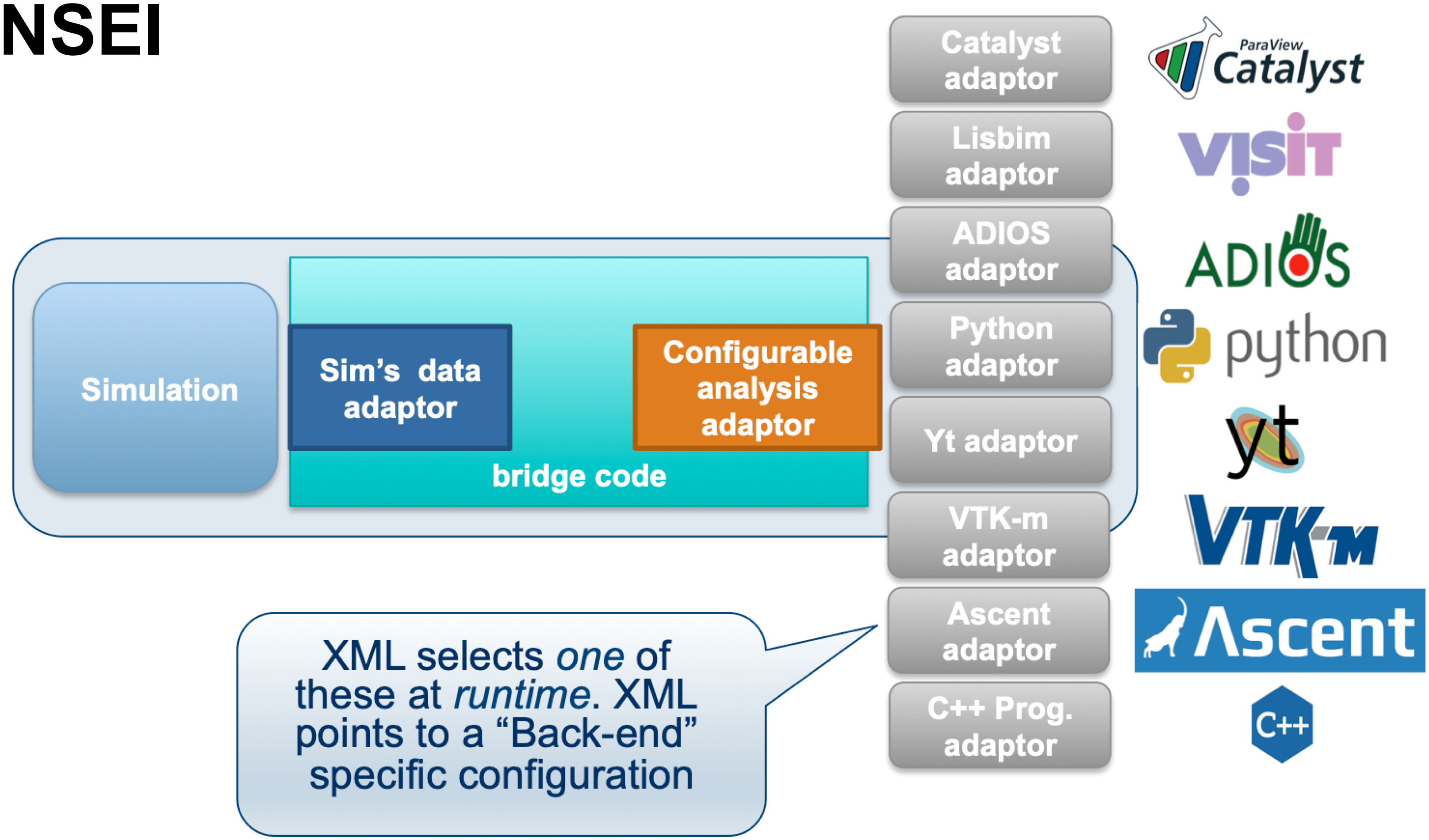
- NekRS
  - Rooted in the Spectral Element Method (SEM)
  - GPU-accelerated thermal-fluid simulation code
  - Predecessor is Nek5000
  - Supports modern heterogenous systems (CPU/GPU)

- 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



Simulation
Sim's data adaptor
Configurable analysis adaptor
bridge code
Catalyst adaptor
Lisbim adaptor
ADIOS adaptor
Python adaptor
Yt adaptor
VTK-m adaptor
Ascent adaptor
C++ Prog. adaptor
ParaView Catalyst
VisIt
ADIOS
python
yt
VTK-m
Ascent
C++
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]

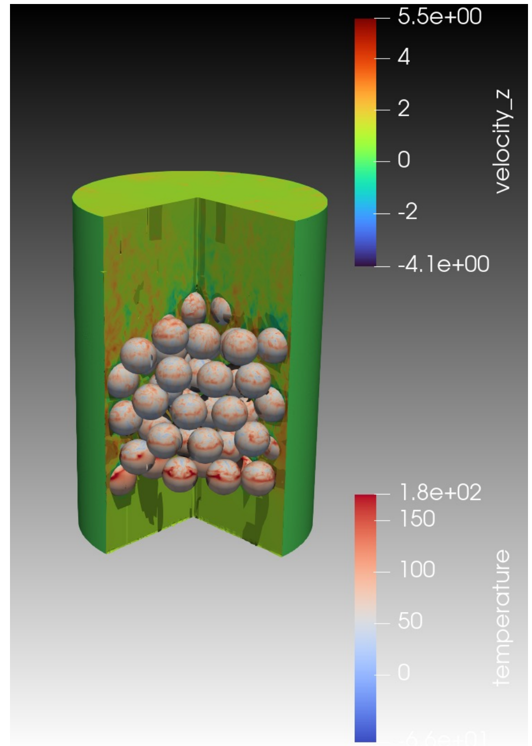
1. Victor A. Mateevitsi, Mathis Bode, Nicola Ferrier, Paul Fischer, Jens Henrik Göbbert, Joseph A. Insley, Yu-Hsiang Lan, Misun Min, Michael E. Papka, Saumil Patel, Silvio Rizzi, and Jonathan Windgassen. 2023. Software and Analysis for paper: Scaling Computational Fluid Dynamics: In Situ Visualization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

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

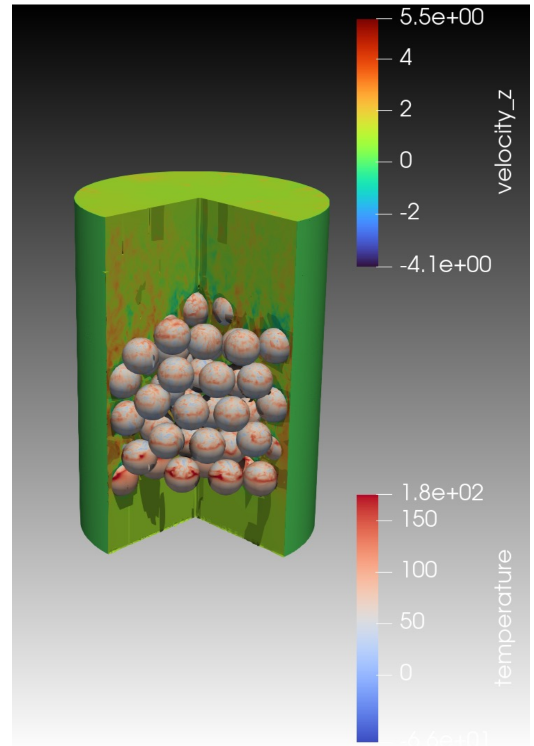


Visualization of the pb146 use case simulation, illustrating flow dynamics within a pebble-bed nuclear reactor

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)



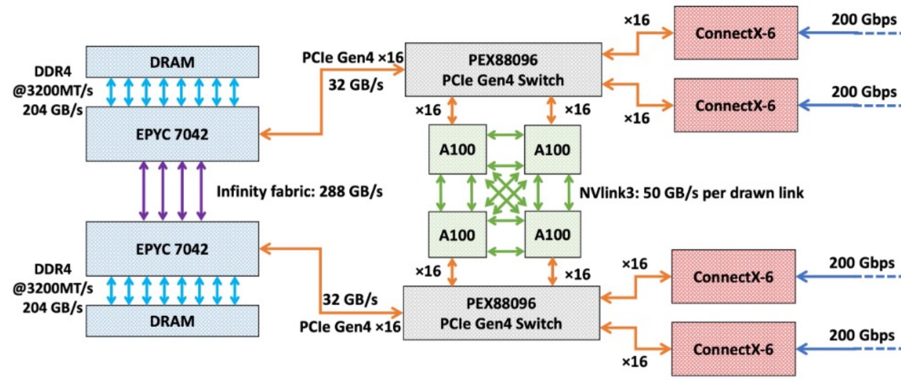
Visualization of the pb146 use case simulation, illustrating flow dynamics within a pebble-bed nuclear reactor

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



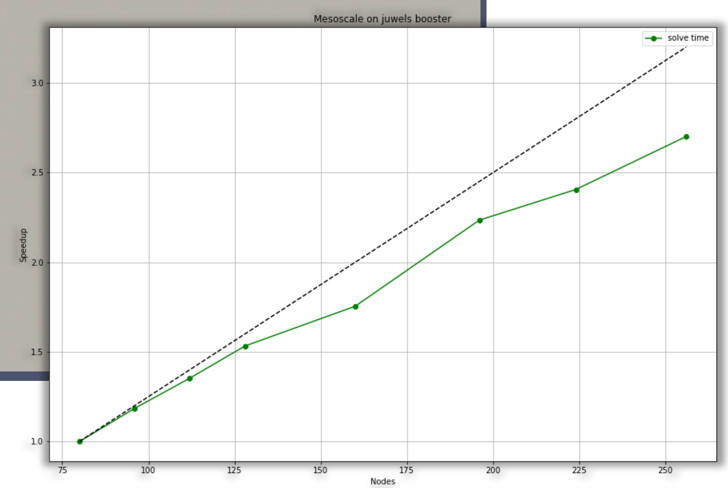
RESULTS – IN TRANSIT MESOSCALE CASE

- **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



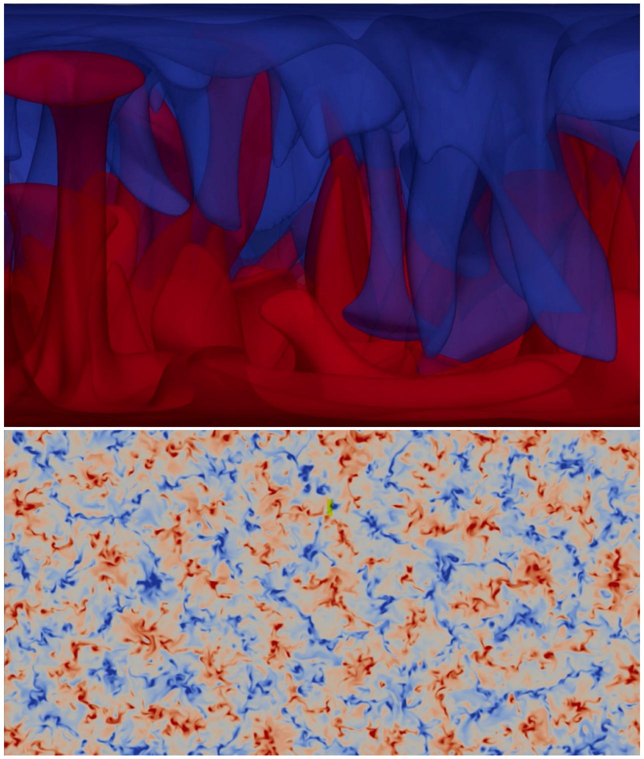
Strong-scaling plot for JUWELS Booster

[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



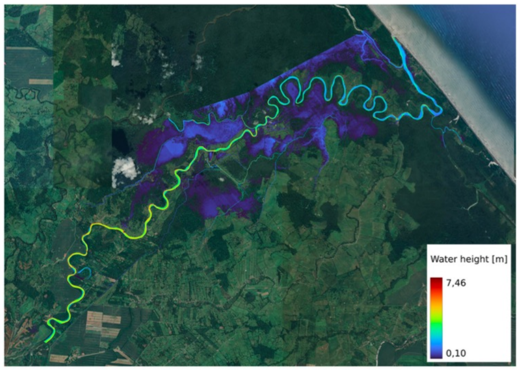
Visualization of the RBC case.
A side view and a top view colored by temperature.

RESEARCHERS FROM CENAT (COSTA RICA) ENABLING IN SITU VIS ON POLARIS

U.S. DEPARTMENT OF ENERGY
Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

Argonne NATIONAL LABORATORY

In-Situ Visualization for River Simulations using GPUs

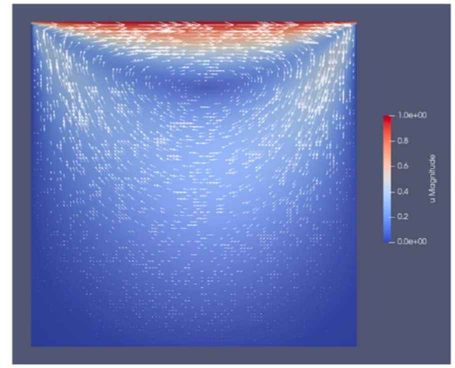


- Aids flood risk assessment in Costa Rica, which has a dense hydrographic network.
- Uses the SERGHEI code base to solve the Shallow Water Equations.
- Main goal is to add an in-situ visualization SERGHEI with the Ascent and Catalyst frameworks.
- Working on testing its performance portability across GPU architectures.

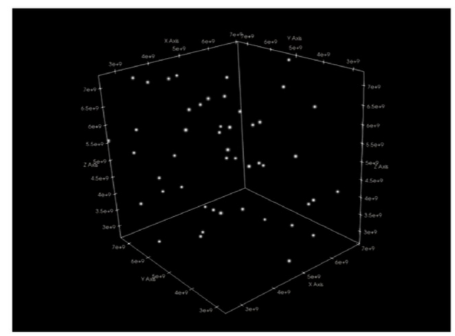
Johansell Villalobos
National Center of High Technology - Costa Rica
Contact: jovillalobos@cenat.ac.cr
LinkedIn: johansellvilla

CeNAT

Performance portability Evaluation

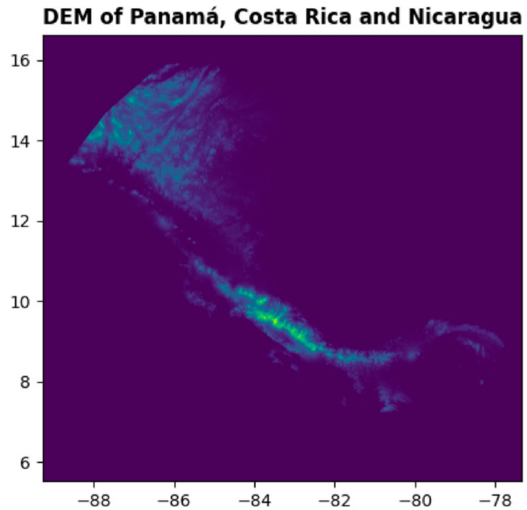


- This work aims to port two simulation mini apps to the Kokkos, OpenMP target, OCCA, and RAJA performance portability libraries to evaluate their performance at scale. A statistical evaluation will then be carried out to quantify the differences between them.



kokkos OpenMP OCCA Performance Portability Transparency RAJA

In-situ analysis and visualization of seismic simulations



Christian Asch Burgos
National Center of High Technology - Costa Rica
Contact: casch@cenat.ac.cr

- Extension of the SeisSol simulation to integrate in-situ capabilities.
- We want to add in-situ visualization and steering to this software.
- We will work with elevation data from the Nicaragua-Costa Rica-Panamá region to validate the simulation.

BRIDGING GAPS IN SIMULATION ANALYSIS THROUGH A GENERAL-PURPOSE, BIDIRECTIONAL STEERING INTERFACE

U.S. DEPARTMENT OF ENERGY
Argonne National Laboratory is a
U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC.

Argonne
NATIONAL LABORATORY

BACKGROUND

ASCENT

- Easy-to-use flyweight in situ visualization and analysis library
- Uses Conduit for describing data
- Supports zero-copy GPU-GPU
- Uses VTK-m as a toolkit for scientific visualization algorithms
- Has a python and jupyter interface

BACKGROUND

Bidirectional steering

SENSEI insitu

ParaView Catalyst

VisIt Libsim

Drawbacks

- Limited steering functionality
- Do not allow sending back modified data
- Require changes in the code of the GUI version, thus making development more difficult

MOTIVATION

Interviews with scientists

- 7 scientists at Argonne National Laboratory
  - Computational Fluid Dynamics
  - Computational Physics
  - Water Resource Engineering
  - Environmental Science
  - Computational Biology
  - Fluid Structure Interactions

MOTIVATION

Interview Insights

- Steering
  - Change parameters (velocity, temperature, pressure, etc.)
  - Adjust resolution
  - Add/modify geometry
  - Modify simulation data
- Exploration and Discovery
  - Checking on simulation status
  - Interactively exploring data and changing filters/cameras
  - Trial and error
- External Notifications
  - Be alerted when things have gone wrong
  - Manually investigate interesting phenomena

CONTRIBUTIONS

General Purpose Steering Interface

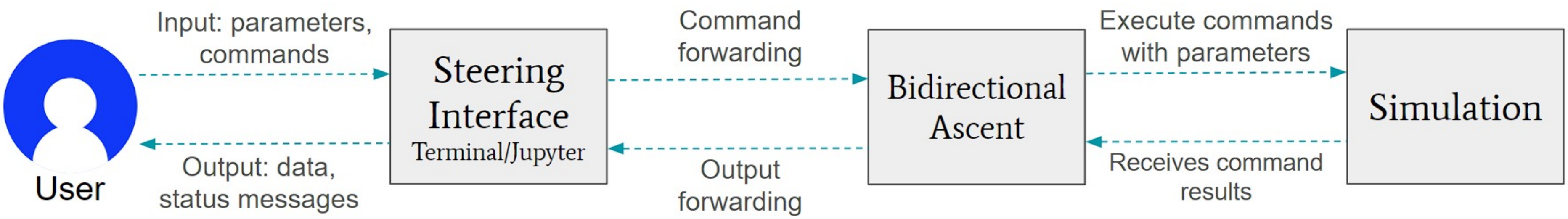
- Build upon Ascent
  - — Lightweight.
  - — Scalable.
  - — Convenient instrumentation with existing simulation codes.
- Let users define their own steering behaviors
  - — Function callbacks give users total control over a simulation.
- Bidirectional interactive interface
  - — Use Ascent's existing remote Jupyter notebook capabilities.

BONUS CONTRIBUTION

Shell commands

```
-
  action: "add_commands"
  commands:
    c1:
      params:
        shell_command: echo "Unstable!" | mail -s "Unstable Simulation" $email
        mpi_behavior: "root"
```

Can run on the root node, or on all nodes

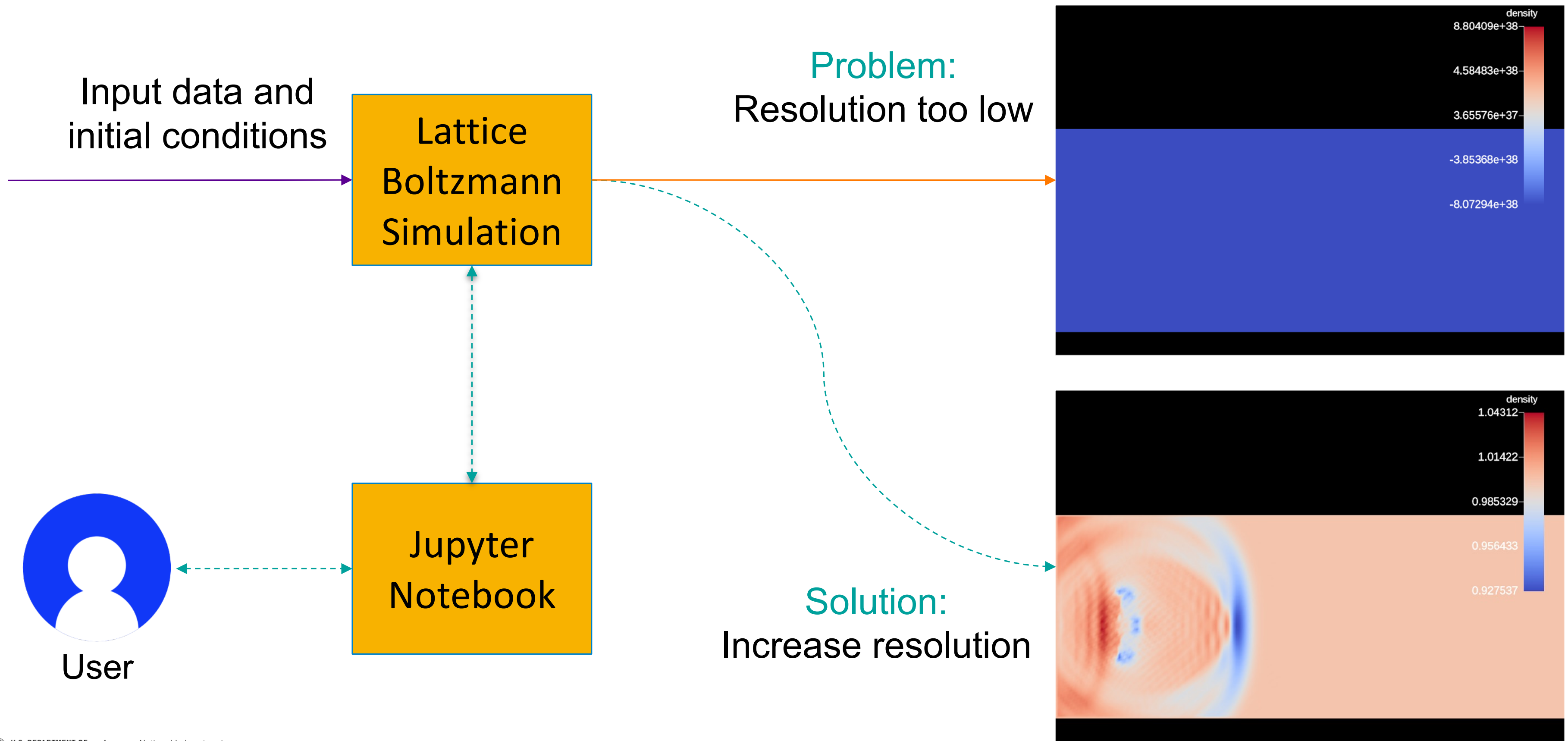

Input: parameters,
commands
User
Output: data,
status messages
Steering
Interface
Terminal/Jupyter
Command
forwarding
Output
forwarding
Bidirectional
Ascent
Execute commands
with parameters
Receives command
results
Simulation

COMPARISON OF IN SITU FRAMEWORKS BY STEERING CAPABILITY

Steering Type	Catalyst	SENSEI	Ascent	Ascent w/ bidirectional
Simulation Variables	Yes	Partial	Partial	Yes
Internal Commands	No	No	No	Yes
External Commands	No	No	No	Yes
Modifying data	No	No	No	Yes

USE CASE

Instability



USE CASE

Workflow

1. Detect instability: simulation

2. Pause simulation: Ascent

3. Notify the user: Ascent

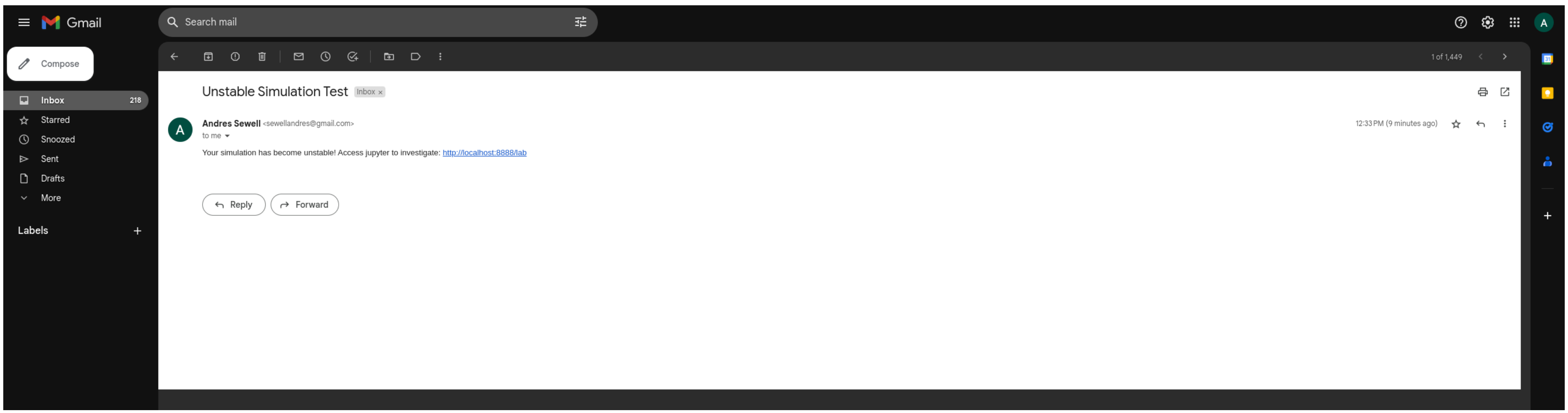
```yaml
-
  action: "add_commands"
  commands:
    c1:
      params:
        callback: "setStability"
        mpi_behavior: "all"
-
  action: "add_triggers"
  triggers:
    t1:
      params:
        callback: "isStable"
        actions_file : "stable_actions.yaml"
    t2:
      params:
        callback: "isUnstable"
        actions_file : "unstable_actions.yaml"
```

```yaml
-
  action: "add_commands"
  commands:
    c1:
      params:
        shell_command: echo "Unstable!" | mail -s "Unstable Simulation" $email
        mpi_behavior: "root"
```

Example Ascent actions

USE CASE

Workflow



USE CASE

Workflow

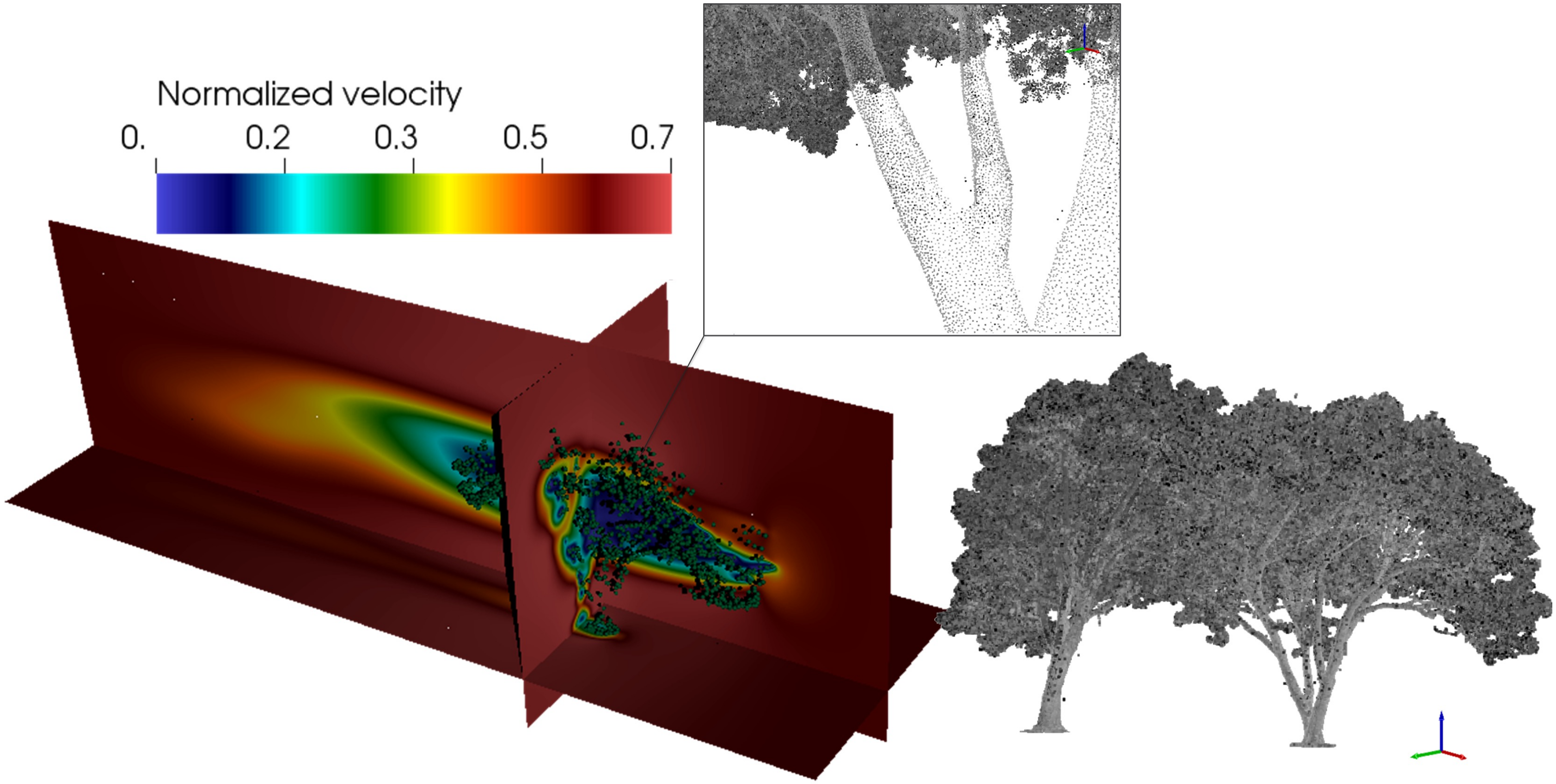
4. Access the steering interface: User
5. Restore checkpoint: User
6. Increase number of timesteps: User
7. Resume the simulation: User

```
# Connect to our simulation instance via Ascent, pausing it
%connect

# This is a function callback that takes no parameters and returns no output
execute_callback("restoreStableState", conduit.Node(), conduit.Node())

# This is a function callback that takes one parameter and returns no output
params = conduit.Node()
params["timesteps"] = 1000
execute_callback("setTimeSteps", params, conduit.Node())

# This is a function callback that takes no parameters and does return an output
output = conduit.Node()
execute_callback("setTimeSteps", conduit.Node(), output)
print(output)

# Disconnect from the simulation, resuming it
%disconnect
```

Example Jupyter notebook

SIMULATION OF WIND ACROSS A CITY

WIP



SIMULATION OF WIND ACROSS A CITY

- Use case
  - LIDAR of trees in a city
  - Simulating wind and effect of trees
- Problem
  - Case takes 20-30 mins to load
  - Scientist examines result and then modifies lidar resolution
  - This is a trial and error scenario. They don’t know a priori what numbers they should select

SIMULATION OF WIND ACROSS A CITY

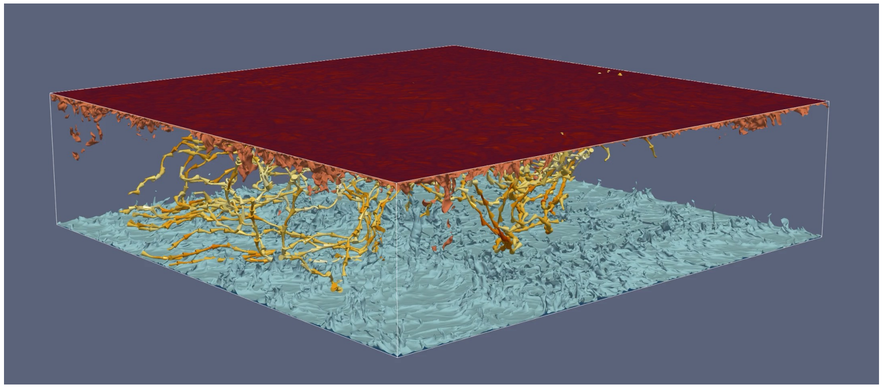
- Bidirectional solution
  - Load case once (20-30 mins)
  - Real-time bidirectional steering
    - Sees the results immediately

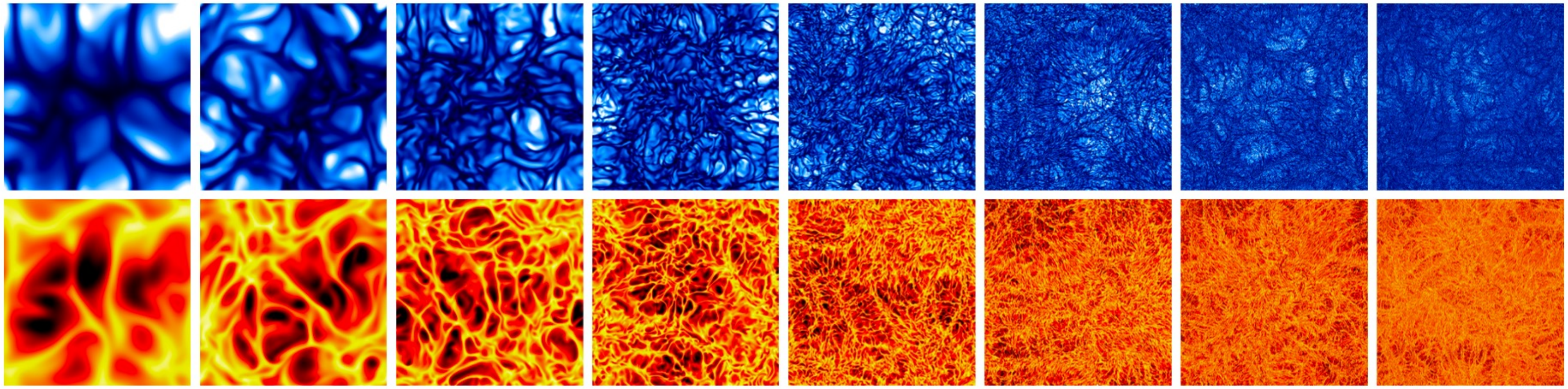
- Time without steering (rough estimates)
  - 20 times trial and error x 30 minutes = 600 compute minutes
  - Scientist need to wait in the queue again
- Time with steering (rough estimates)
  - Load case once x 30 minutes = 30 compute minutes
  - 20 times trial and error x 2 minutes = 40 compute minutes
  - Total: 70 compute minutes

A PEAK INTO THE (NOT SO DISTANT) FUTURE

LARGE SCALE ASCENT RUNS

- Ascent
  - GPU – GPU
  - More realistic rendering
  - ParaView integration
- Workflows that connect the two sites





THANKS

- 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

Silvio Rizzi
srizzi@anl.gov

Victor Mateevitsi
vmateevitsi@anl.gov