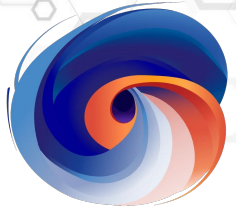


Tools for Scientific Visualization at (Exa)Scale

2025-11-06 - F.Mazen

kitware

francois.mazen@kitware.com

Kitware / Delivering Innovation, Advancing Knowledge

Software and AI R&D Services
Customers in government, industry, and academia
200+ active projects worldwide

Sustained Growth
100% employee-owned
Over $50M revenue

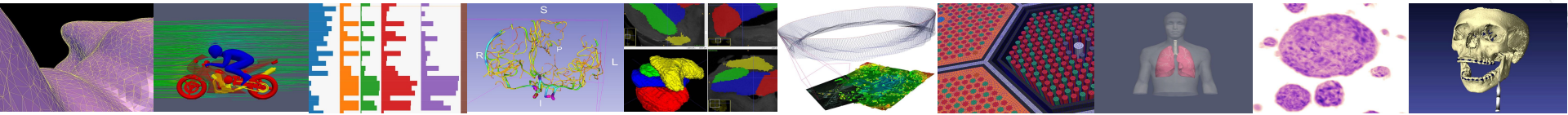
230 Employees Worldwide
Europe, France - Lyon

Deep Expertise
Strong academic reputation
90% staff hold a graduate degree

25+ Years of Experience
Kitware USA, 1998
Kitware Europe, 2010

Founded on Open Source
Vibrant, enduring platform/communities
Strong commitment to Open Science

Kitware / Open Source Platforms - Open Science Focus



CMake

VTK

ParaView

Pulse Physiology Engine

KWIVER

scikit-build

CMB

xaitk

ITK

3D Slicer

VIAME

Girder

tomviz

ADIOS

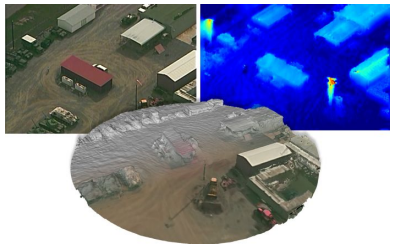
trame

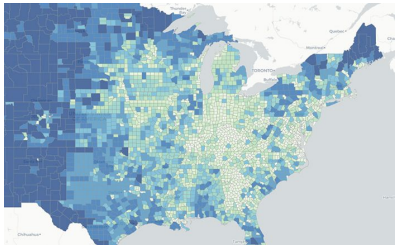
LidarView

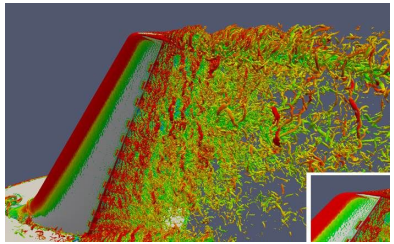
TeleSculptor

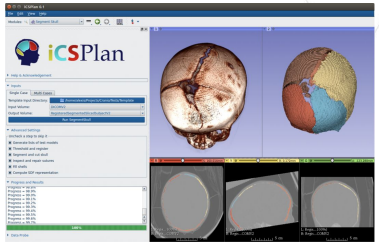
MONAI+

Areas of expertise / Built on open source









Computer Vision

Data and Analytics

Scientific Computing

Medical Computing

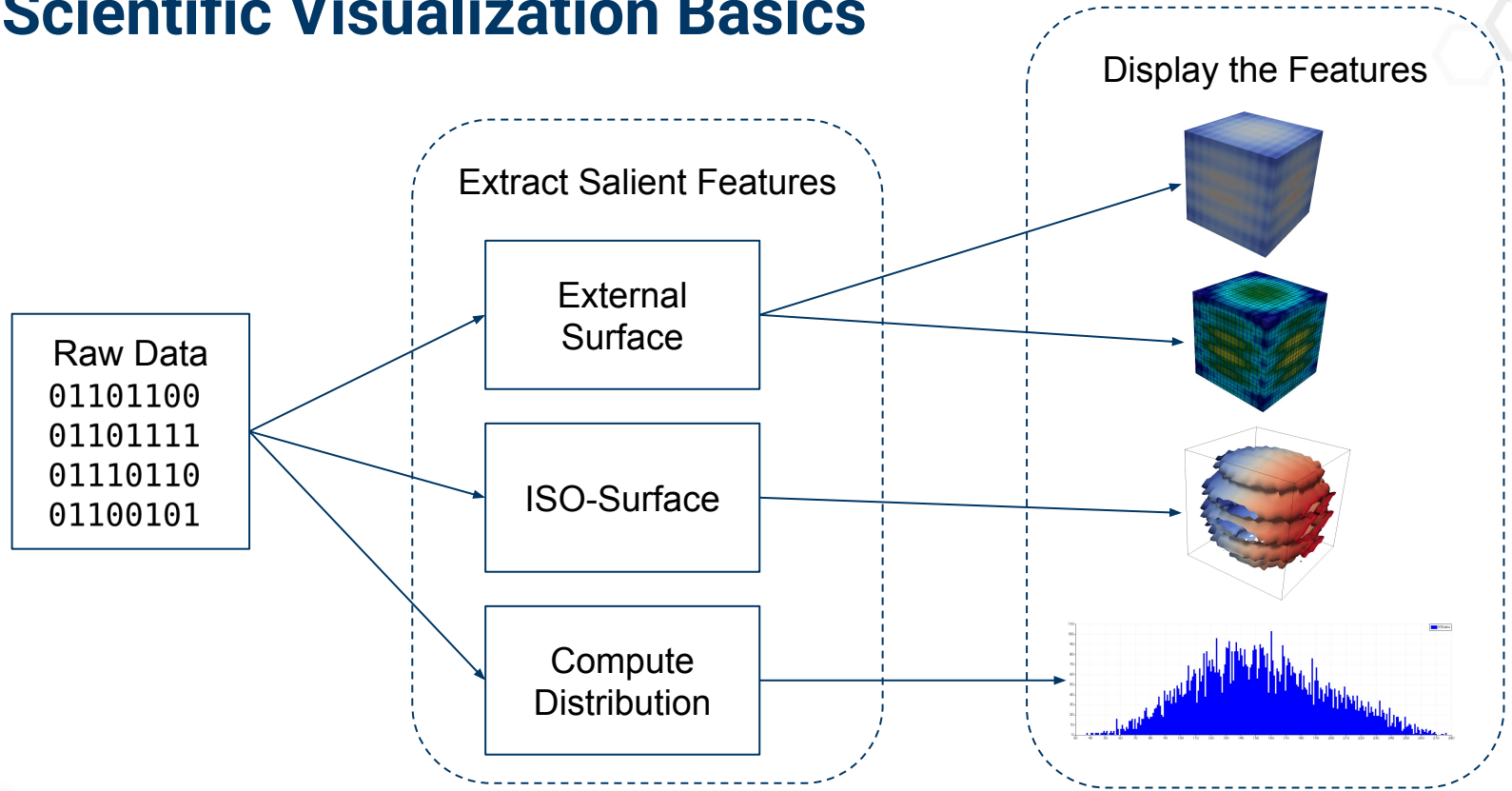
Software Solutions

kitware

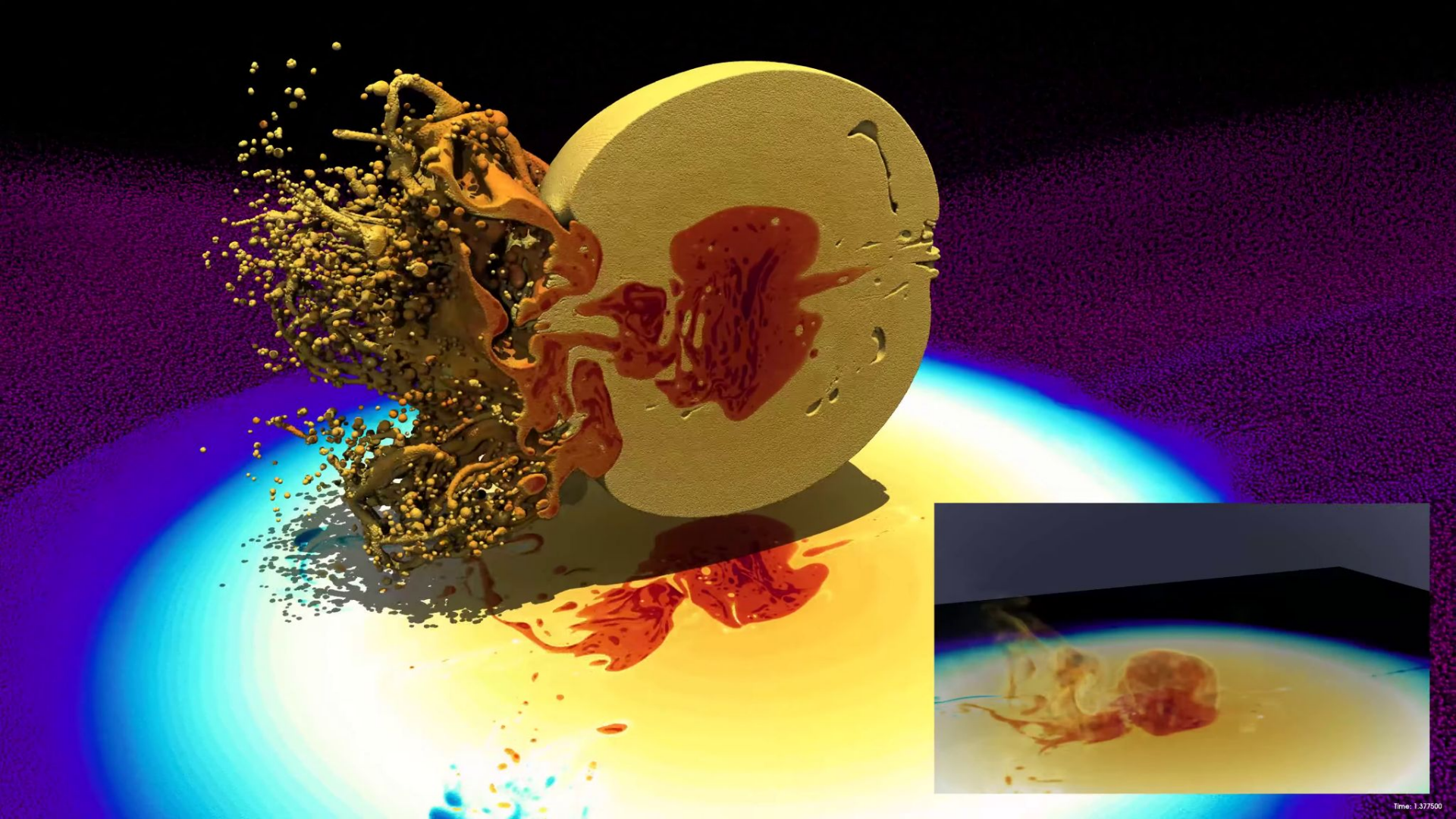
Kitware / Services

kitware

Scientific Visualization Basics

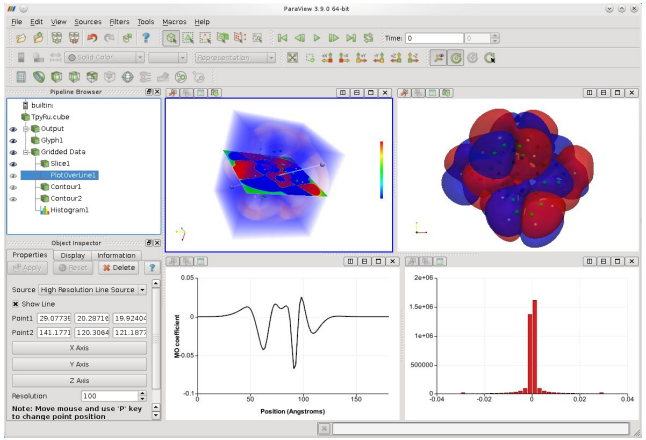


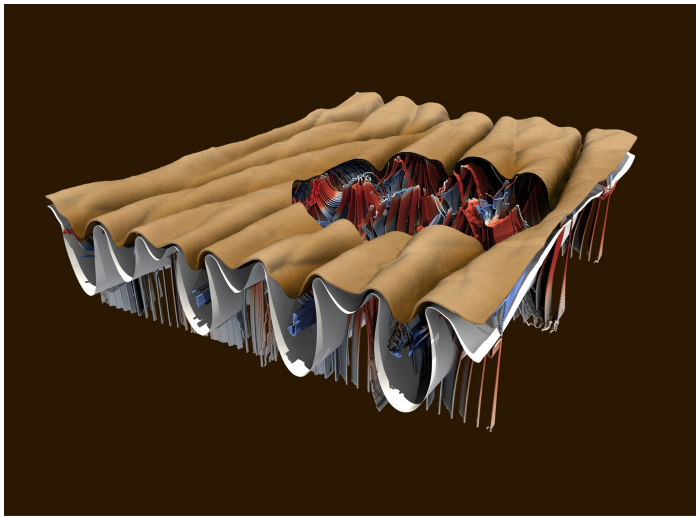
The birth of Jupiter: a first visualization computed on CSCS's 'Alps' machine


Time: 1.377500

ParaView / High-Performance Post-Processing (2002)

- **Open-source, multi-platform, data analysis and visualization application**
- **Analysis of extremely large datasets using distributed memory computing resources**





kitware

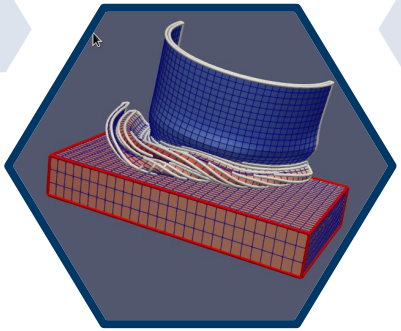
ParaView Community

- Open Source Software (BSD license)
- Run on most of Top500 HPC
- 300000+ download yearly from Kitware servers
- Estimated 1M users worldwide
- 157k commits made by 339 contributors since 2000
- 1.6M lines of code

kitware

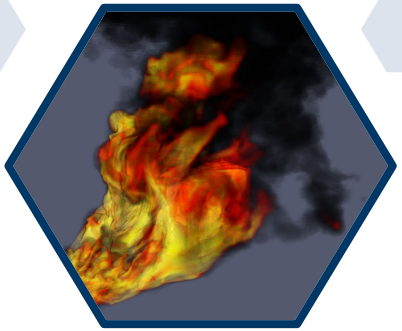
Features / Application Domains

Fluid Dynamic



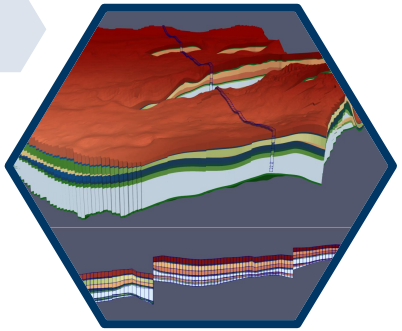
Structural Analysis

Medical



Particles

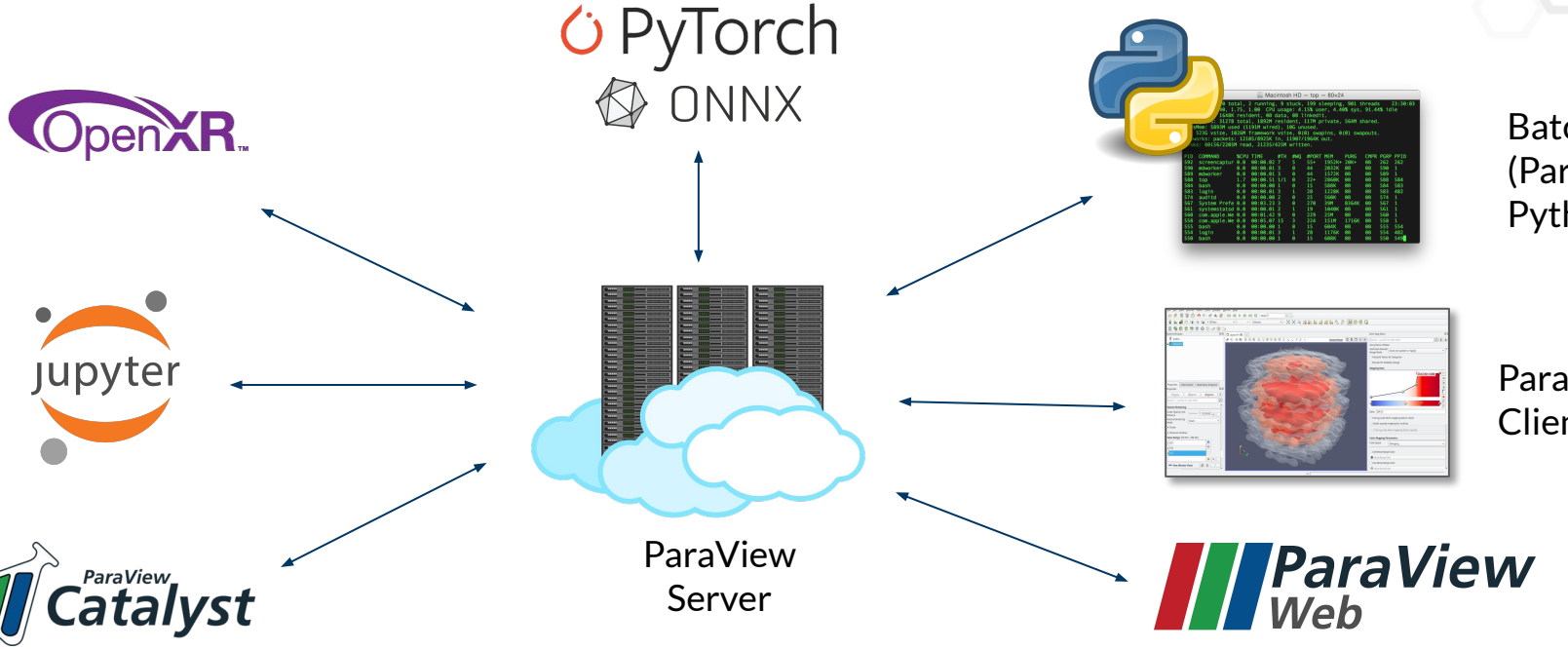
Astrophysic



Geoscience

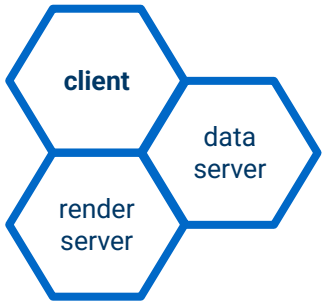
kitware

ParaView Ecosystem

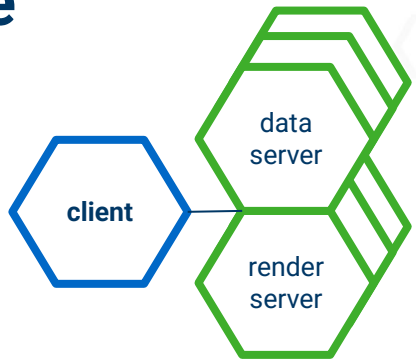


ParaView's Client Server Architecture

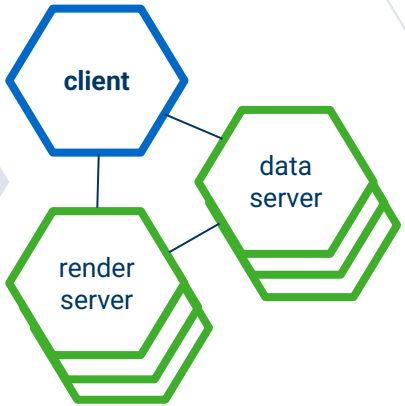
Built-in
paraview



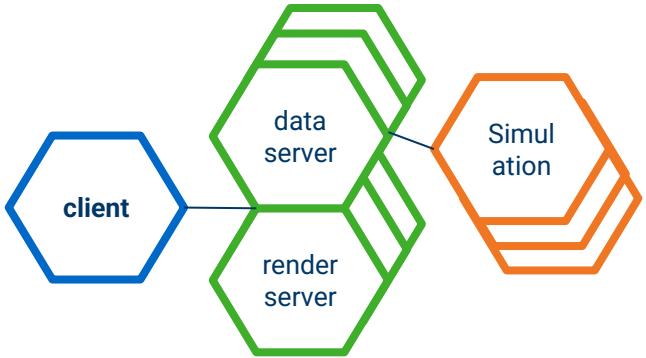
Distributed
pvserver



Graphic Nodes
data/render server



In Situ
catalyst

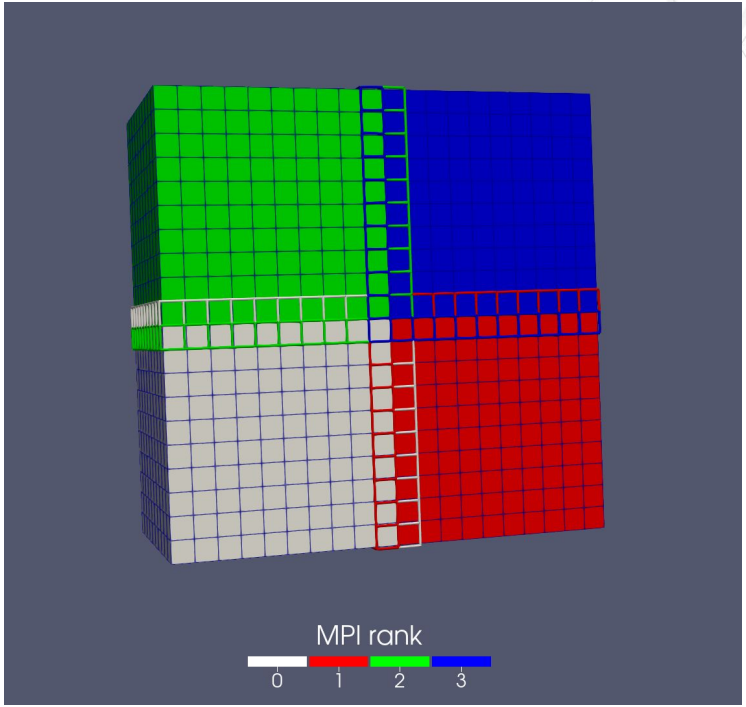


kitware

Data Distribution Analysis

- **Based on MPI standard**
- **Readers distribute data over ranks**
  - load balanced analysis
- **Filters support Ghost Cells**
  - when neighborhood info is needed
- **Filters can redistribute data**
  - ensure load balancing

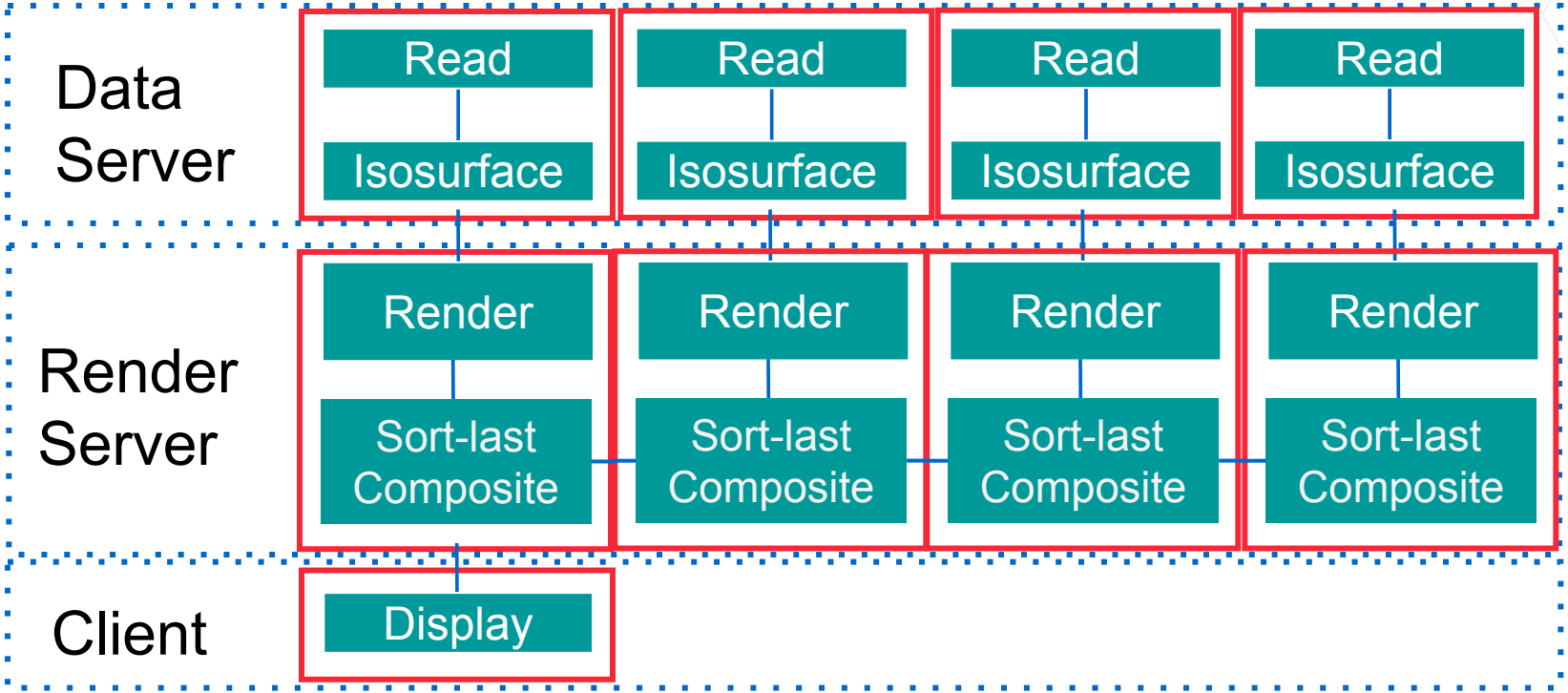
```
$ mpirun -n 4 pvserver
```



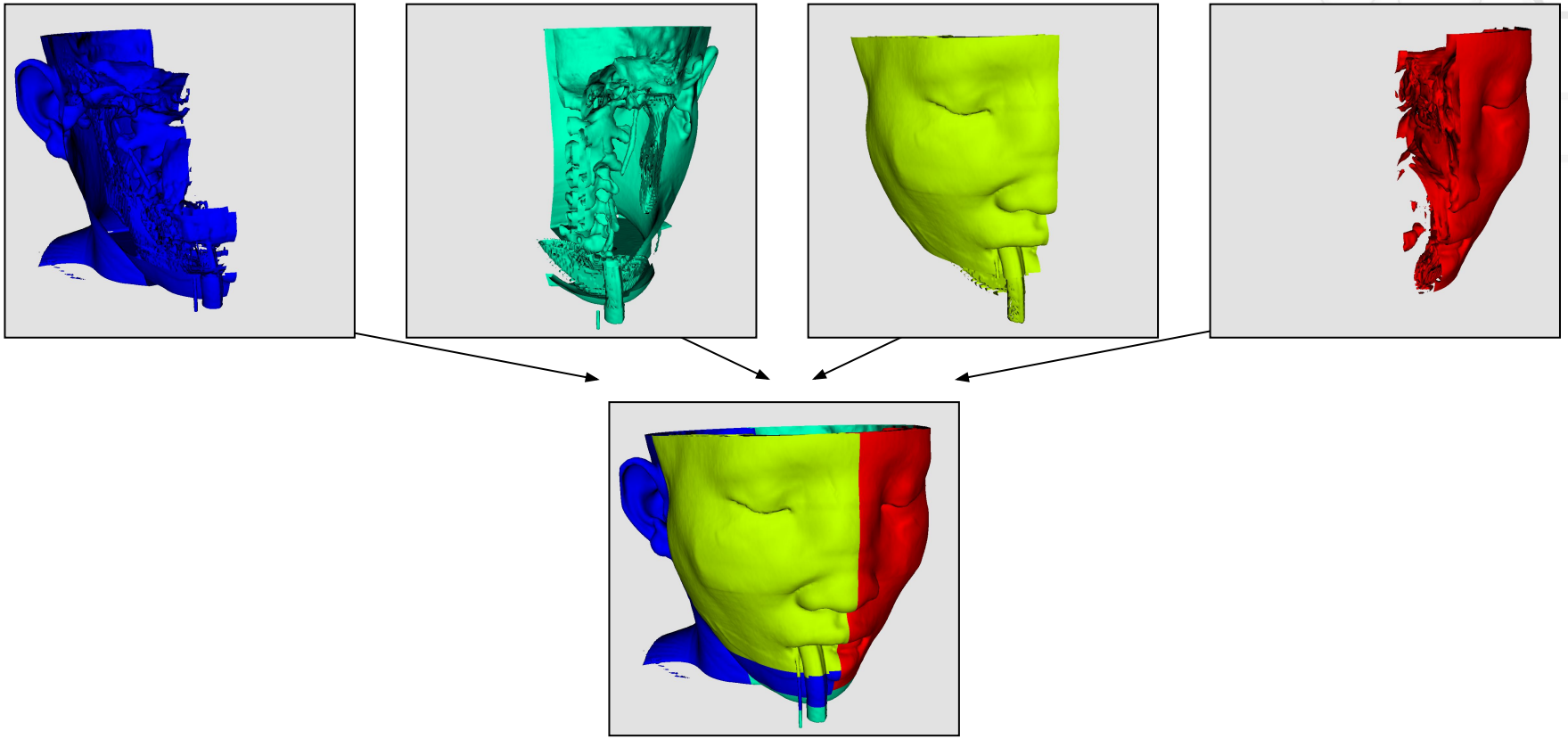
Ghost cells in wireframe

kitware

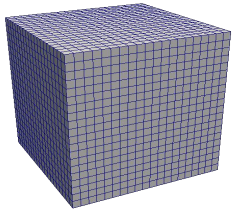
ParaView: Distributed Computation

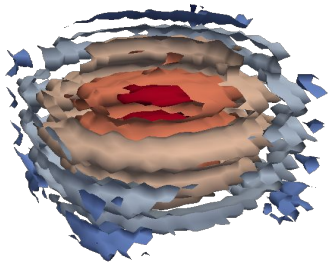


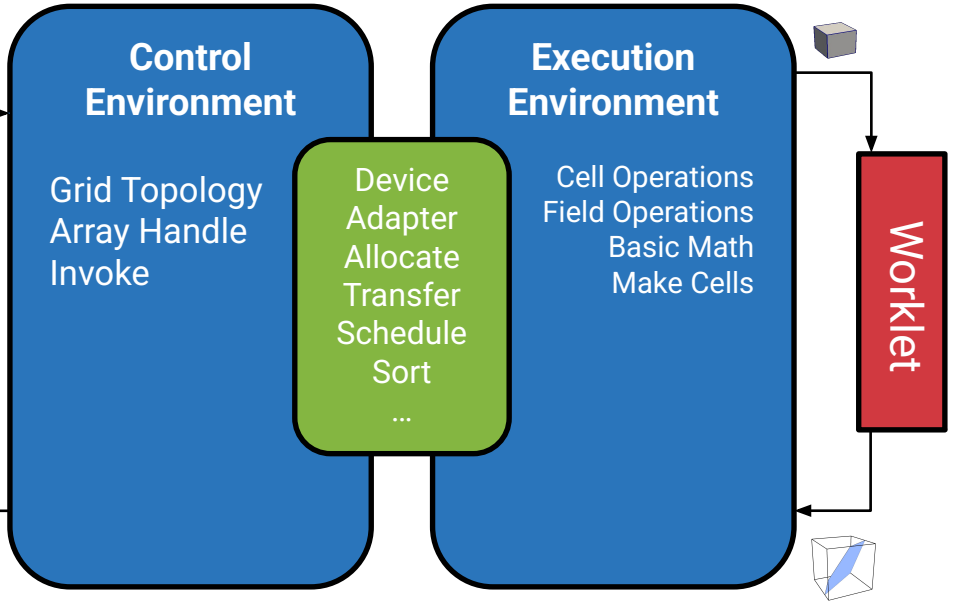
kitware



Viskores / VTK-m







NVIDIA CUDA

kokkos

MPI

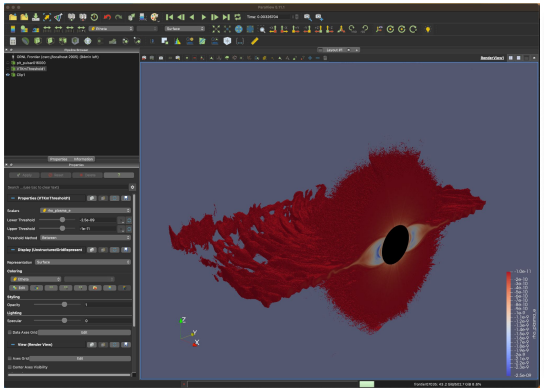
OpenMP

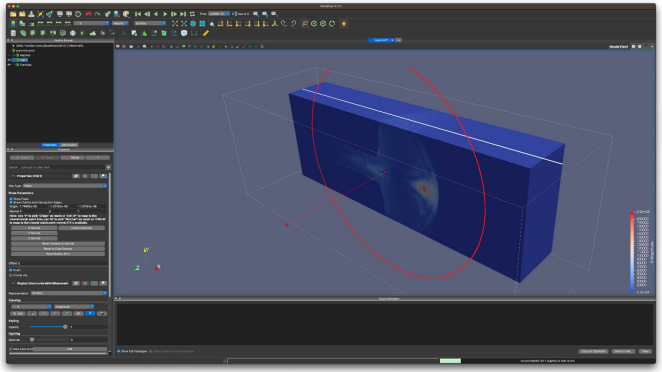
Intel oneTBB

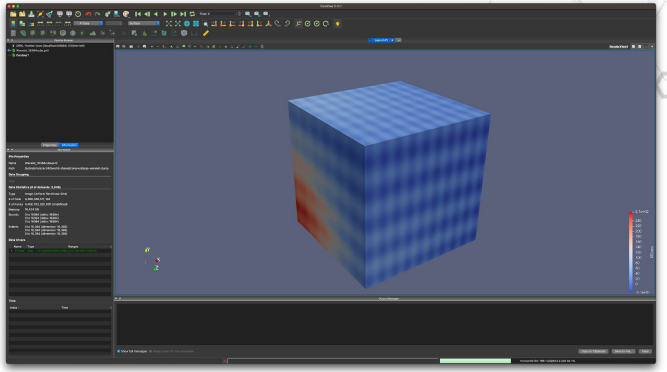
kitware

2016-2024: ECP Achievements (WarpX on Frontier)

Courtesy of Revathi Jambunathan and Axel Huebl at Lawrence Berkeley National Lab







5.75B element / 1.16TB of data.
128 nodes / 512 GPUs

13.6B elements / 3.3 TB per timestep.
256 nodes / 2048 GPUs

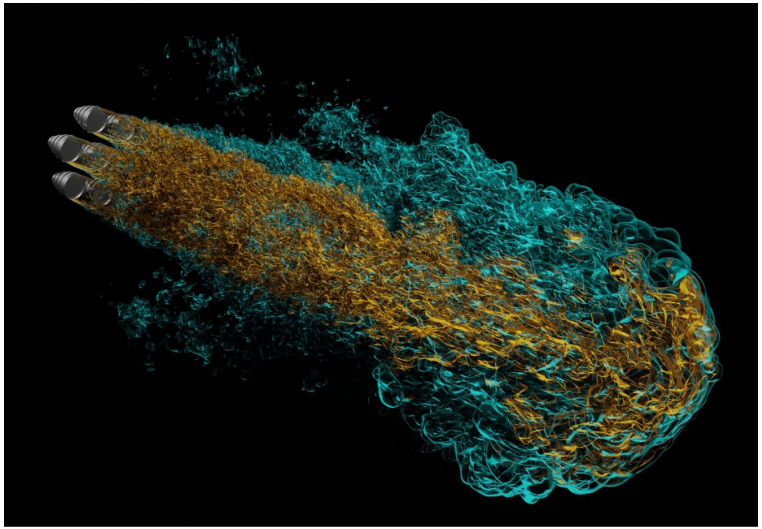
4.4T elements / 16.4TB on disk.
256 nodes / 2048 GPUs

Data Processing: Viskores + Kokkos (HIP/RocM)
Data Rendering: OSMesa (no graphic API on AMD GPUs) :-(

kitware

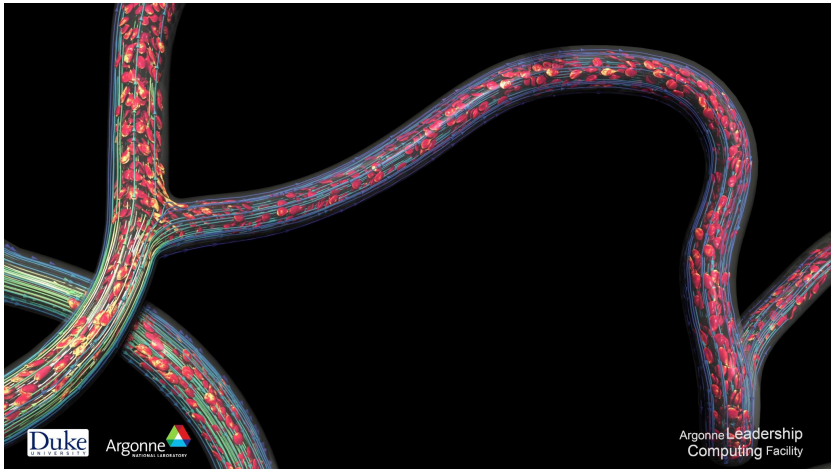
Exascale in 2025-...

SpaceX Starship Super Heavy Booster simulation (Frontier)
200 trillion grid points / 1 quadrillion degrees of freedom



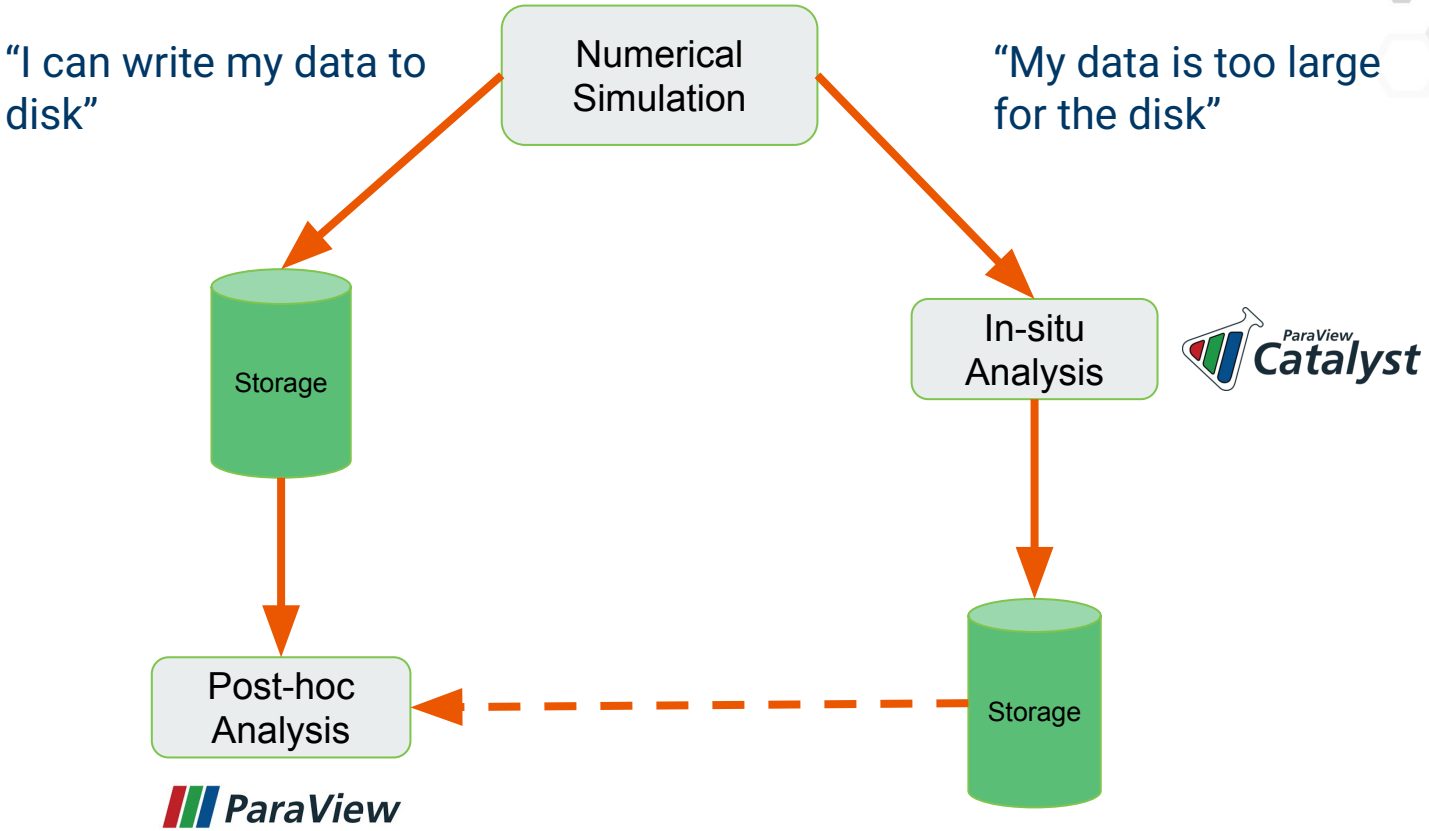
https://www.hpcwire.com/off-the-wire/fluid-flow-simulation-on-frontier-earns-gordon-bell-finalist-selection/

“Preparing to Battle Cancer at the Exascale” on Aurora



https://sc24.supercomputing.org/proceedings/art_of_hpc/art_of_hpc_pages/art129.html

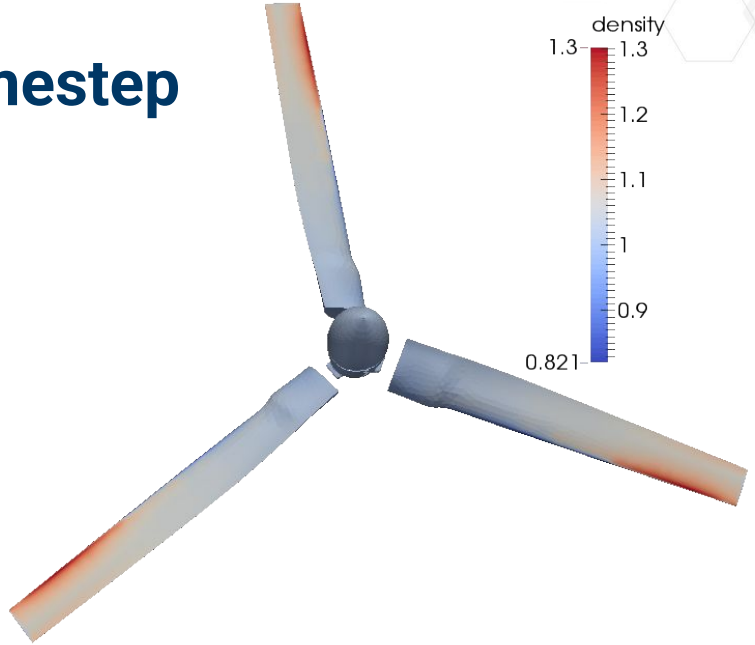
kitware


“I can write my data to disk”
Numerical Simulation
“My data is too large for the disk”
Storage
In-situ Analysis
ParaView Catalyst
Post-hoc Analysis
Storage
ParaView

kitware

In-situ Example in numbers

- **Rotorcraft simulation, per timestep**
  - Full data set – 448 MB
  - Surface of blades – 2.8 MB
  - Image – 71 KB

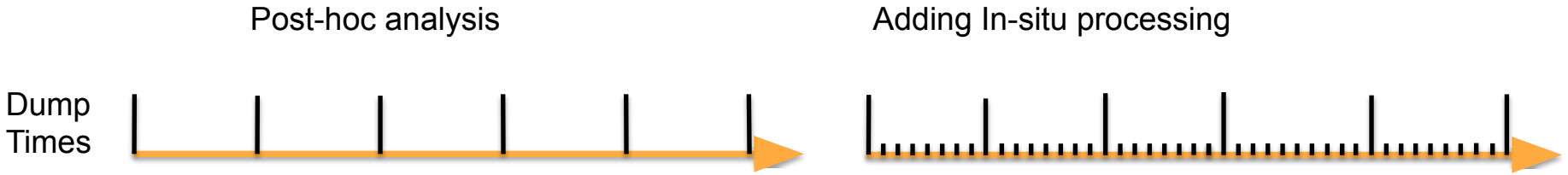


HPCMP CREATE-AV Helios (Army AFDD/AMRDEC) simulation

kitware

Access to More Data

- Increase frequency of data saving
- Reduce size of written data



kitware

Catalyst 2 Application Programmable Interface (API)

https://catalyst-in-situ.readthedocs.io

Simple, stable C API, independent of ParaView

```
enum catalyst_status catalyst_initialize(const conduit_node* params);
enum catalyst_status catalyst_finalize(const conduit_node* params);
enum catalyst_status catalyst_execute(const conduit_node* params);
enum catalyst_status catalyst_results(conduit_node* params);
enum catalyst_status catalyst_about(conduit_node* params);
```

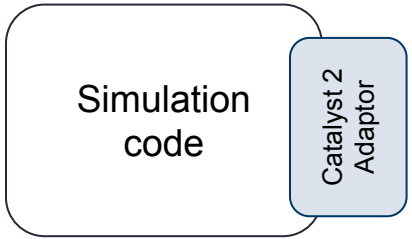
mandatory

optional

Python and Fortran Wrapping available!

Catalyst 2 with ParaView Workflow (1/2)

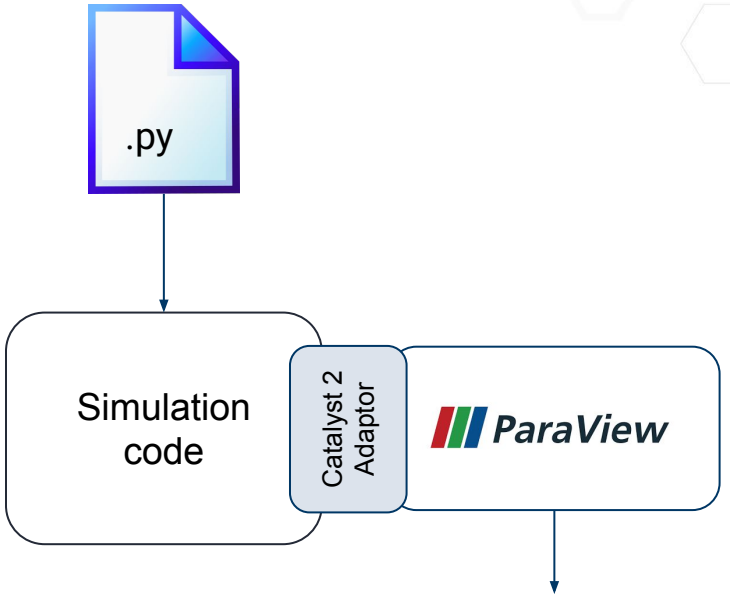
- **Instrument your code with Catalyst 2 once**



- **Prepare a python pipeline with ParaView**

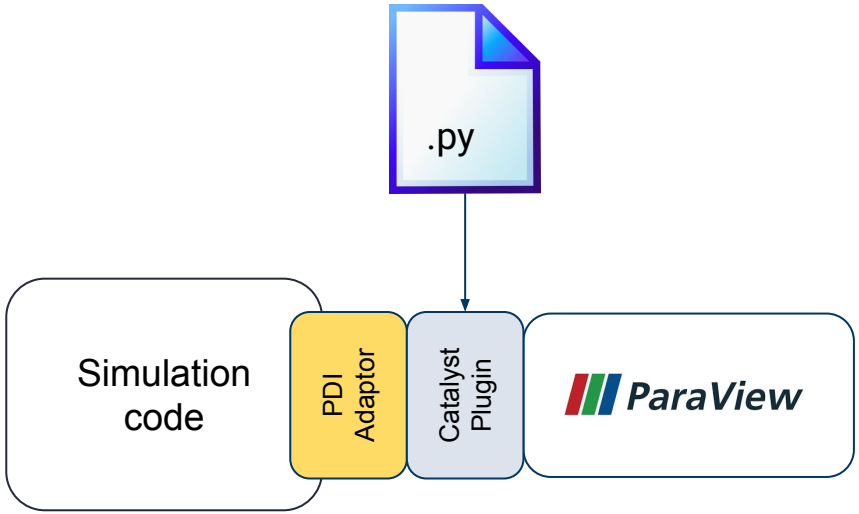
Catalyst 2 Workflow (2/2)

- **Run your simulation code**
  - Specify the python pipeline in the initialize method
  - Specify the Catalyst 2 implementation (i.e. Catalyst-ParaView)



kitware

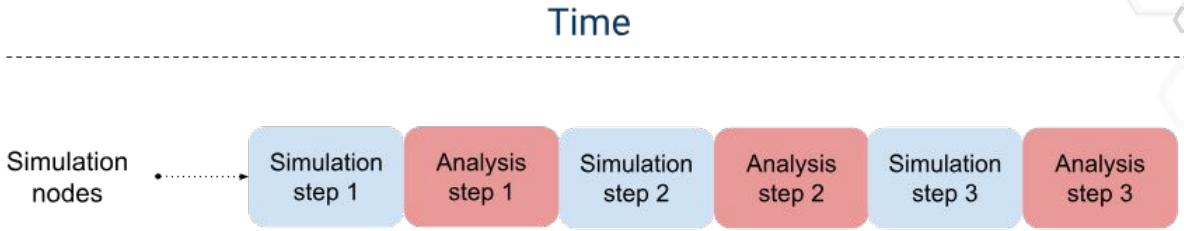
PDI Catalyst Plugin



https://github.com/pdidev/pdi/pull/496

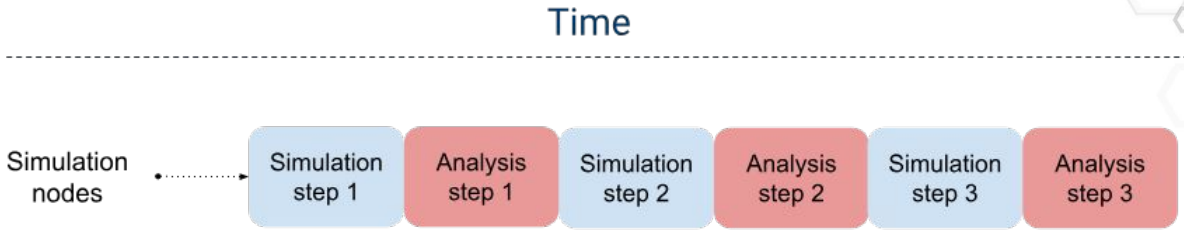
kitware

In situ :

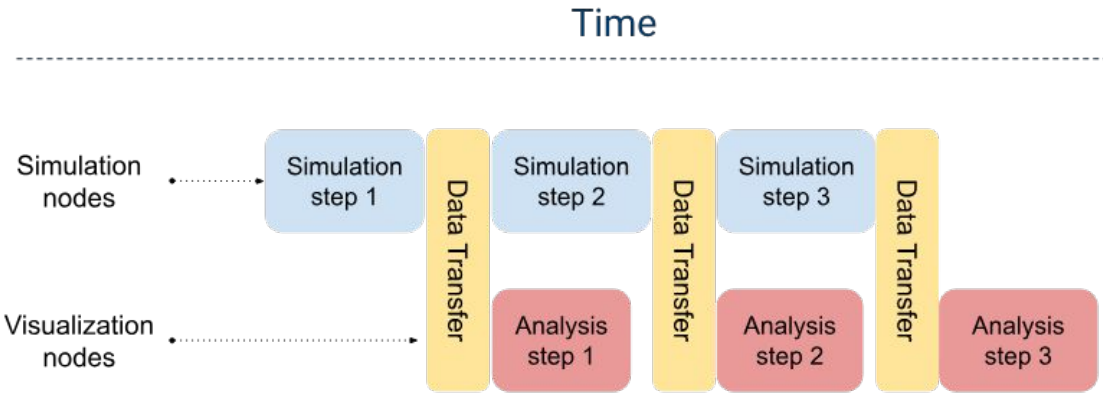


CINECA: Supersonic turbulent boundary layer over a micro-ramp

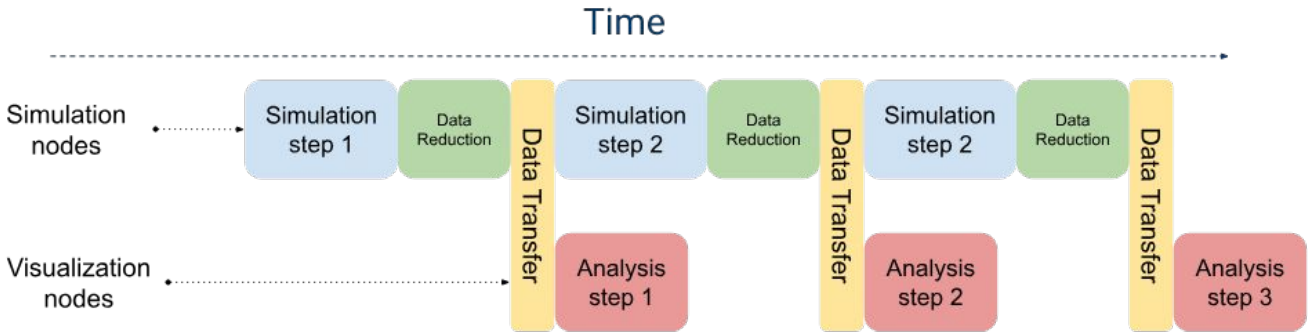
In situ :



In transit :



in situ / in transit Hybrid Analysis

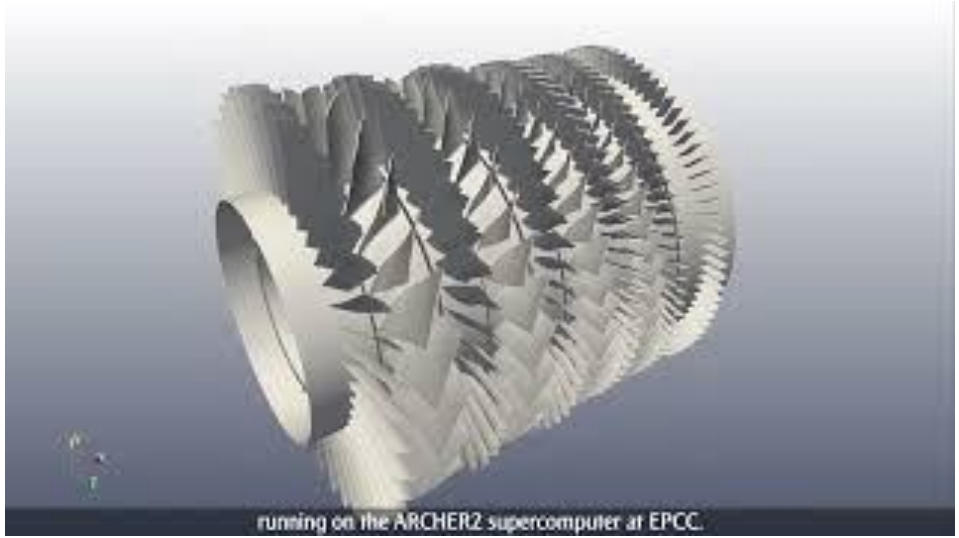


https://www.kitware.com/in-situ-in-transit-hybrid-analysis-using-catalyst-adios2-and-paraview/

kitware

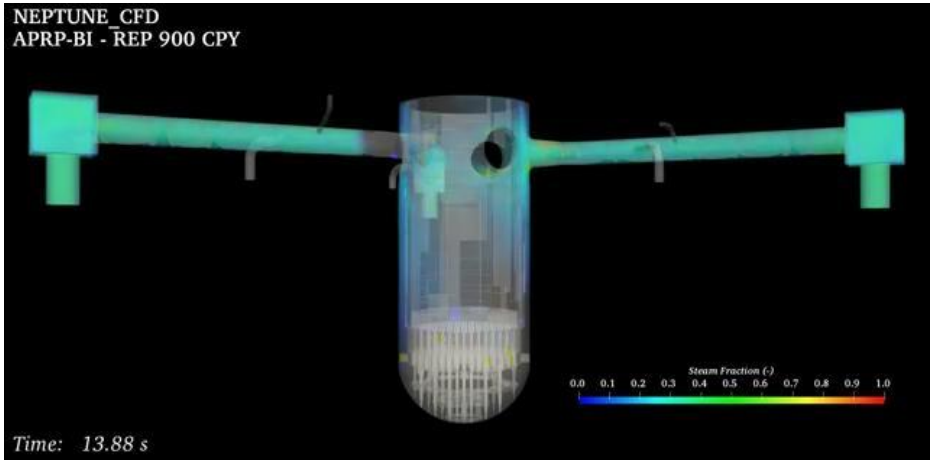
Industrial Use Cases of Catalyst

Rolls-Royce



SC22 SciViz Contest Winner

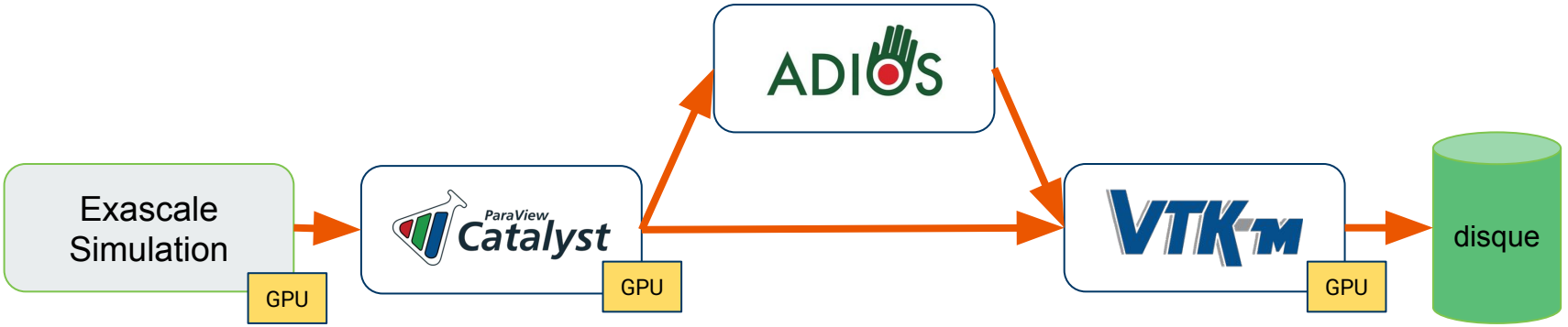
EDF



Courtesy of Nicolas MERIGOUX & Yvan FOURNIER, EDF R&D

kitware

Catalyst GPU-resident workflows

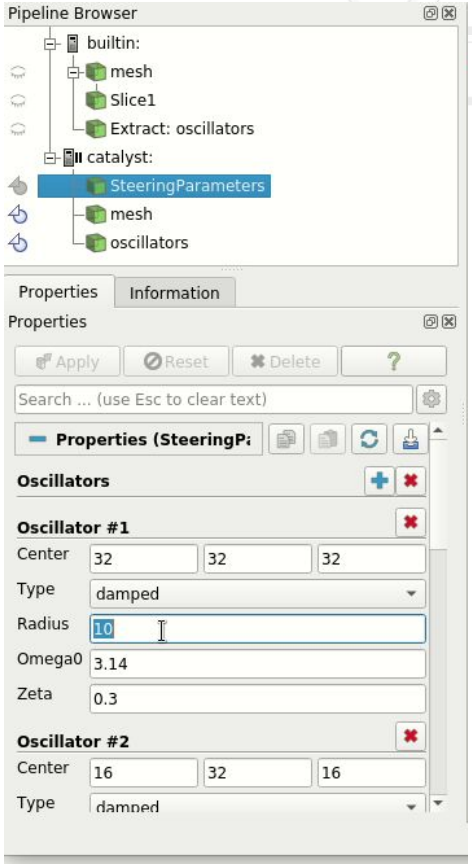


https://www.kitware.com/catalyst2-gpu-resident-workflows/

Catalyst 2 Steering

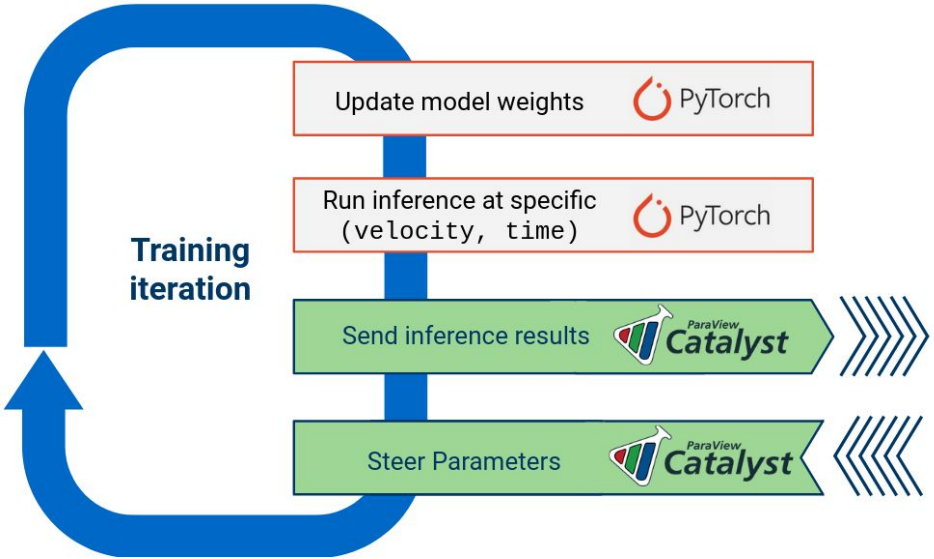
- **Drive the simulation from ParaView via Catalyst API and Live Visualization**

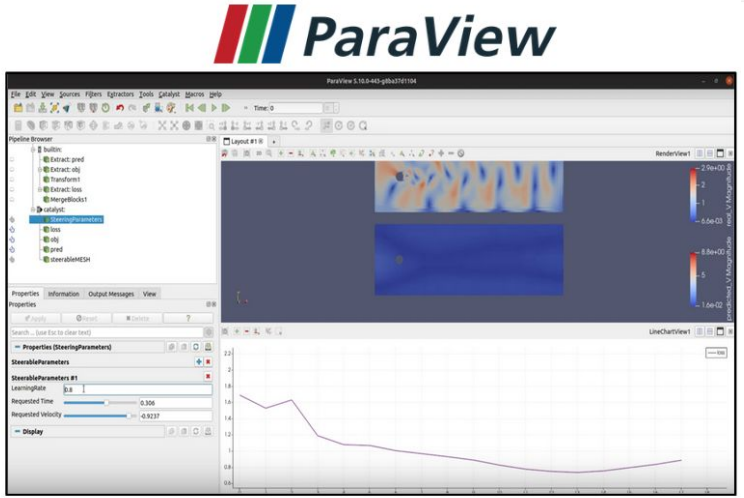
- **Use cases:**
  - Stop simulation nicely :-)
  - Change simulation parameters live
  - Monitor convergence criteria
  - Post-process transient output and send it back to the simulation
  - AI Reinforcement learning



kitware

Deep-learning Model Training Monitoring with Catalyst



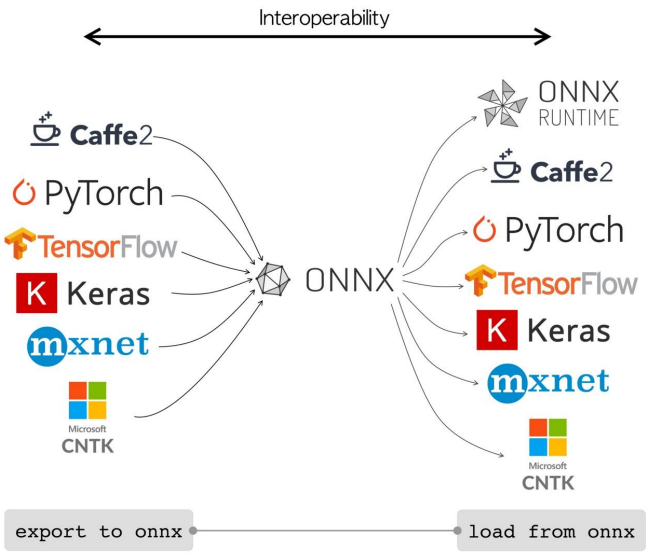


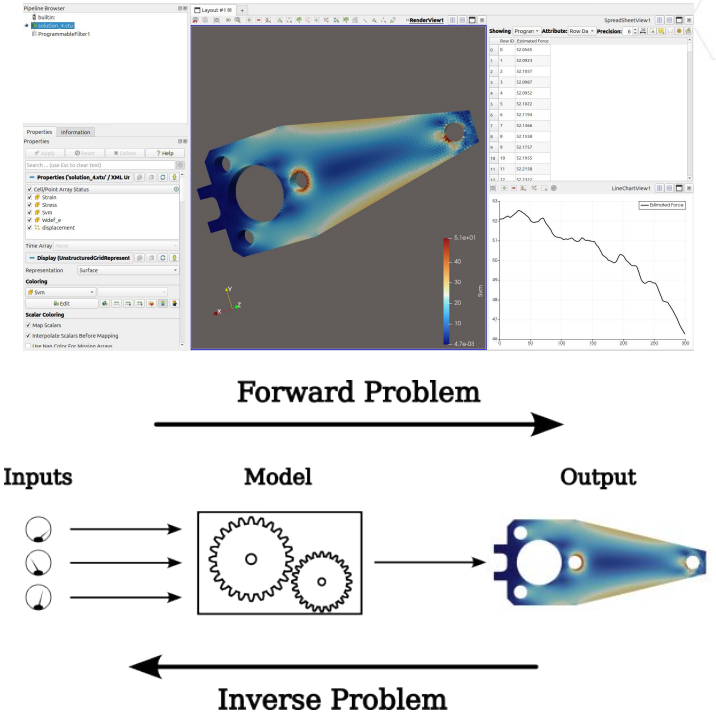
Inference result and loss graph shown in real time

https://www.kitware.com/deep-learning-surrogate-models-in-paraview-viewing-inference-results-and-monitoring-the-training-process-in-real-time-with-catalyst/

kitware

ONNX Models in VTK and ParaView





https://www.kitware.com/enhance-your-paraview-and-vtk-pipelines-with-artificial-neural-networks/

+

= VTKHDF

A new file format to tackle visualization on HPC challenges.

https://docs.vtk.org/en/latest/design_documents/VTKFileFormats.html#vtkhdf-file-format

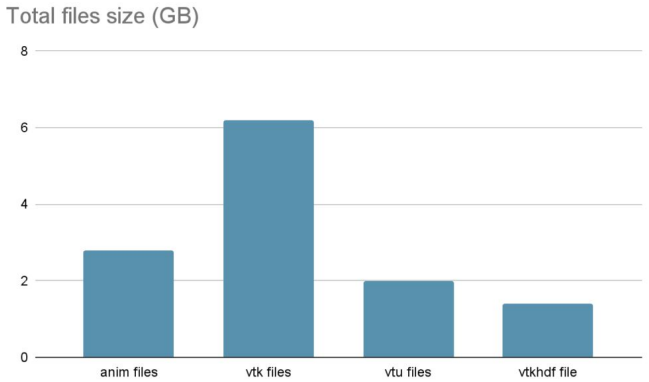
kitware

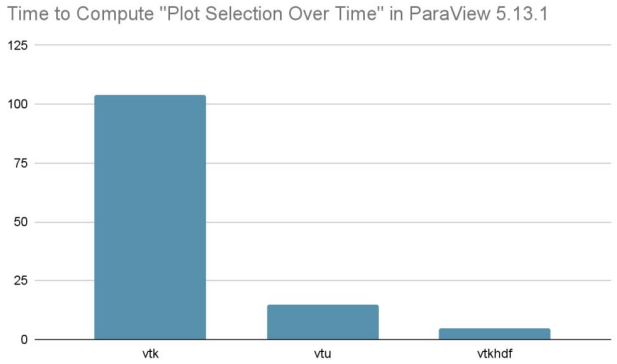
Leverage all HDF5 benefits:

- Distributed read/write
- Offsets, Chunks, Multi Volumes
- Compression
- Performance

VTKHDF:

- Do not depend on VTK when writing file (h5py is all you need)
- “Static Mesh”
- Read Performance





kitware

https://www.kitware.com/post-process-your-openradioss-results-with-paraview/

VTKHDF : doc & status

The VTKHDF tutorial

> A gentle introduction to the VTKHDF file format

VTKHDF is a new generation file format to store VTK datasets, that was introduced in 2021, and still being improved as of 2025. This format succeeds XML-based VTK file formats, using a unified API and a single extension for all the different data types it can contain. The general structure of a VTKHDF file is the same for all data types.

VTKHDF Basics

VTKHDF is a specification based on the HDF5 container format. This tutorial uses the h5py (3.13.0) Python module to build data files, which you can then read using ParaView (>=6.0). Each step should yield a valid file, progressively enabling more features of the format.

Using the HDF5 container format, the VTKHDF standard defines a hierarchy of groups and datasets, the root of all of them being the `VTKHDF` group. All other root HDF5 groups will be ignored by the reader. The `VTKHDF` group needs to have 2 attributes: `Version`, a vector of 2 elements defining the version of the reader required to read the current file, and `Type`, a fixed-length string indicating which type of dataset should be read.

Using h5py, you can create a new HDF5 file, create a VTKHDF group and the required attributes for an unstructured grid:

```
import h5py as h5
file = h5.File("step1.vtkhdf", "w")
root = file.create_group("VTKHDF")
root.attrs["Version"] = (2, 3)
root.attrs["Type"] = "UnstructuredGrid"
```

Creating a basic Unstructured Grid

Unstructured Meshes are one of the base data types that VTKHDF can encode. Unstructured Meshes are defined using points, meshed together to form cells. In this example, we will write a file creating 2 cells, one tetrahedron and one triangle.

VTKHDF: Implementation Status

This document specifies the features currently supported by the VTK implementation of the VTKHDF file format, `vtkHDFReader` and `vtkHDFWriter`. The following list only showcases complete and planned developments.

File Format Specifications Status

Data Types

VTKHDF File Format should support at some point every data type of VTK (e.g. `vtkTable`, `vtkMolecule`,...).

The following list shows the VTK data types currently supported, at least partially.

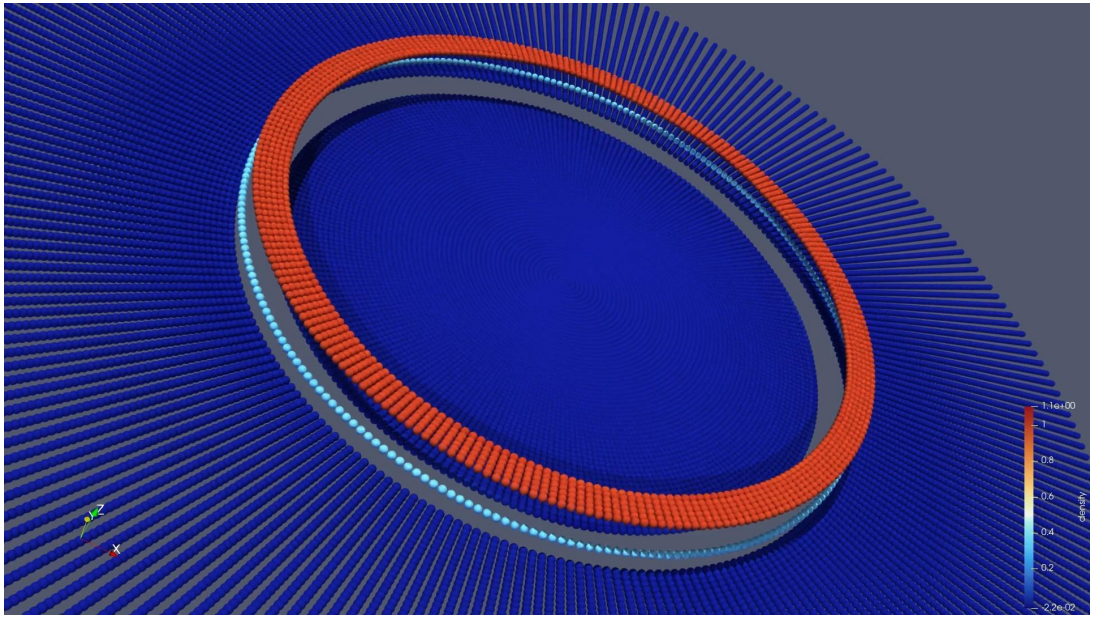
VTK Data Type	Status
vtkCellGrid	not implemented
vtkHyperTreeGrid	supported
vtkImageData	supported
vtkMultiBlockDataSet	supported
vtkPartitionedDataSet	supported
vtkPartitionedDataSetCollection	supported
vtkOverlappingAMR	supported
vtkPolyData	supported
vtkRectilinearGrid	not implemented
vtkStructuredGrid	not implemented
vtkUnstructuredGrid	partially supported

kitware

https://gitlab.kitware.com/keu-public/vtkhdf/vtkhdf-scripts/-/tree/main/tutorial

https://docs.vtk.org/en/latest/vtk_file_formats/vtkhdf_file_format/vtkhdf_status.html

Conclusion



Gysela-X Diocotron Instability

- Kitware
- ParaView
- Distributed Rendering
- Catalyst
- VTKHDF

RFC: Request For Collaboration!

kitware

Kitware USA
kitware@kitware.com
+1 (518) 371-3971

Kitware Europe
kitware@kitware.eu
+33 437-450-415