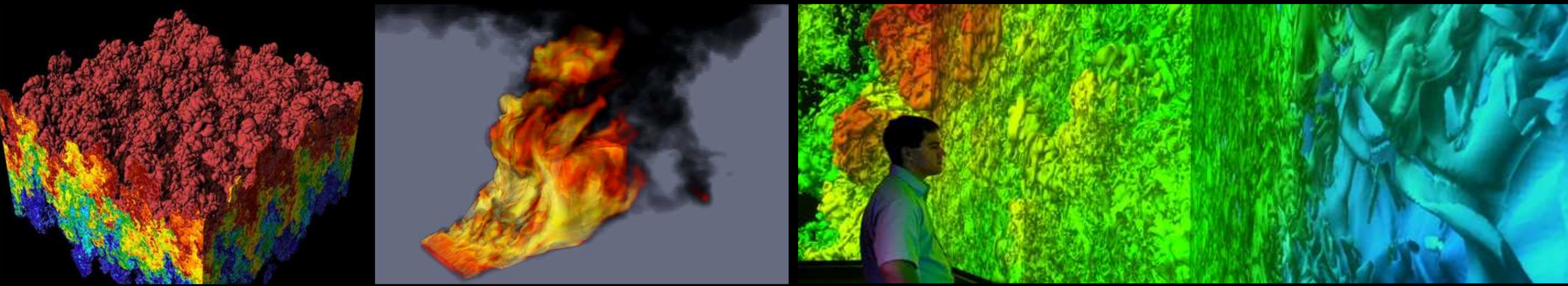
*Exceptional service in the national interest*

Sandia National Laboratories



# US DOE Research and Development in HPC Scientific Visualization

Kenneth Moreland Sandia National Laboratories

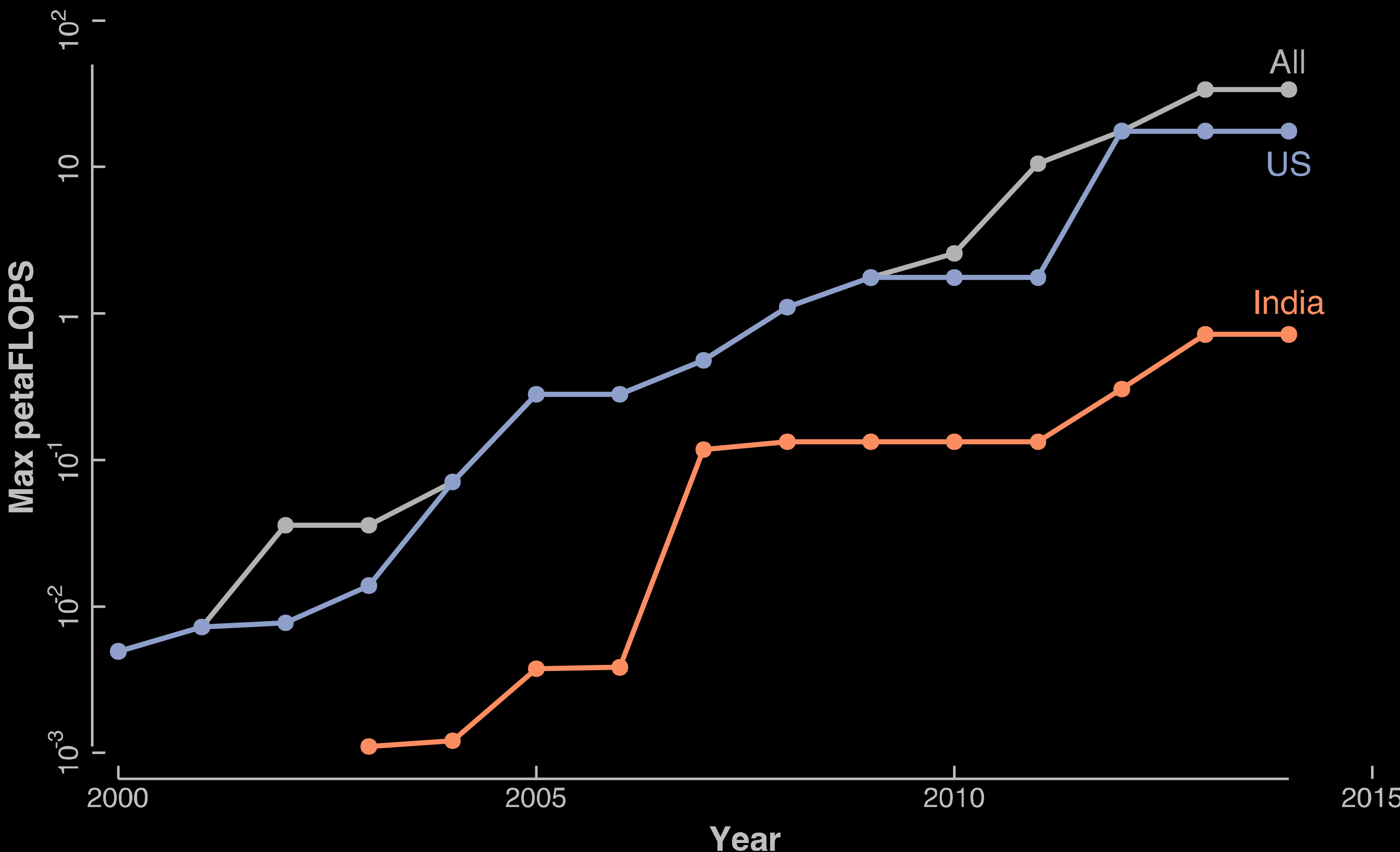
December 20, 2014

Indo-US Workshop on Virtual Institutes for Computational and Data-Enabled Science & Engineering

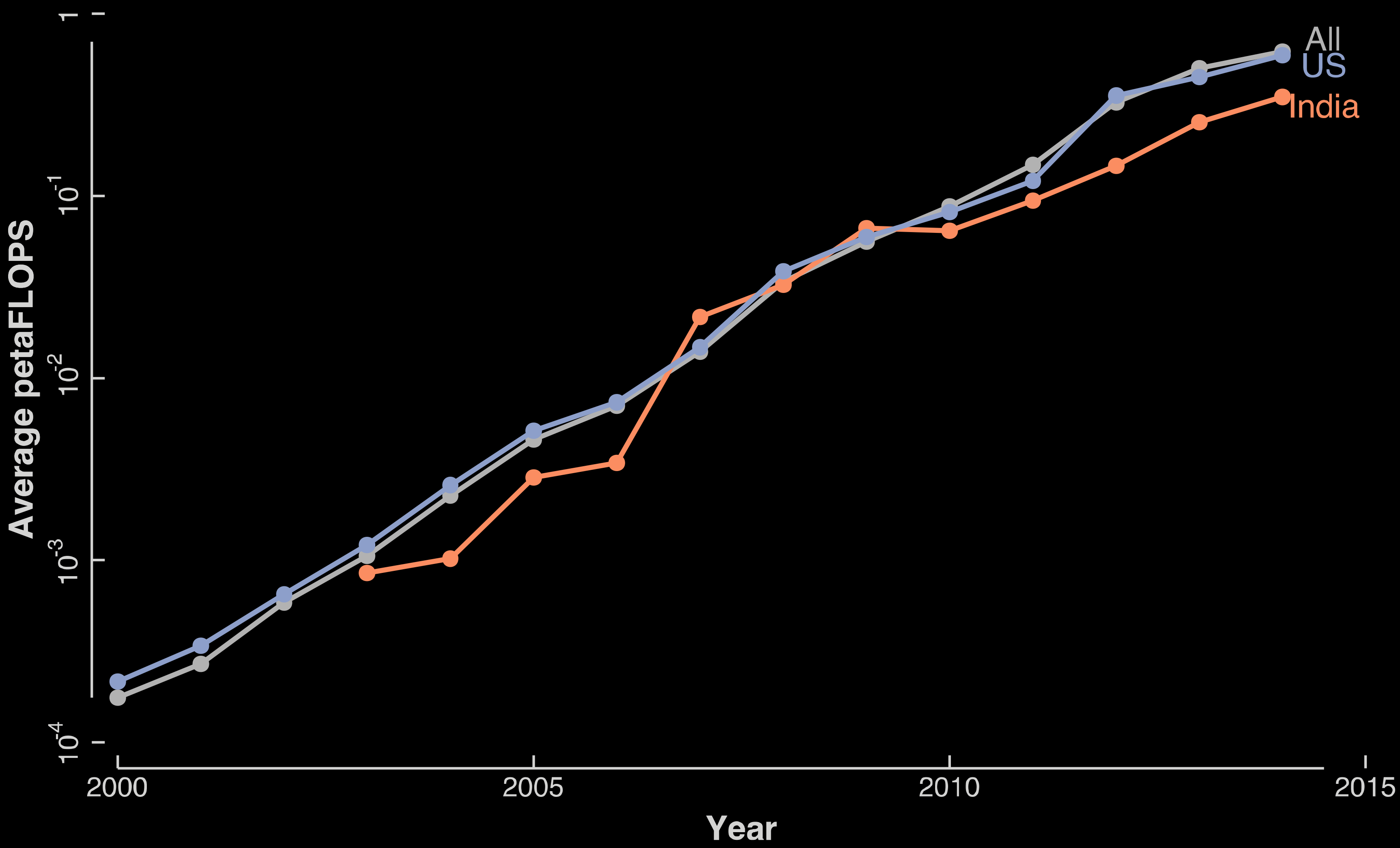
U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

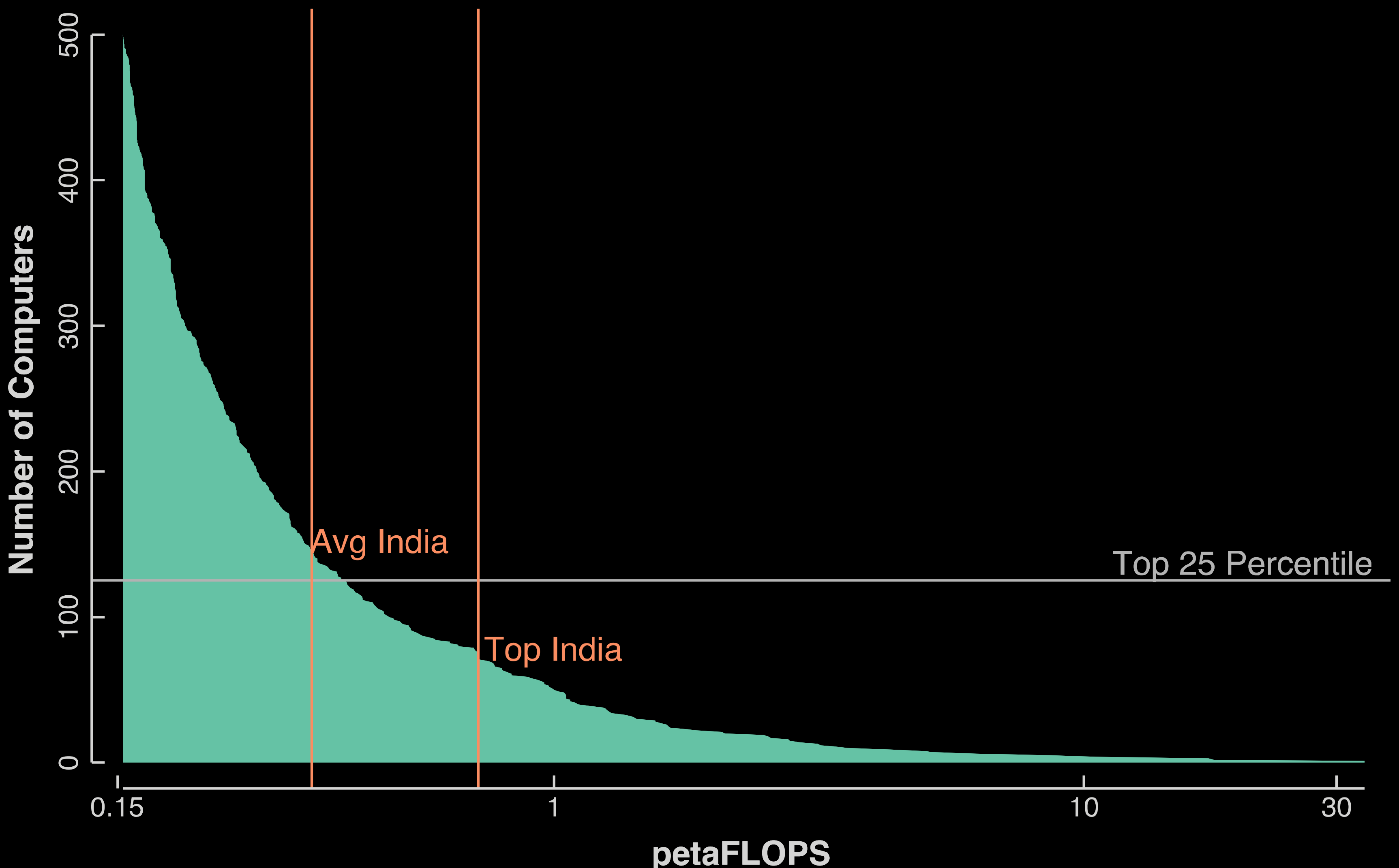
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP



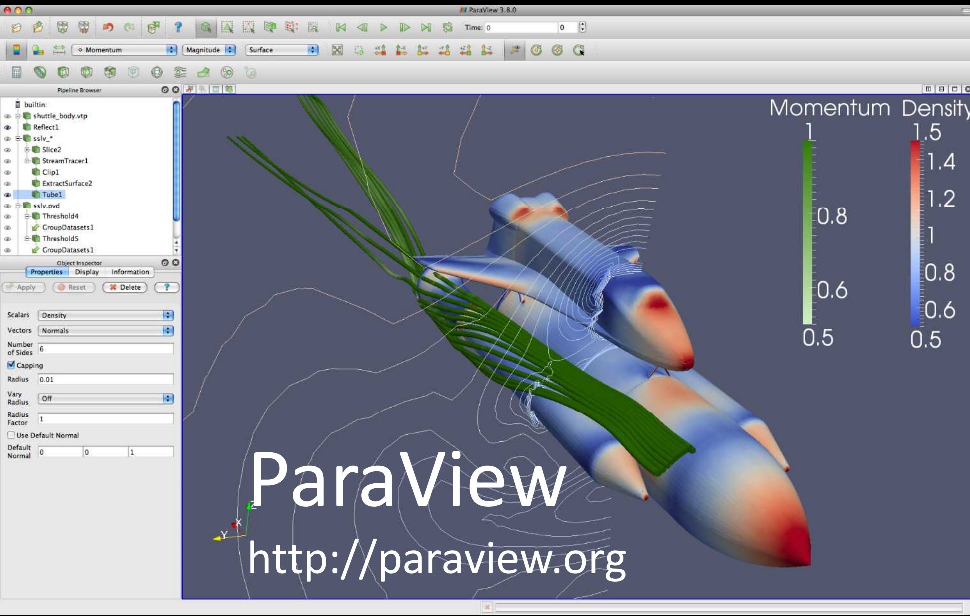

Source: Top500.org

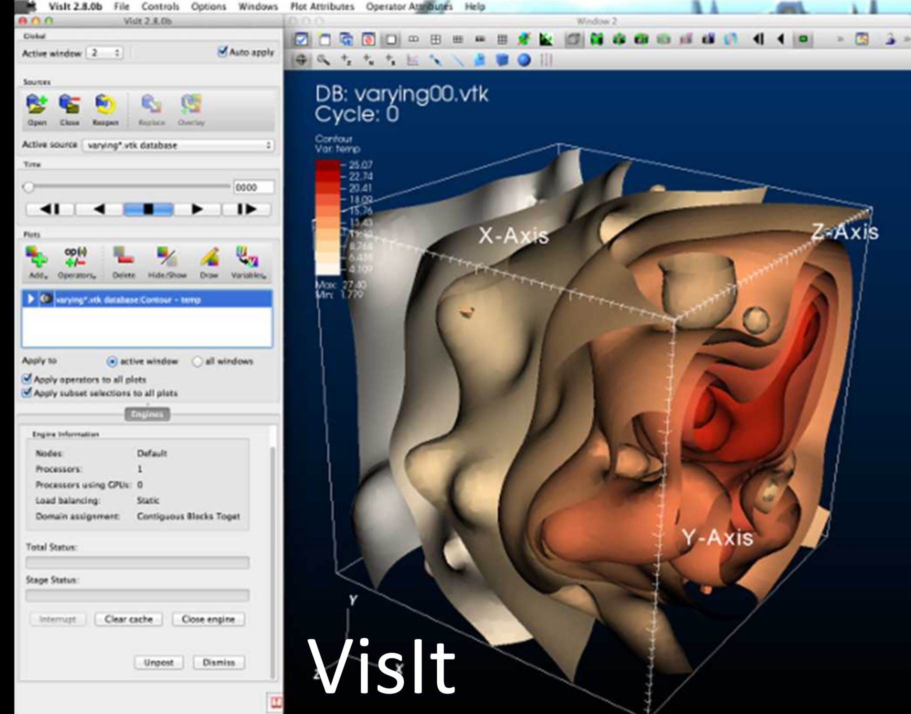



Source: Top500.org

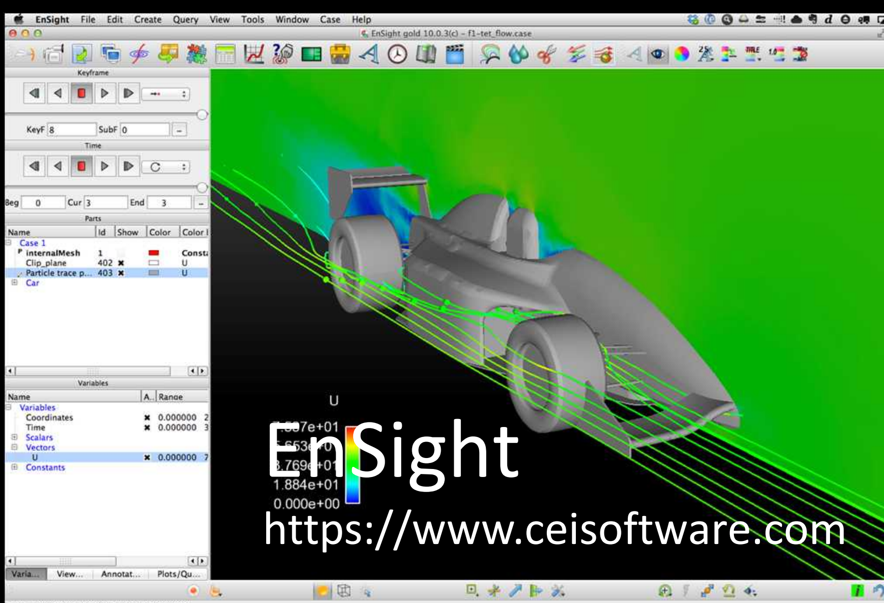



Source: Top500.org




http://visit.llnl.gov




FieldView

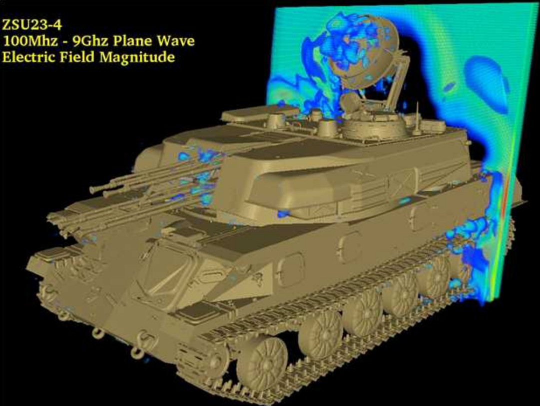
http://www.ilight.com

Vapor

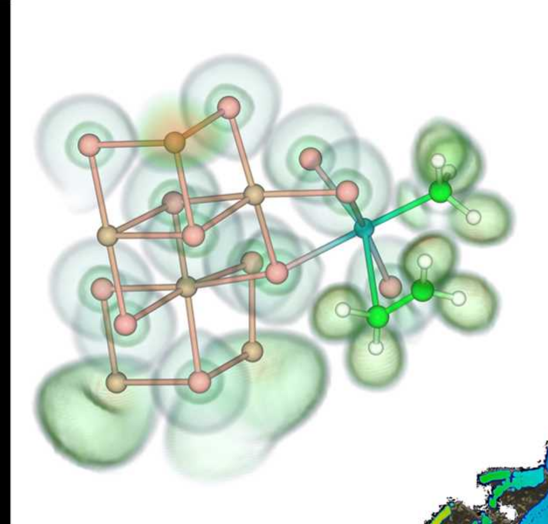
https://www.vapor.ucar.edu

VMD

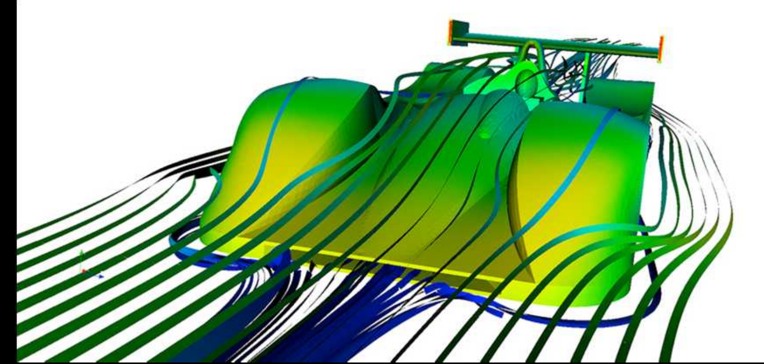
http://www.ks.uiuc.edu/Research/vmd



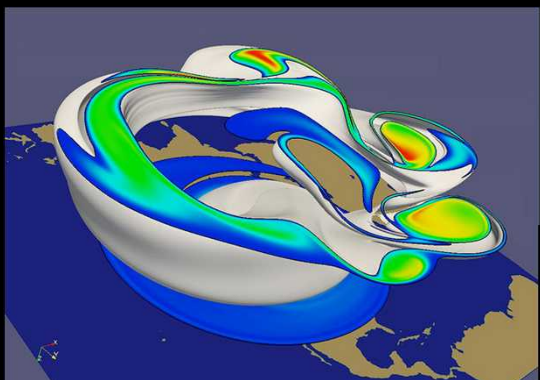

Jerry Clarke, US Army Research Laboratory

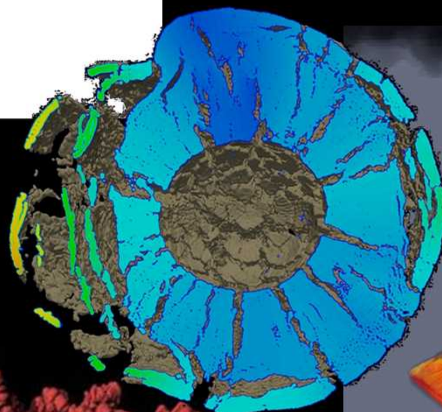


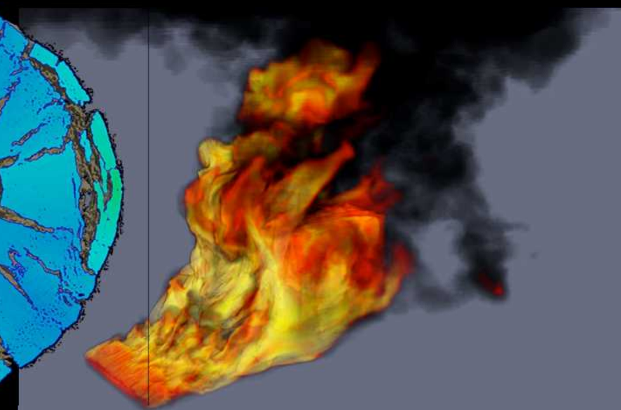
Swiss National Supercomputing Centre

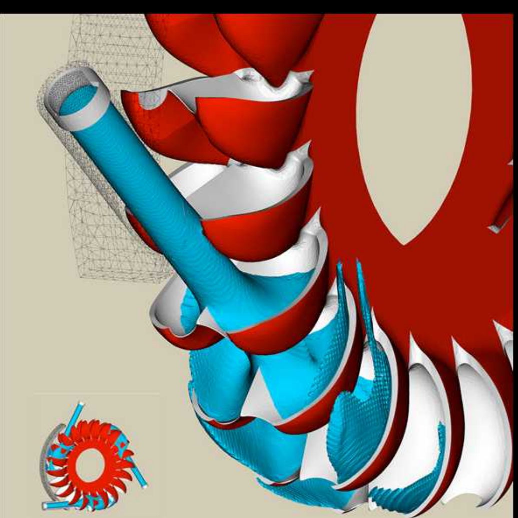


Renato N. Elias, NACAD/COPPE/UFRJ, Rio de Janerio, Brazil

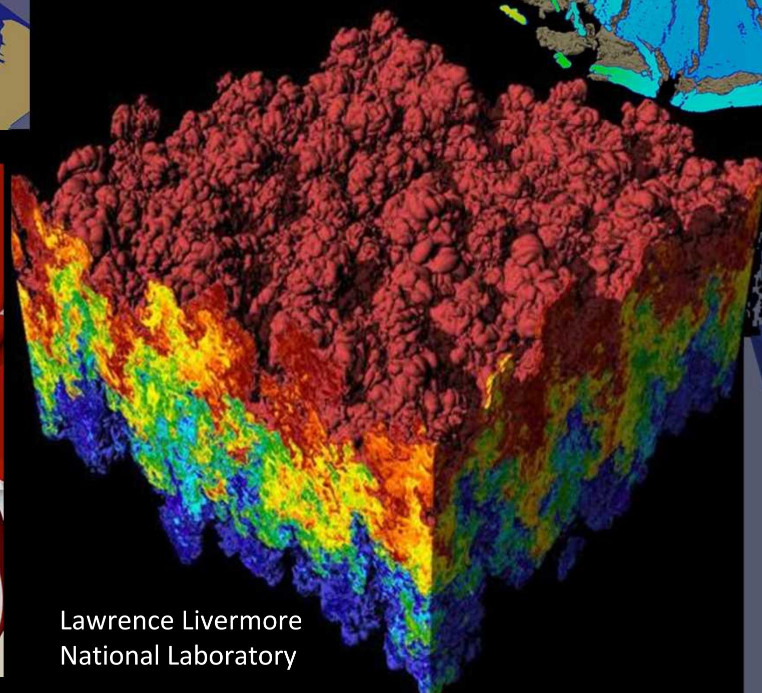




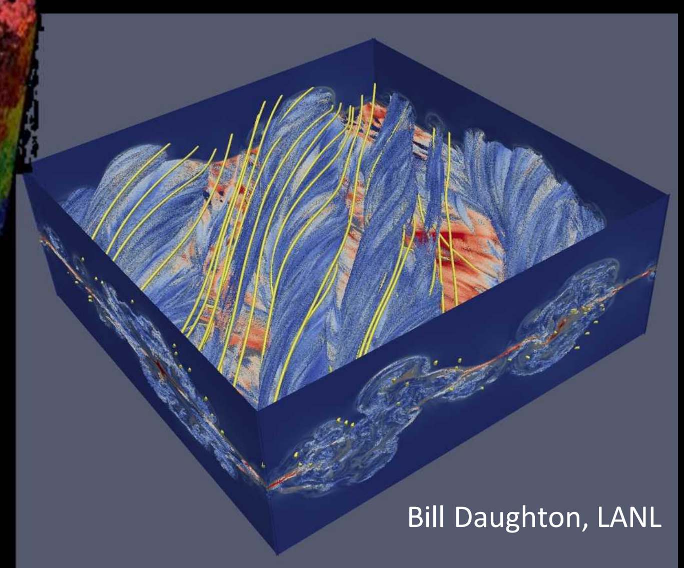




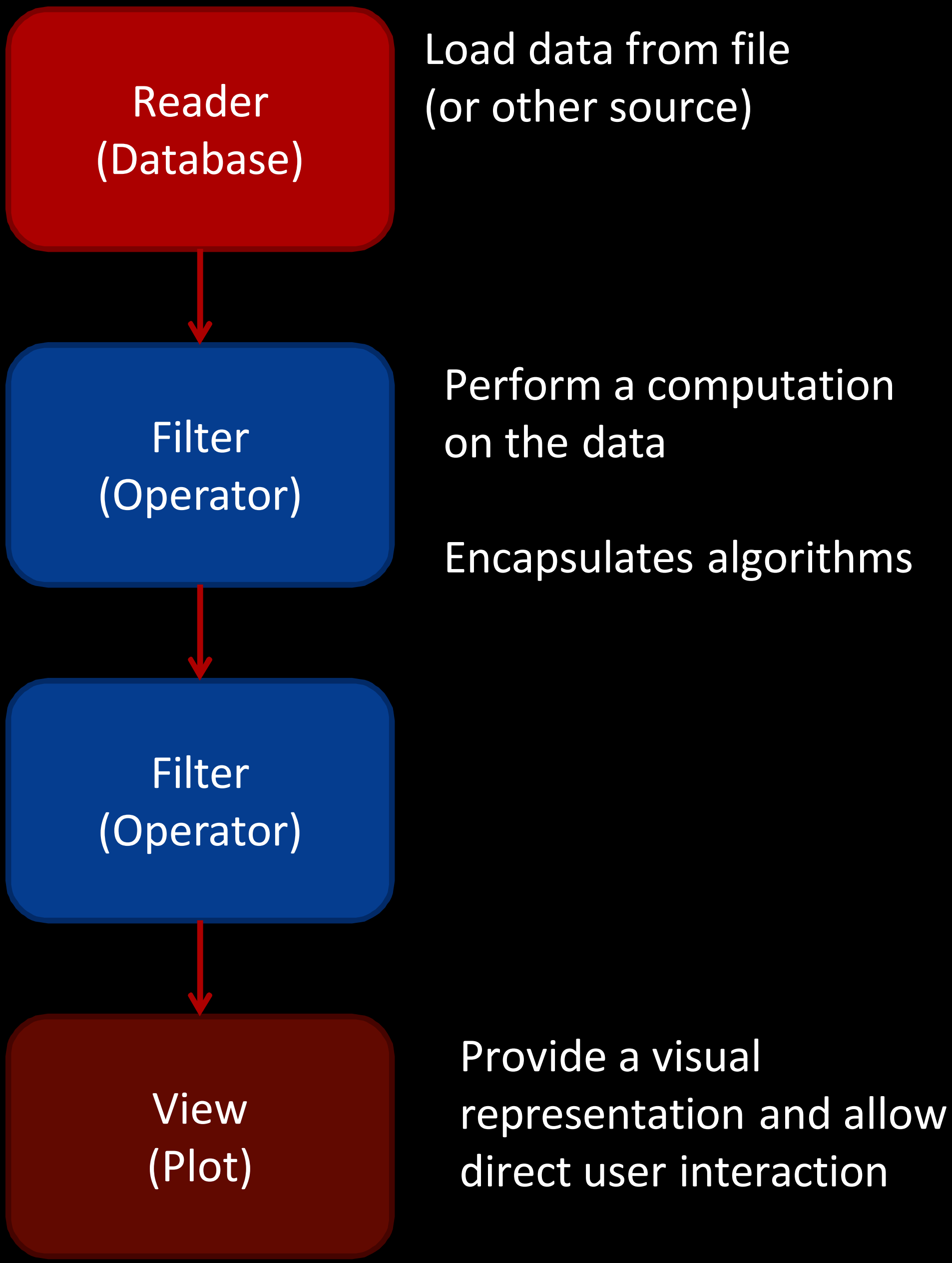
Swiss National Supercomputing Centre

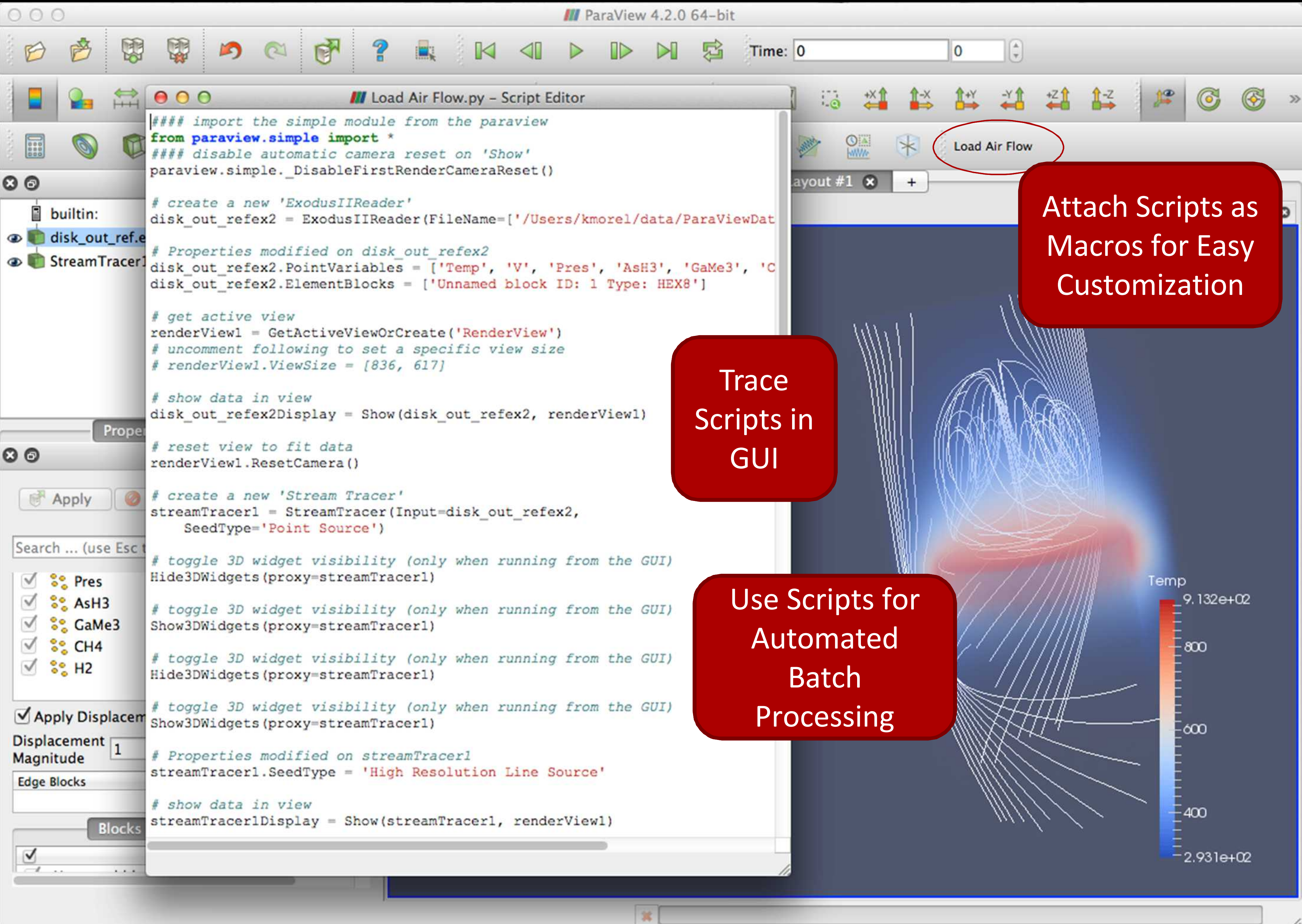


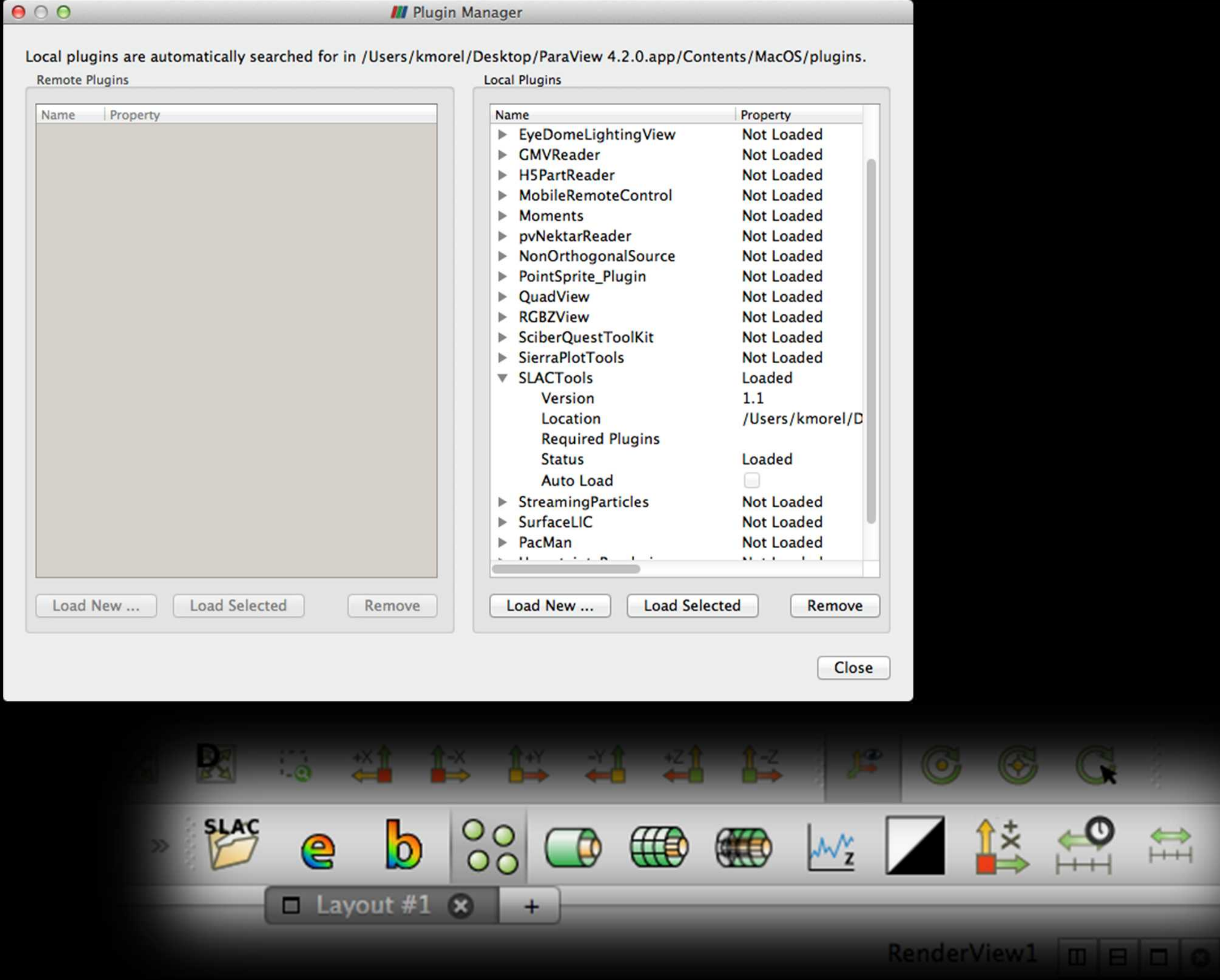
Lawrence Livermore National Laboratory

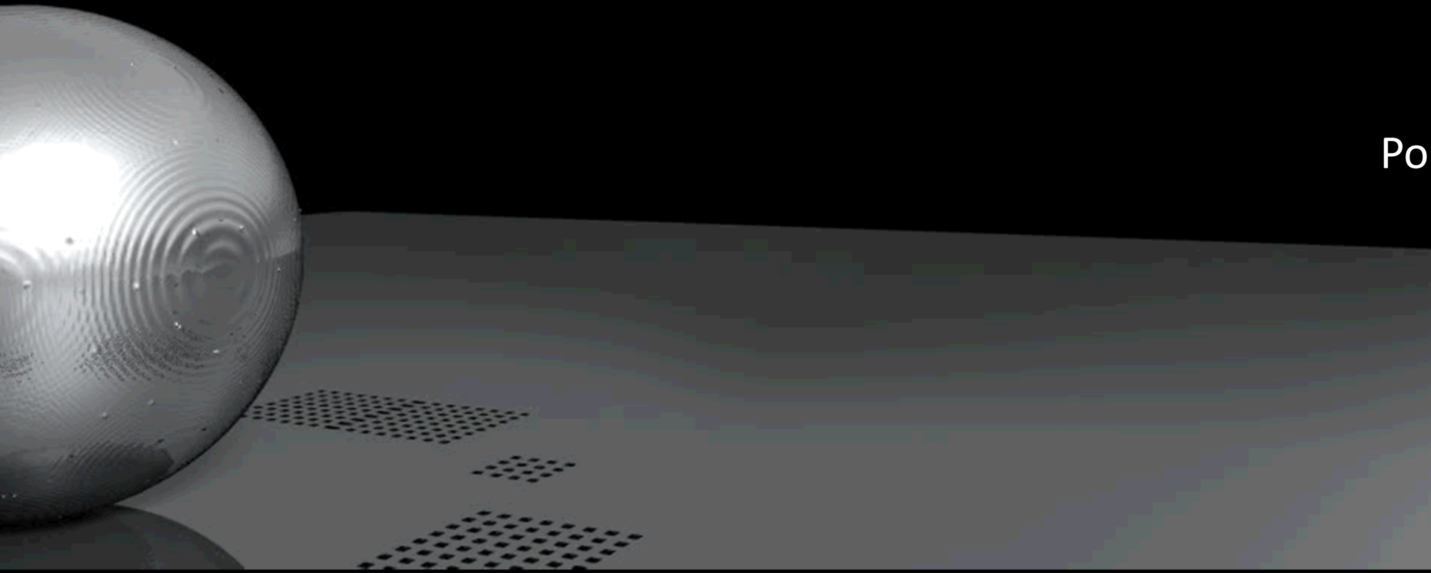


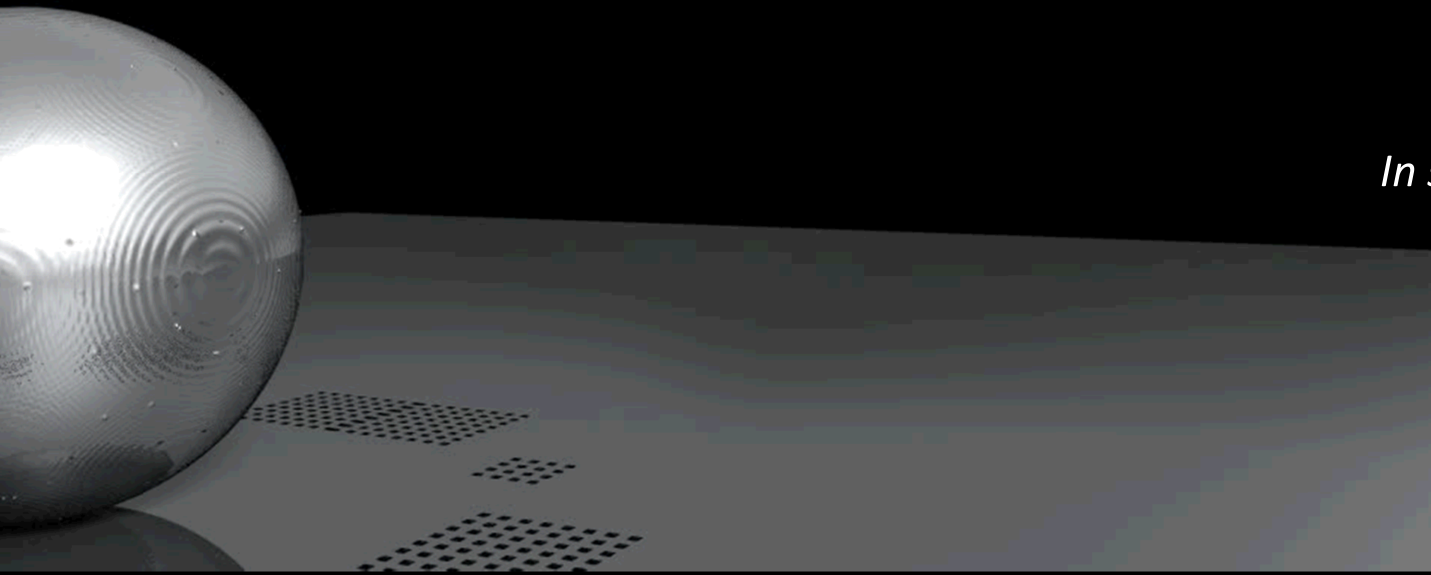
Bill Daughton, LANL


Reader
(Database)
Load data from file
(or other source)
Filter
(Operator)
Perform a computation
on the data
Encapsulates algorithms
Filter
(Operator)
View
(Plot)
Provide a visual
representation and allow
direct user interaction


ParaView 4.2.0 64-bit
Time: 0
Load Air Flow.py - Script Editor
Load Air Flow
Attach Scripts as Macros for Easy Customization
Trace Scripts in GUI
Use Scripts for Automated Batch Processing
#### import the simple module from the paraview
from paraview.simple import *
#### disable automatic camera reset on 'Show'
paraview.simple._DisableFirstRenderCameraReset()
# create a new 'ExodusIIReader'
disk_out_refex2 = ExodusIIReader(FileName=['/Users/kmorel/data/ParaViewDat
# Properties modified on disk_out_refex2
disk_out_refex2.PointVariables = ['Temp', 'V', 'Pres', 'AsH3', 'GaMe3', 'C
disk_out_refex2.ElementBlocks = ['Unnamed block ID: 1 Type: HEX8']
# get active view
renderView1 = GetActiveViewOrCreate('RenderView')
# uncomment following to set a specific view size
# renderView1.ViewSize = [836, 617]
# show data in view
disk_out_refex2Display = Show(disk_out_refex2, renderView1)
# reset view to fit data
renderView1.ResetCamera()
# create a new 'Stream Tracer'
streamTracer1 = StreamTracer(Input=disk_out_refex2,
SeedType='Point Source')
# toggle 3D widget visibility (only when running from the GUI)
Hide3DWidgets(proxy=streamTracer1)
# toggle 3D widget visibility (only when running from the GUI)
Show3DWidgets(proxy=streamTracer1)
# toggle 3D widget visibility (only when running from the GUI)
Hide3DWidgets(proxy=streamTracer1)
# toggle 3D widget visibility (only when running from the GUI)
Show3DWidgets(proxy=streamTracer1)
# Properties modified on streamTracer1
streamTracer1.SeedType = 'High Resolution Line Source'
# show data in view
streamTracer1Display = Show(streamTracer1, renderView1)
builtin:
disk_out_ref.e
StreamTracer1
Apply
Pres
AsH3
GaMe3
CH4
H2
Apply Displacem
Displacement Magnitude
Edge Blocks
Blocks
Temp
9.132e+02
800
600
400
2.931e+02


Plugin Manager
Local plugins are automatically searched for in /Users/kmorel/Desktop/ParaView 4.2.0.app/Contents/MacOS/plugins.
Remote Plugins
Name
Property
Local Plugins
Name
Property
EyeDomeLightingView Not Loaded
GMVReader Not Loaded
H5PartReader Not Loaded
MobileRemoteControl Not Loaded
Moments Not Loaded
pvNektarReader Not Loaded
NonOrthogonalSource Not Loaded
PointSprite_Plugin Not Loaded
QuadView Not Loaded
RGBZView Not Loaded
SciberQuestToolKit Not Loaded
SierraPlotTools Not Loaded
SLACTools Loaded
Version 1.1
Location /Users/kmorel/D
Required Plugins
Status Loaded
Auto Load
StreamingParticles Not Loaded
SurfaceLIC Not Loaded
PacMan Not Loaded
Load New ...
Load Selected
Remove
Load New ...
Load Selected
Remove
Close
SLAC
Layout #1
RenderView1




Roughly equal data stored at simulation time

*Reflections and shadows added in post-processing for both examples*

# Extreme Scale Computing

Sandia National Laboratories

- Trends: More FLOPS with comparatively less storage, I/O bandwidth
  - Consequence: A smaller fraction of data can be captured on disk

## Oak Ridge National Laboratory

| | System Peak | I/O BW |
|---|---|---|
| Jaguar (2008) | 263 TFLOPS | 44 GB/s |
| Jaguar PF (2009) | 1.75 PFLOPS | 240 GB/s |
| Titan (2012) | 20 PFLOPS | 240 GB/s |
| **Factor Change** | **76×** | **5.5×** |

Bland, Kendall, Kothe, Rogers, and Shipman. "Jaguar: The World's Most Powerful Computer"
http://archive.hpcwire.com/hpcwire/2012-10-29/titan_sets_high-water_mark_for_gpu_supercomputing.html?featured=top

## Argonne National Laboratory

| | System Peak | I/O BW |
|---|---|---|
| Intrepid (2003) | 560 TFLOPS | 88 GB/s |
| Mira (2011) | 10 PFLOPS | 240 GB/s |
| **Factor Change** | **17.8×** | **2.7×** |

https://www.alcf.anl.gov/intrepid
https://www.alcf.anl.gov/mira

## Lawrence Livermore National Laboratory

| | System Peak | I/O BW |
|---|---|---|
| ASC Purple (2005) | 100 TFLOPS | 106 GB/s |
| Sequoia (2012) | 20 PFLOPS | 1 TB/s |
| **Factor Change** | **200×** | **9.4×** |

http://www.sandia.gov/supercomp/sc2002/flyers/SC02ASCIPurplev4.pdf
https://asc.llnl.gov/publications/Sequoia2012.pdf

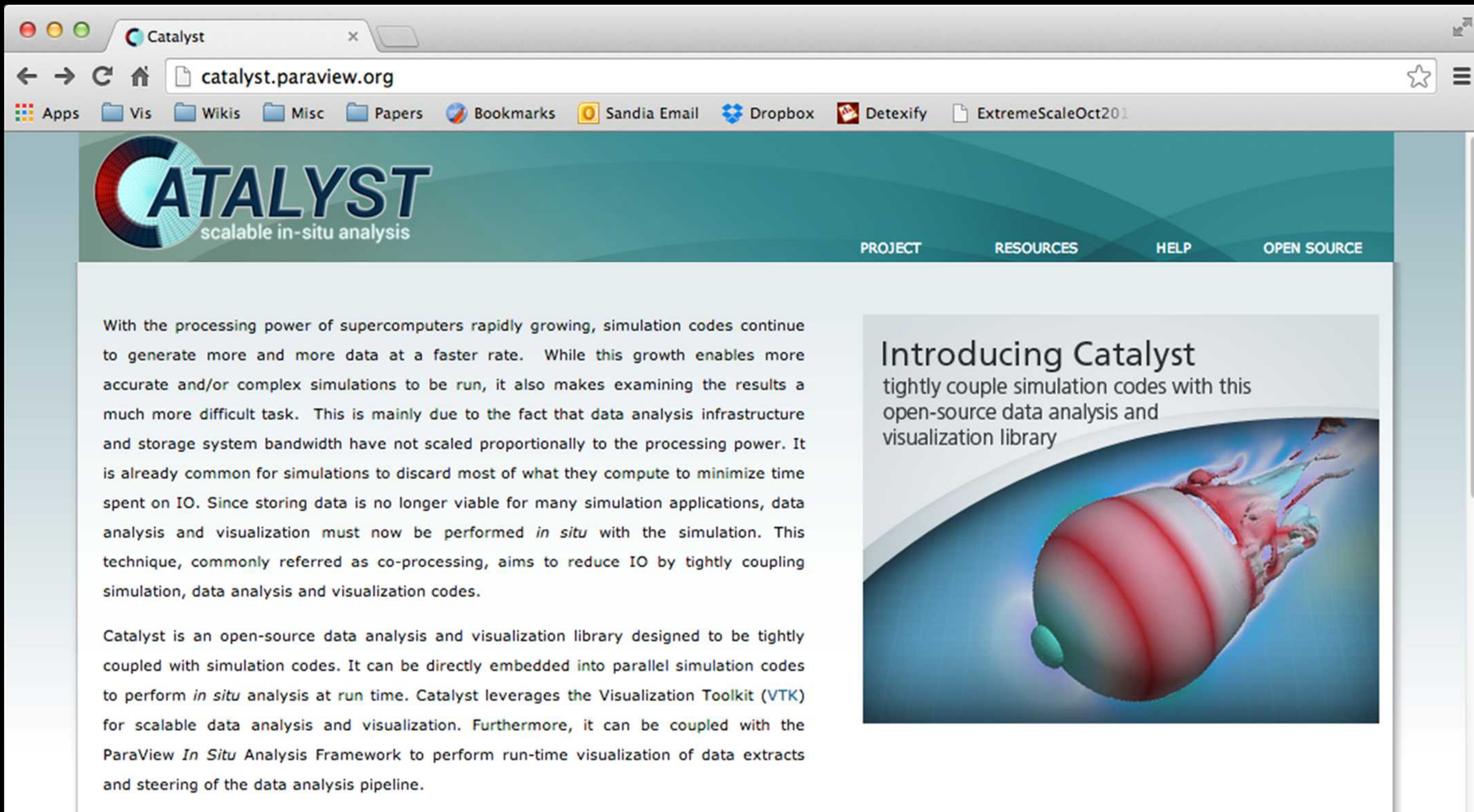
## Sandia National Laboratories

| | System Peak | I/O BW |
|---|---|---|
| Red Storm (2003) | 180 TFLOPS | 100 GB/s |
| Cielo (2011) | 1.4 PFLOPS | 160 GB/s |
| **Factor Change** | **7.8×** | **1.6×** |

https://cfwebprod.sandia.gov/cfdocs/CCIM/docs/033768p.pdf
http://www.lanl.gov/orgs/hpc/cielo/

# The Catalyst In Situ Library

Sandia National Laboratories


Catalyst

catalyst.paraview.org

Apps | Vis | Wikis | Misc | Papers | Bookmarks | Sandia Email | Dropbox | Detexify | ExtremeScaleOct20

CATALYST
scalable in-situ analysis

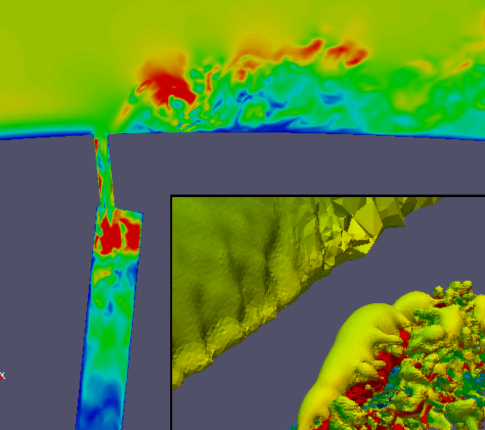
PROJECT | RESOURCES | HELP | OPEN SOURCE

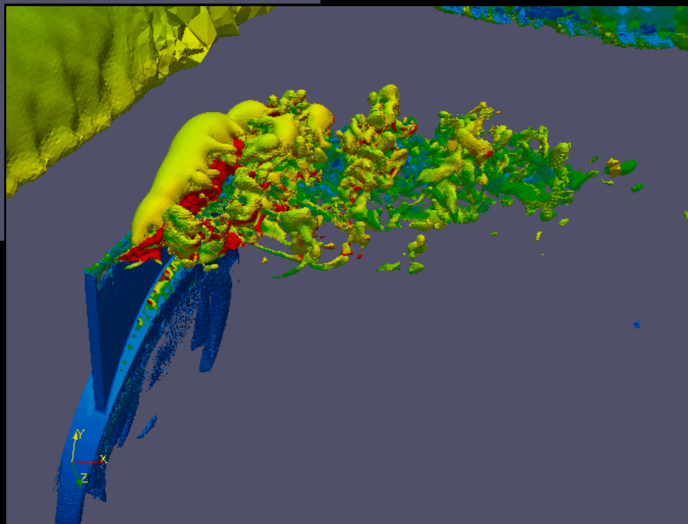
With the processing power of supercomputers rapidly growing, simulation codes continue to generate more and more data at a faster rate. While this growth enables more accurate and/or complex simulations to be run, it also makes examining the results a much more difficult task. This is mainly due to the fact that data analysis infrastructure and storage system bandwidth have not scaled proportionally to the processing power. It is already common for simulations to discard most of what they compute to minimize time spent on IO. Since storing data is no longer viable for many simulation applications, data analysis and visualization must now be performed *in situ* with the simulation. This technique, commonly referred as co-processing, aims to reduce IO by tightly coupling simulation, data analysis and visualization codes.

Catalyst is an open-source data analysis and visualization library designed to be tightly coupled with simulation codes. It can be directly embedded into parallel simulation codes to perform *in situ* analysis at run time. Catalyst leverages the Visualization Toolkit (VTK) for scalable data analysis and visualization. Furthermore, it can be coupled with the ParaView *In Situ* Analysis Framework to perform run-time visualization of data extracts and steering of the data analysis pipeline.

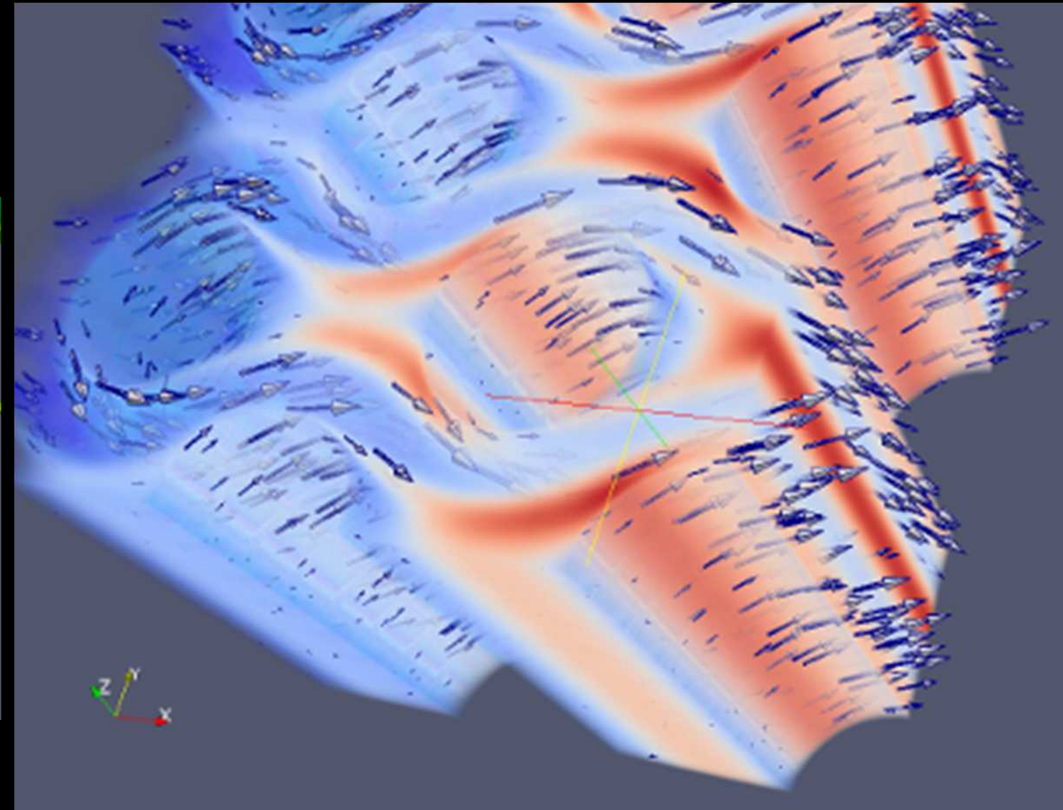
## Introducing Catalyst

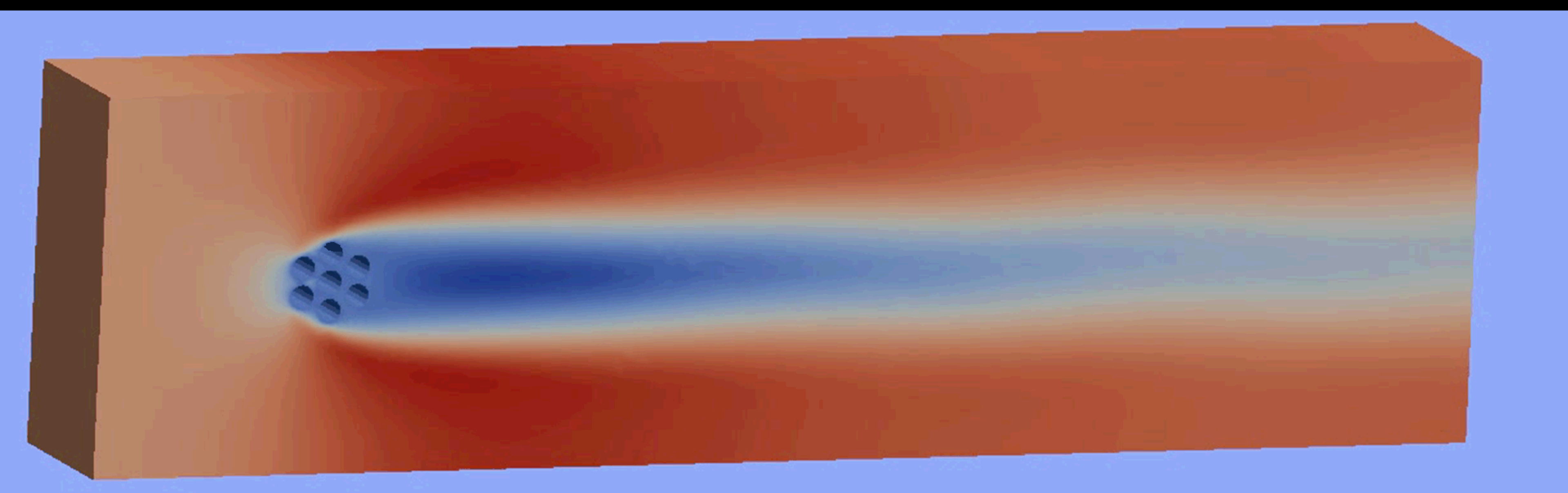
tightly couple simulation codes with this open-source data analysis and visualization library

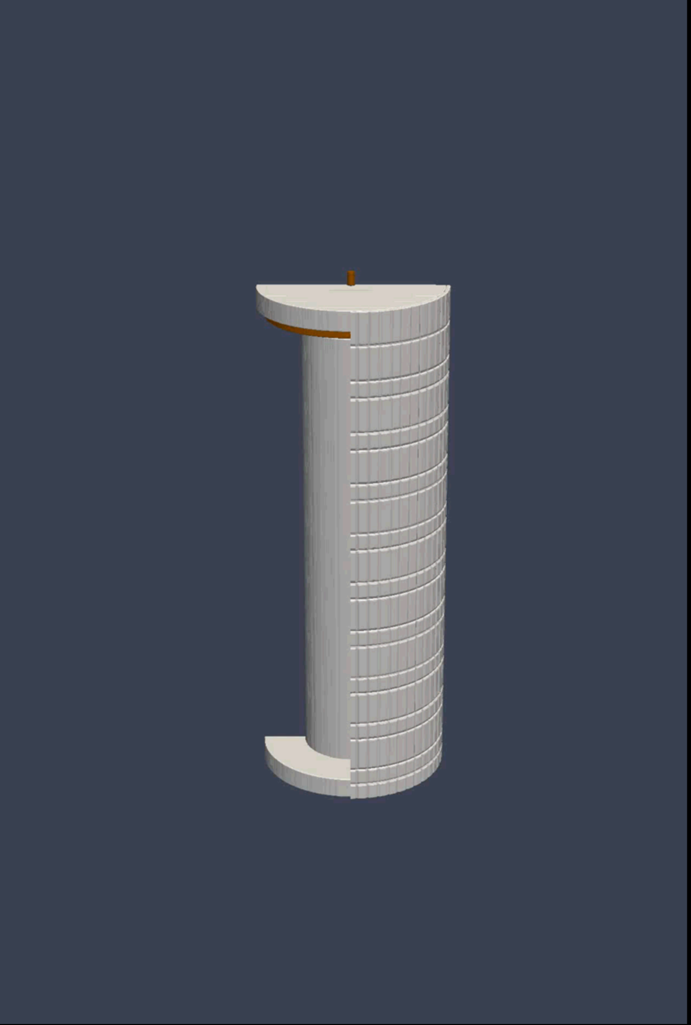


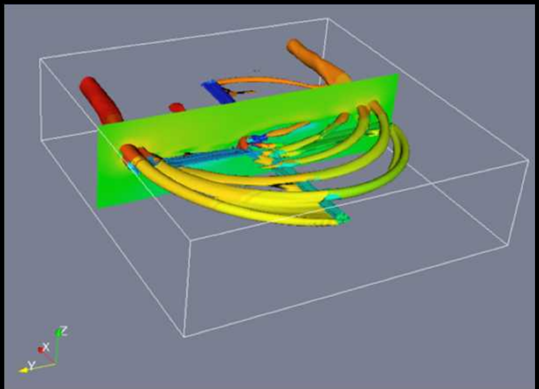


Lorendeau, Fournier, and Ribes. “In-Situ visualization in fluid mechanics using Catalyst: a case study for Code_Saturne.”

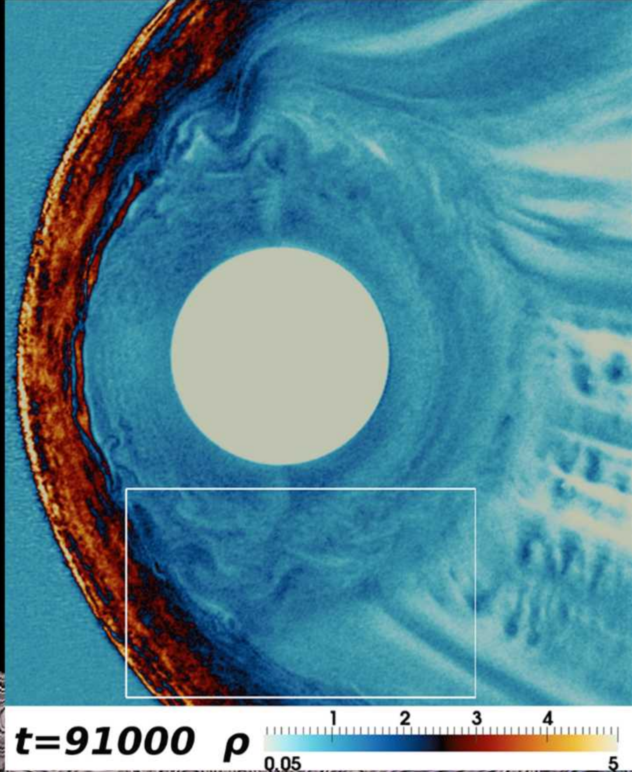


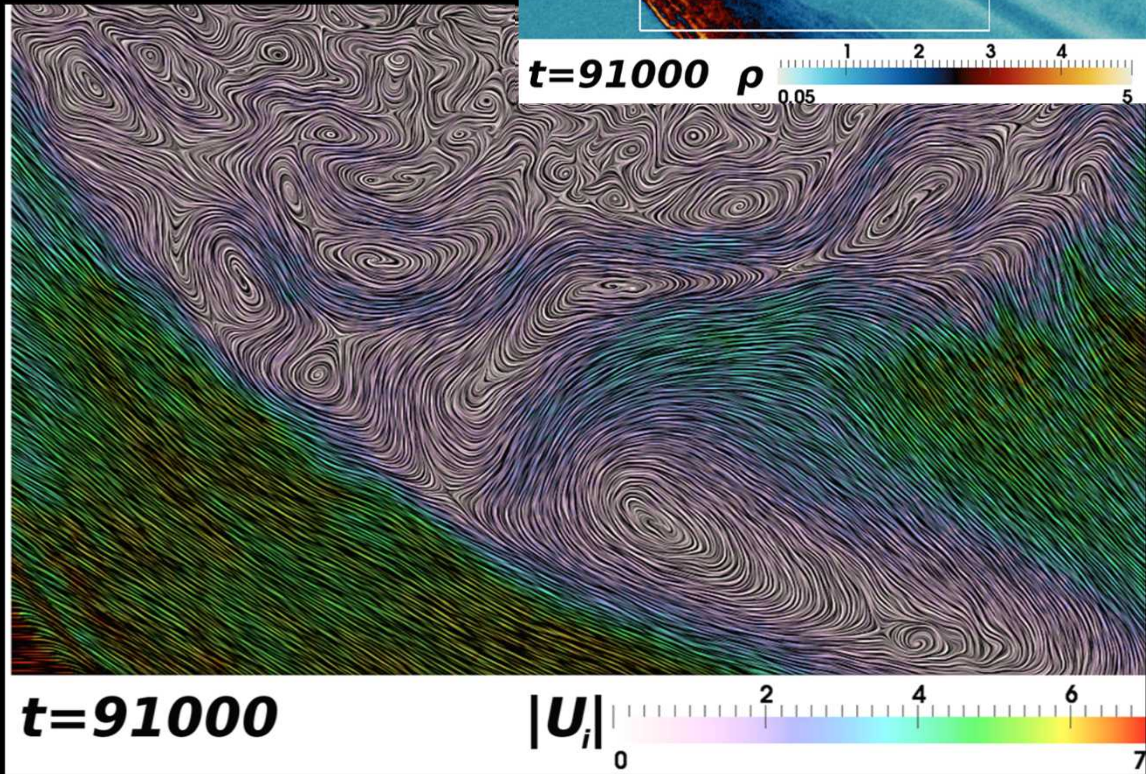


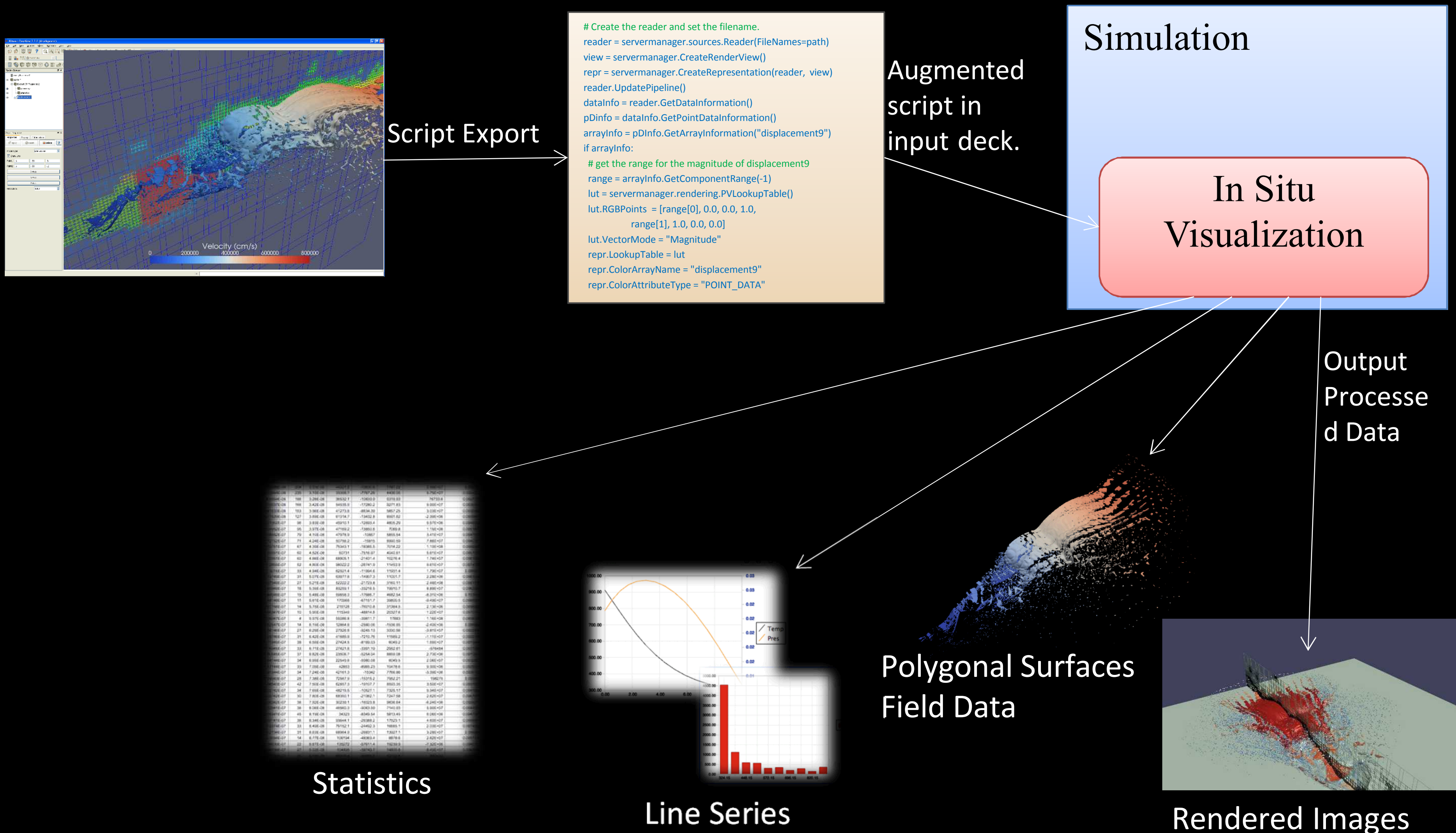


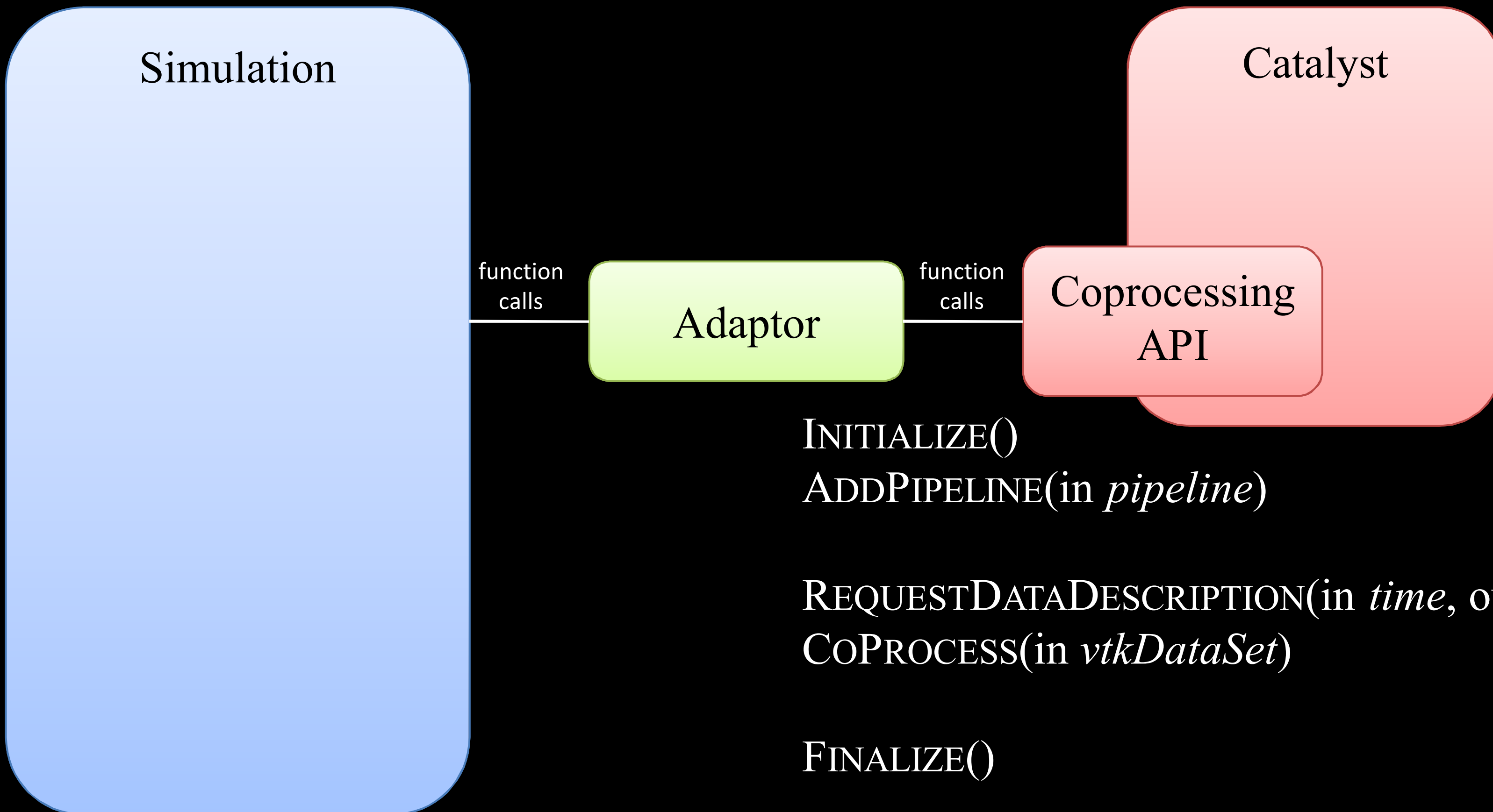


Krimabadi, O'Leary, Tatineni, Loring, Majumdar, and Geveci. "In-Situ Visualization for Global Hybrid Systems."




Script Export
# 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]
lut.VectorMode = "Magnitude"
repr.LookupTable = lut
repr.ColorArrayName = "displacement9"
repr.ColorAttributeType = "POINT_DATA"
Augmented script in input deck.
Simulation
In Situ Visualization
Output Processe d Data
Velocity (cm/s)
Statistics
Line Series
Polygonal Surfaces Field Data
Rendered Images




Initialize()
AddPipeline(in *pipeline*)

RequestDataDescription(in *time*, out *fields*)
CoProcess(in *vtkDataSet*)

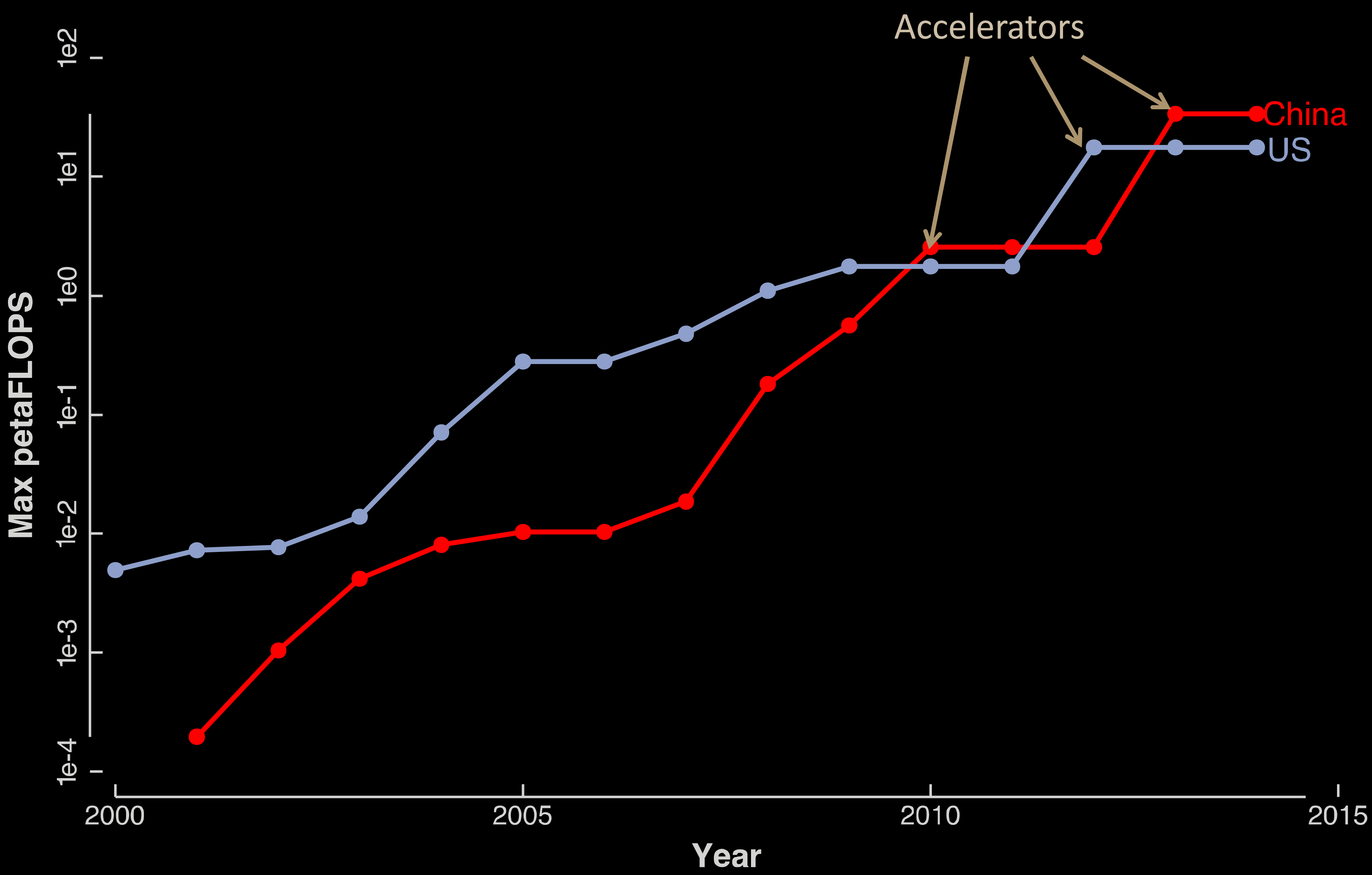
Finalize()

My new computer's got the clocks, it rocks
But it was obsolete before I opened the box

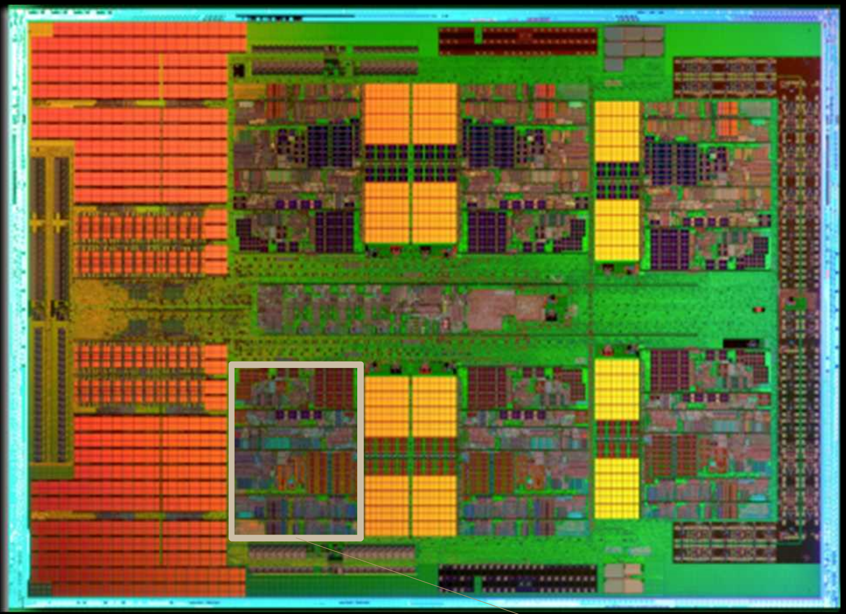
– “Weird” Al Yankovic, *It’s All About the Pentiums*, circa 1999

Moore’s Law is dead.

– Gordon Moore, circa 2005




Source: Top500.org



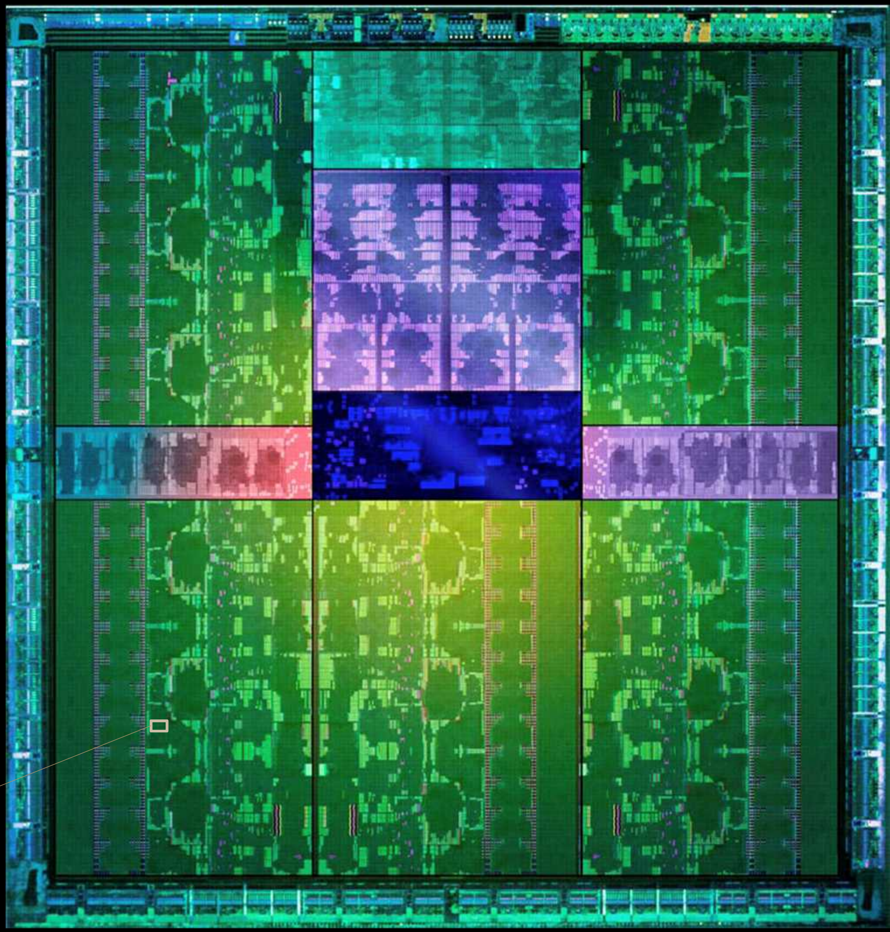
1mm

# AMD x86

Full x86 Core
 + Associated Cache
6 cores per die
MPI-Only feasible

1 x86 core

1 Kepler "core"

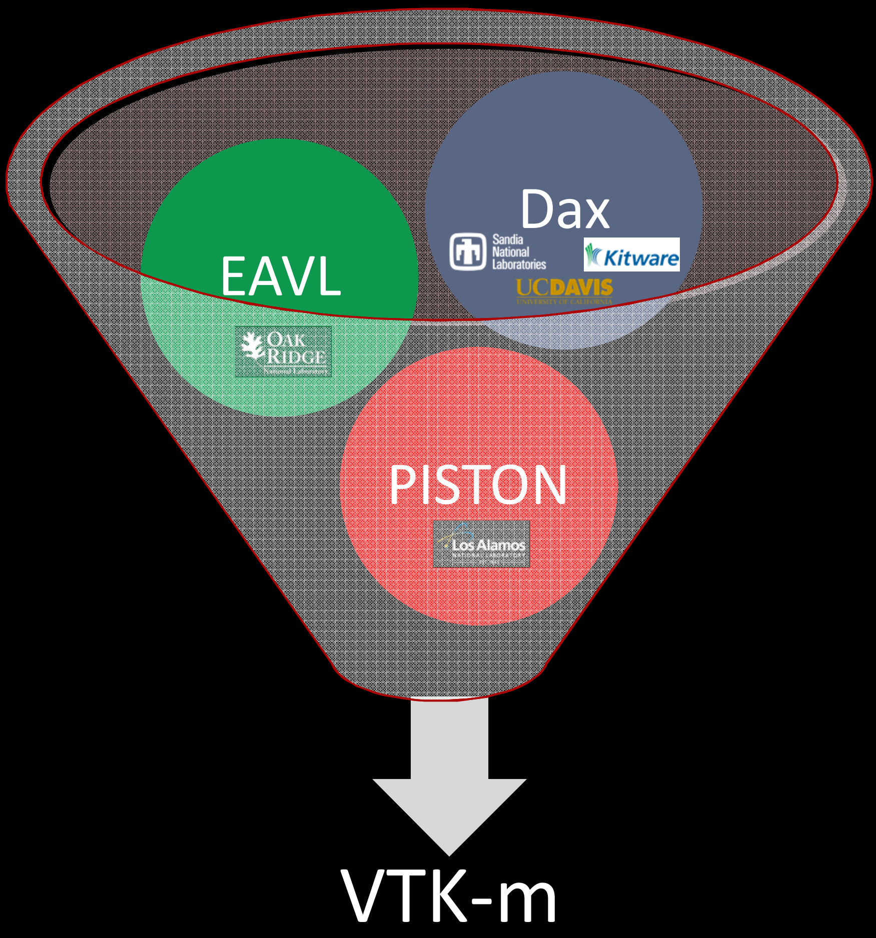


# NVIDIA GPU

2,880 cores collected in 15 SMX
Shared PC, Cache, Mem Fetches
Reduced control logic
MPI-Only not feasible

"Everybody who learns concurrency thinks they understand it, ends up finding mysterious races they thought weren't possible, and discovers that they didn't actually understand it yet after all."

–Herb Sutter


EAVL
Dax
PISTON
VTK-m

# Collaboration Opportunity Summary

- New/Customized Visualization Algorithms
  - Most of US DOE and academics collaborate on a small collection of scientific visualization software tools that are open, modular, and parallel. These are a good entry point to software collaboration.
- In Situ Visualization
  - As HPC systems advance, in situ visualization becomes more critical. In situ visualization also typically requires specific coupling of codes, which is an important collaboration point.
- Emerging Architectures
  - Changes in computer processors are forcing changes in our software and algorithms. Such changes are still in research and development, and collaborating now can have a large impact on future code.