The Scalable Data Management, Analysis, and Visualization Institute http://sdav-scidac.org

ParaView Catalyst

Scalable *In Situ* Analysis and Visualization

Utkarsh Ayachit[1], Berk Geveci[1], Ken Moreland[2], Jim Ahrens[3]

[1]Kitware, Inc., [2]Sandia National Laboratories, [3]Los Alamos National Laboratory,

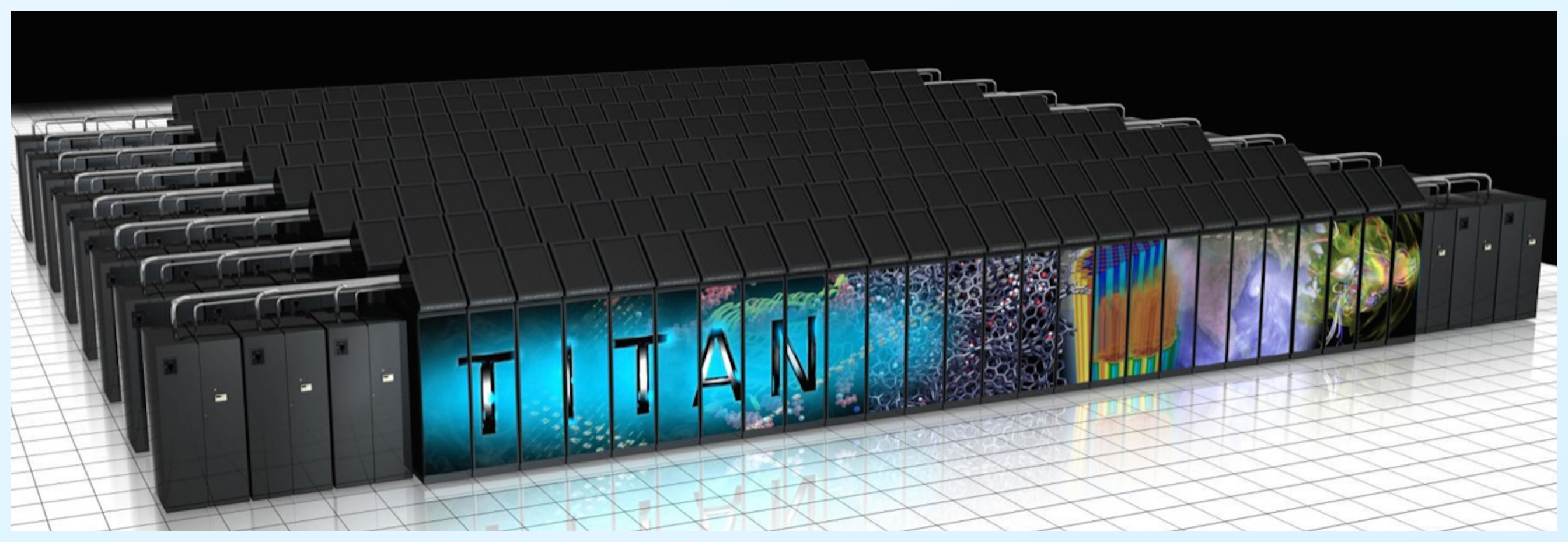
http://catalyst.paraview.org/

Overview

To avoid data bottlenecks in big data simulations, an *in situ* toolset, ParaView Catalyst, integrates core data processing and visualization with simulations to enable scalable data analysis. I/O is one of the most pressing challenges with large-scale simulations, and it is already common for simulations to discard most of what they compute in order to minimize time spent on I/O. Since storing data is no longer viable for many simulation applications, data analysis and visualization must now be performed *in situ* with the simulation to ensure that it is running smoothly and to fully understand the results that the simulation produces.

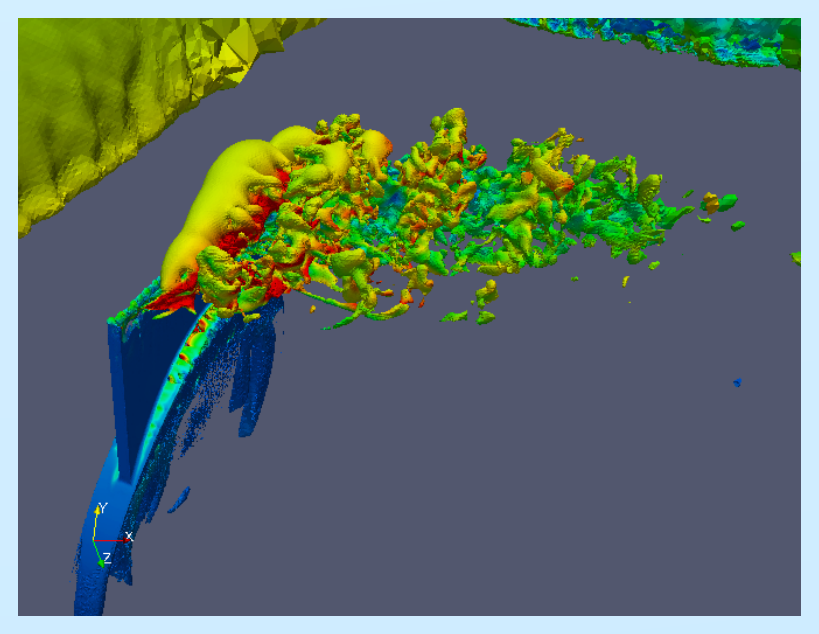
Disparity Between Computational Resources

Typically there is about 2 orders of magnitude difference of computational resources between the leading edge HPC hardware and corresponding visualization hardware. At OLCF, Titan has 299,008 Opteron cores, 18,688 NVIDIA Tesla K20 GPU accelerator cores and 710 TB of system memory. Rhea has 3,136 Intel Xeon cores and 12,544 PB of system memory.

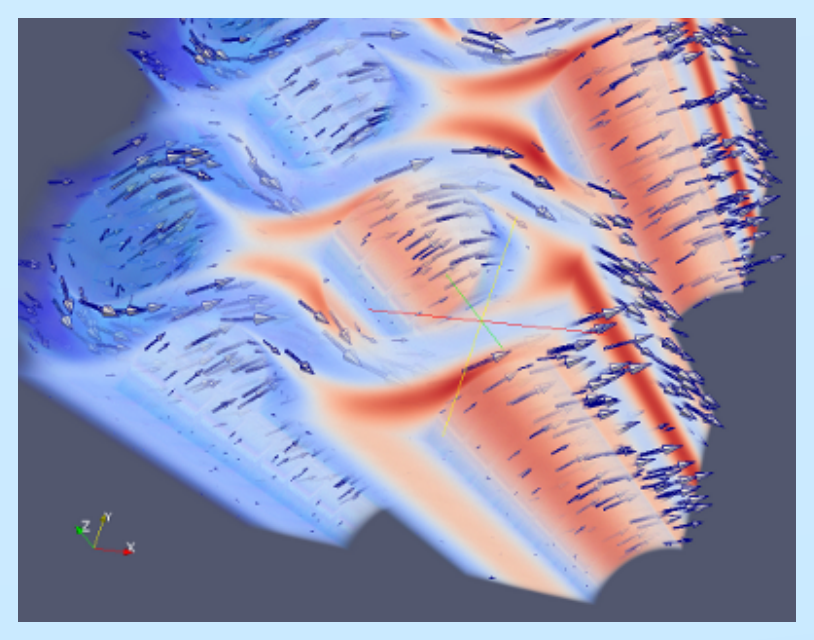


Simulation Codes Instrumented with Catalyst

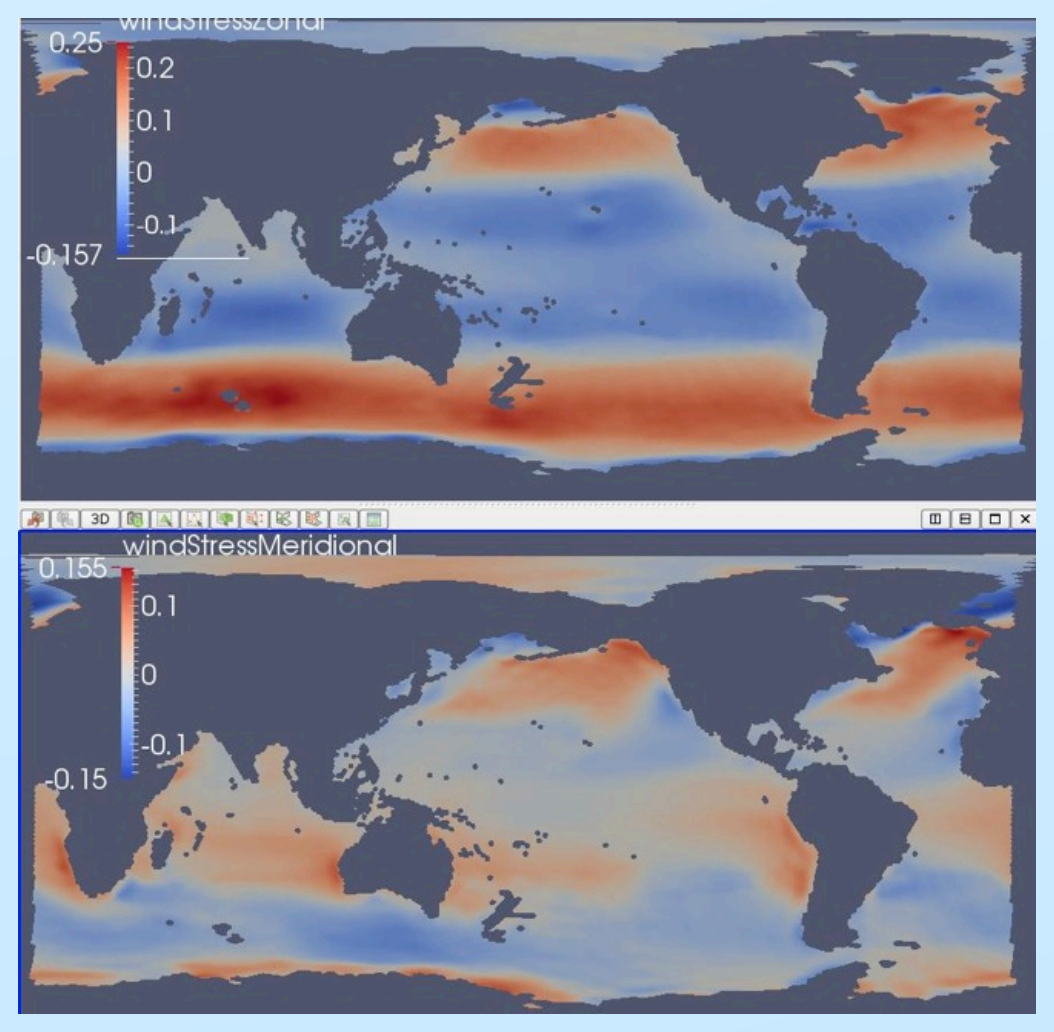
DoE	DoD	Other
Albany	AdH	BEC (William & Mary)
Alegra	GEMS	LESLIE (Georgia Tech)
CTH	Helios	PHASTA (UC Boulder)
Hydra	SM/MURF	Code Saturne (EDF)
Mantevo/miniFE-2.0		UH3D (SciberQuest, Inc.)
MPAS-O		
NPIC		
Sierra		
VPIC		
XRAGE		
CAM5*		



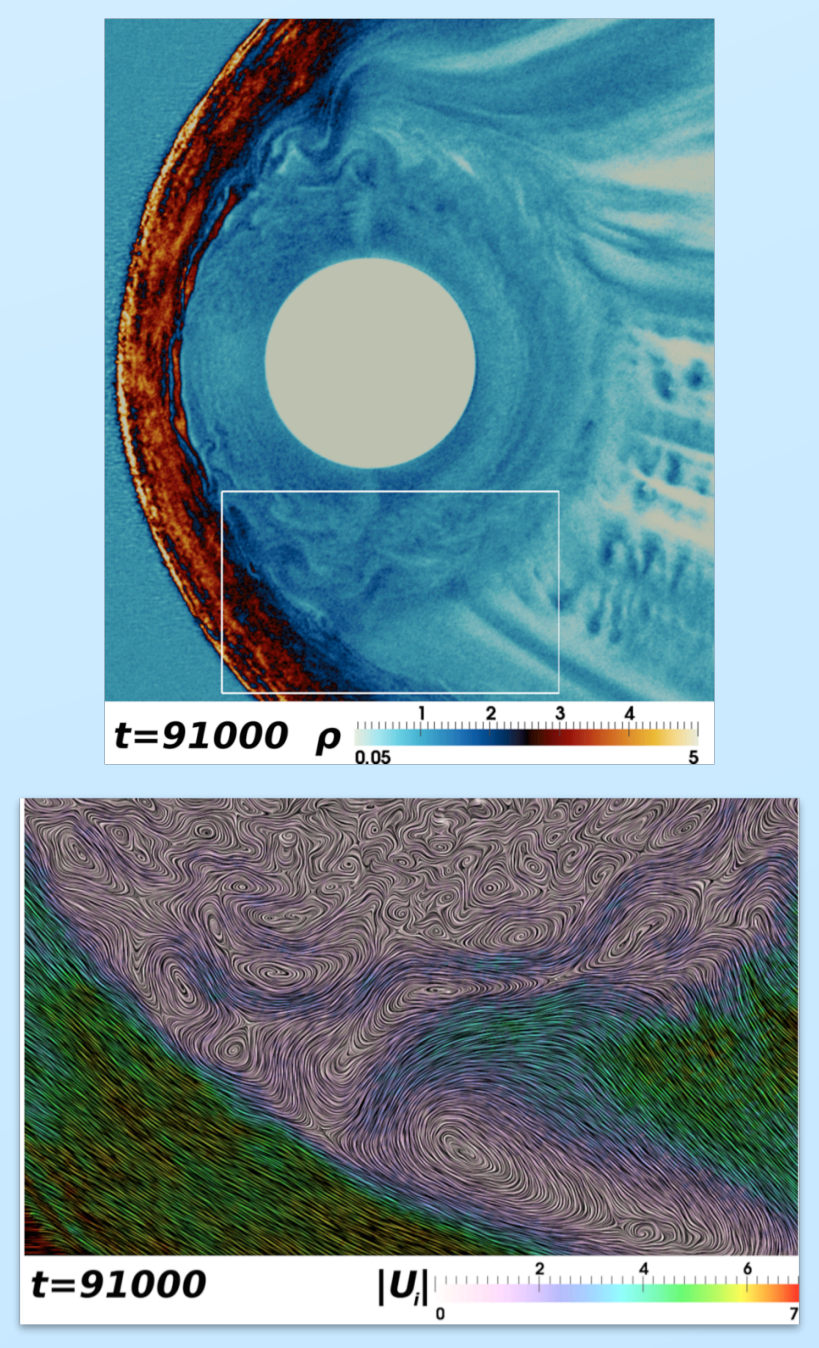
PHASTA



Code Saturne

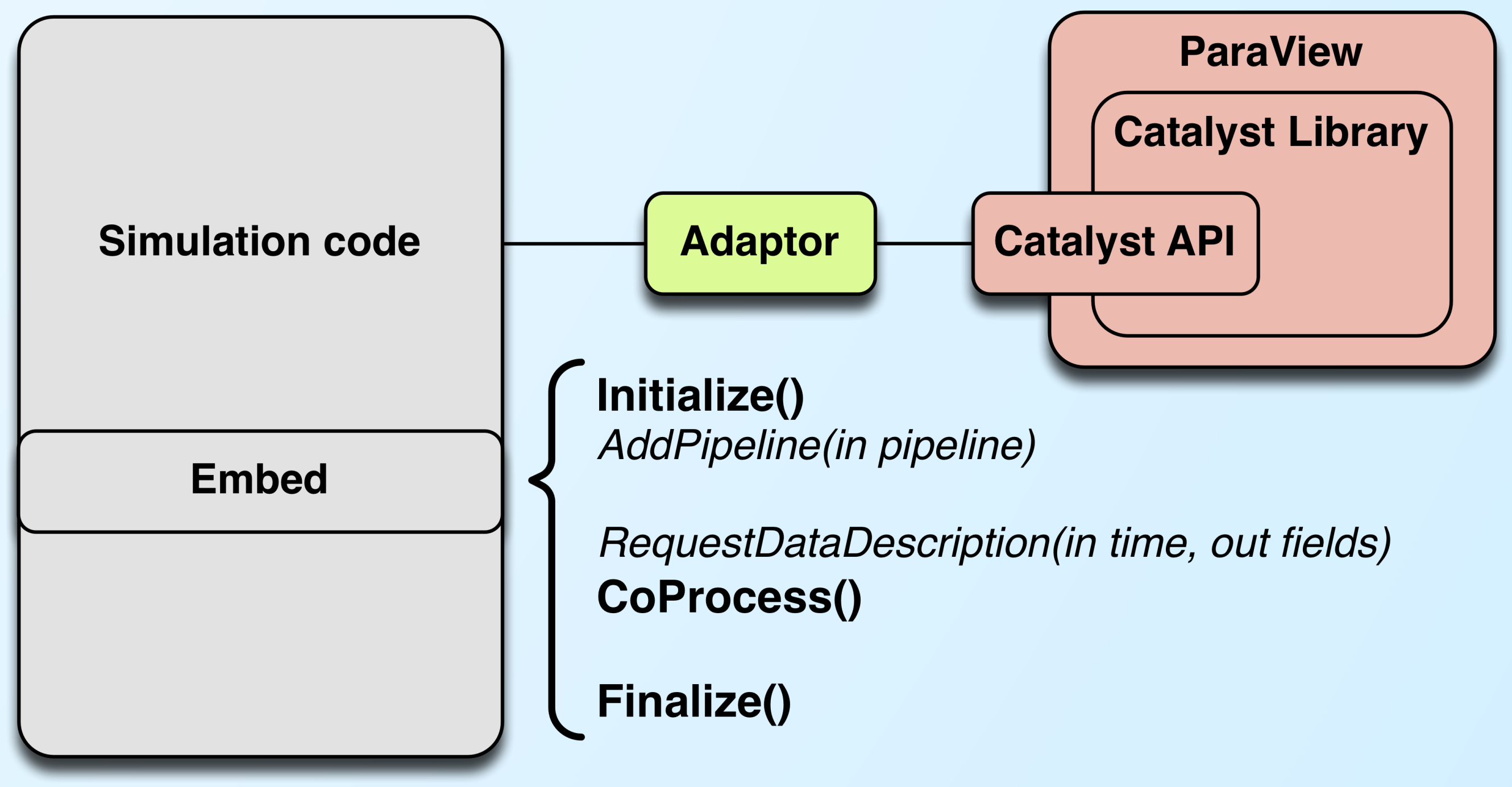


MPAS-O



UH3D

Architecture



Simulation code

Embed

Adaptor

ParaView

Catalyst Library

Catalyst API

Initialize()

AddPipeline(in pipeline)

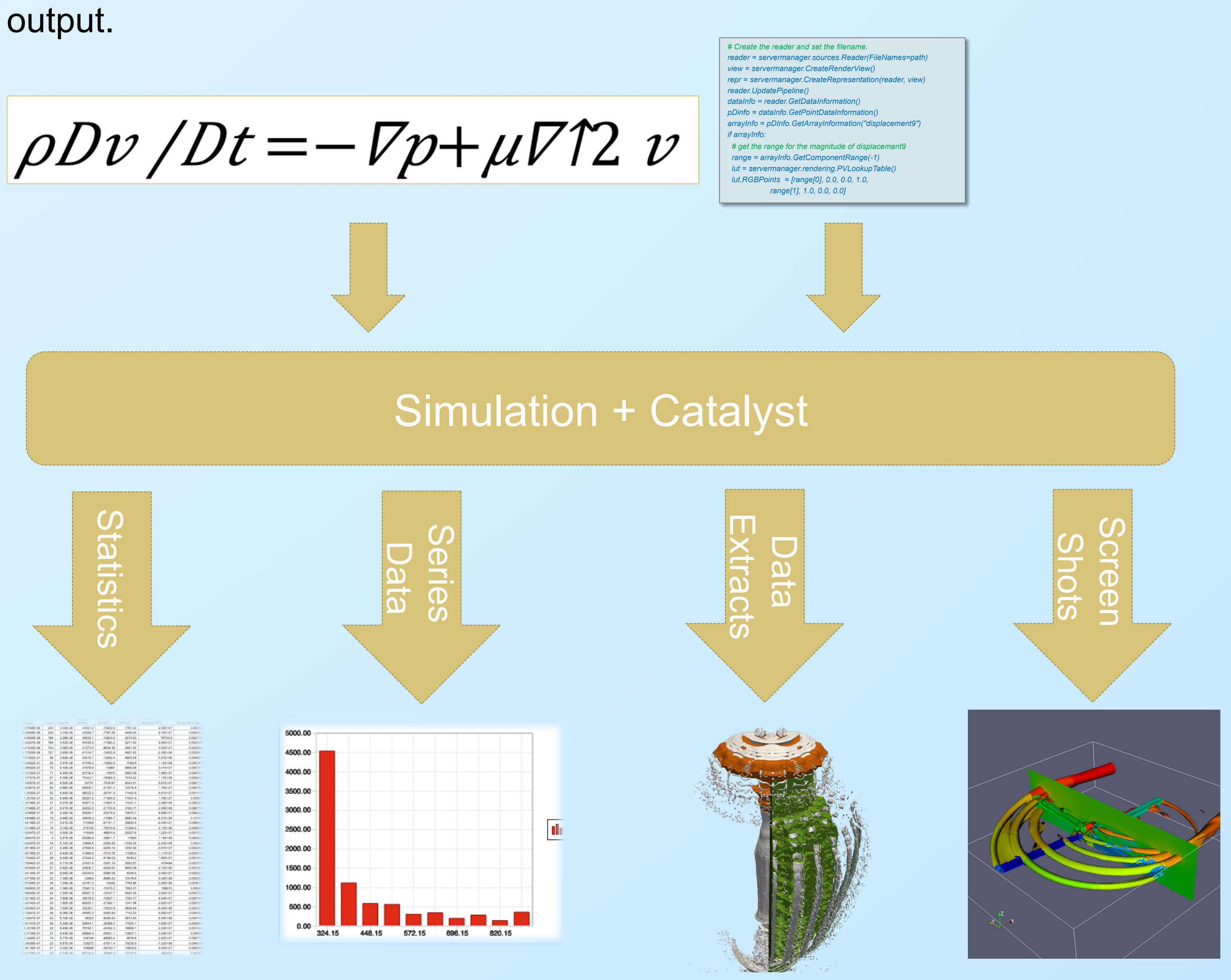
RequestDataDescription(in time, out fields)

CoProcess()

Finalize()

Workflow

The workflow for generating ParaView Catalyst output includes generating Catalyst pipelines to be executed during certain points in the simulation. The simplest way to do this is by generating Catalyst pipeline Python scripts in the ParaView GUI. Besides this, Catalyst C++ or Python pipelines can be hard-coded into the adaptor. In this case, the Catalyst pipelines typically are provided run-time parameters to give a bit more control over the generated output.

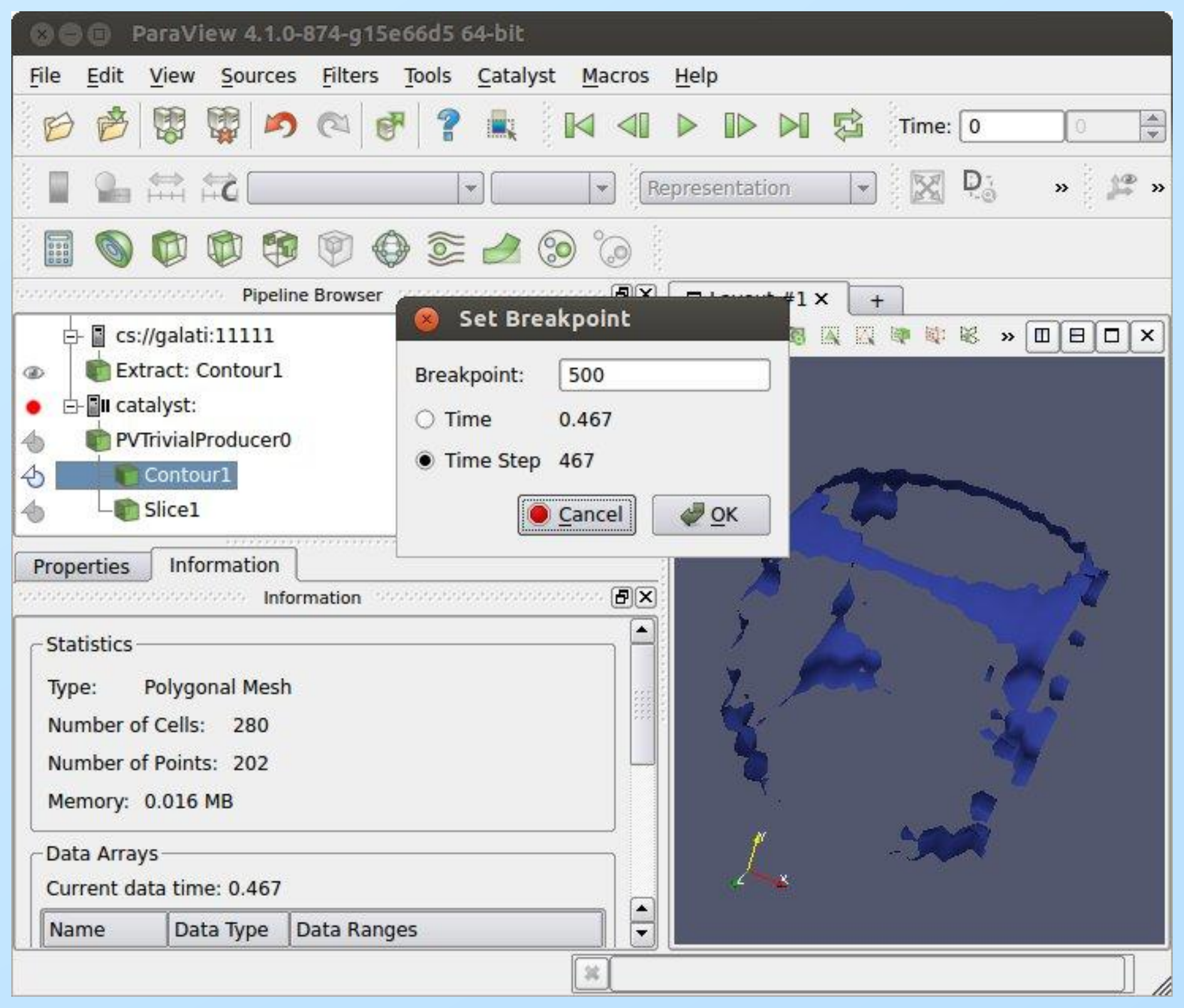


$$\rho Dv/Dt = -\nabla p + \mu \nabla \uparrow 2\ v$$

```
# Create the reader and set the filename.
reader = servermanager.sources.Reader(FileNames=path)
view = servermanager.CreateRenderView()
repr = servermanager.CreateRepresentation(reader, view)
reader.UpdatePipeline()
dataInfo = reader.GetDataInformation()
pDinfo = dataInfo.GetPointDataInformation()
arrayInfo = pDInfo.GetArrayInformation("displacement9")
if arrayInfo:
  # get the range for the magnitude of displacement9
  range = arrayInfo.GetComponentRange(-1)
  lut = servermanager.rendering.PVLookupTable()
  lut.RGBPoints  = [range[0], 0.0, 0.0, 1.0,
                    range[1], 1.0, 0.0, 0.0]
```

Simulation + Catalyst

The ParaView client can connect to a Catalyst enabled simulation run to check on the status as well as change the Catalyst pipelines. For at scale runs the updates occur without stopping the simulation. For debugging runs the simulation can be stopped at specified points to check the simulation state.



ParaView 4.1.0-874-g15e66d5 64-bit

Pipeline Browser:
- cs://galati:11111
- Extract: Contour1
- catalyst:
- PVTrivialProducer0
- Contour1
- Slice1

Set Breakpoint

Breakpoint: 500

○ Time 0.467

● Time Step 467

Information

Statistics

Type: Polygonal Mesh

Number of Cells: 280

Number of Points: 202

Memory: 0.016 MB

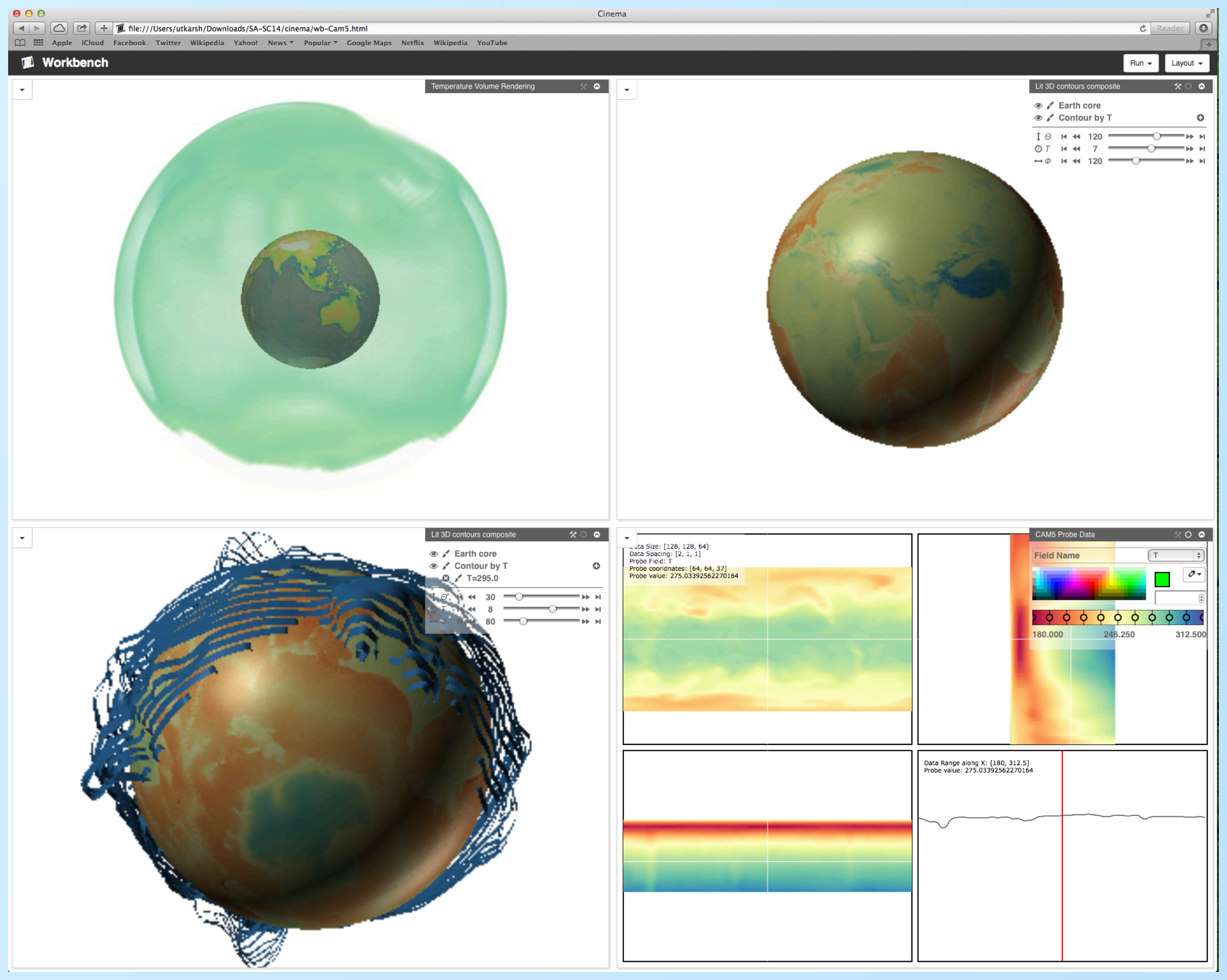
Data Arrays

Current data time: 0.467

Name	Data Type	Data Ranges

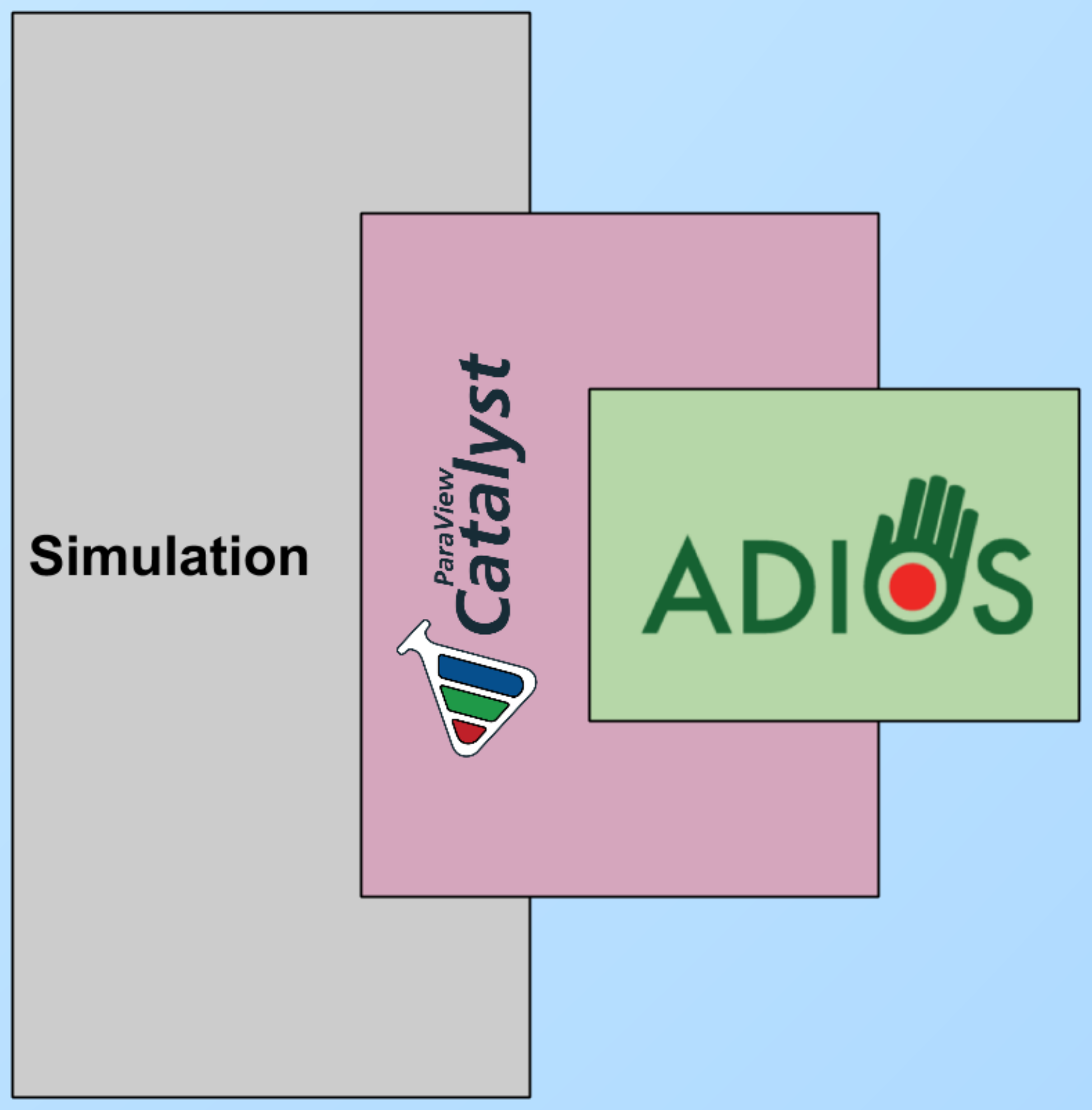
Cinema + Workbench

Cinema can simplify accessing Catalyst output. This is done by having Catalyst automatically generate a variety of images at each time step through exploring parameter spaces of camera angles, filter settings, etc. Then the images are viewed interactively through a web-browser



ADIOS Support

Catalyst can now write extracts generated from the in situ analysis pipeline using the high throughput ADIOS library. ADIOS greatly enhances Catalyst's I/O capabilities including supporting automatic tuning of Lustre file-system writing (auto-stripe size and block size adjustment) for more efficient use of storage resources; batching block write operations for higher throughput; aggregating IO to fewer ranks to consolidate smaller write operations from many processes into larger (and faster) I/O operations. Further advantages include array compression and write-on-change. We are currently working with ORNL to develop a method for automatic load balancing for continuously changing unbalanced IO. We have been performing scaling tests at ERDC on Garnet, a Cray XE6, and are actively evaluating the impact of various tuning parameters for the multiple ADIOS back-ends (MPI, MPI_LUSTRE, and MPI_AGGREGATE).



Simulation

ParaView Catalyst

ADIOS

SDAV Scalable Data Management, Analysis, and Visualization

U.S. DEPARTMENT OF ENERGY Office of Science