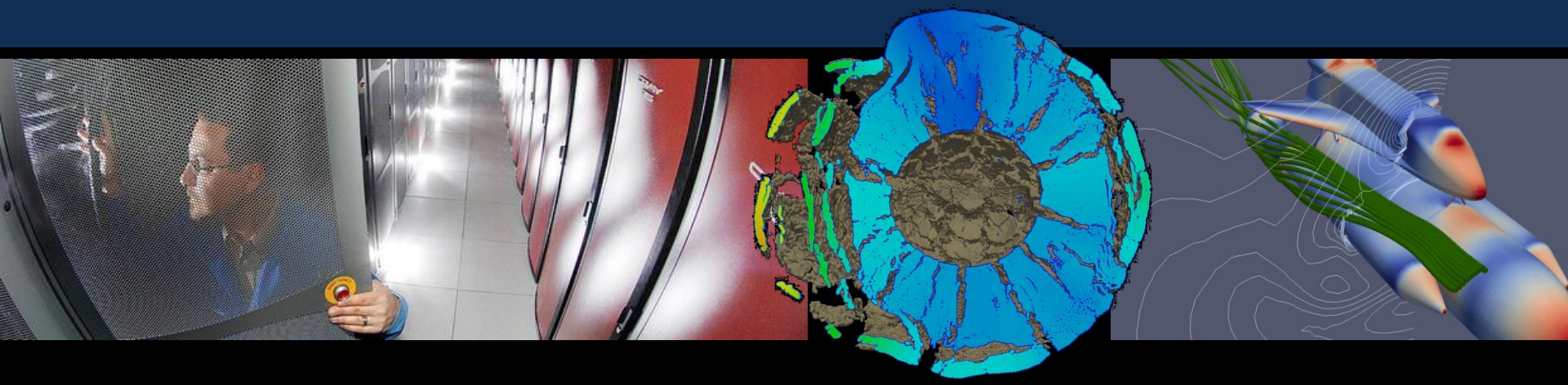
Exceptional service in the national interest

Sandia National Laboratories



Scientific Visualization Then and Now

NITRD Frontiers of Visualization

May 2, 2014

Kenneth Moreland Sandia National Laboratories

SDAV

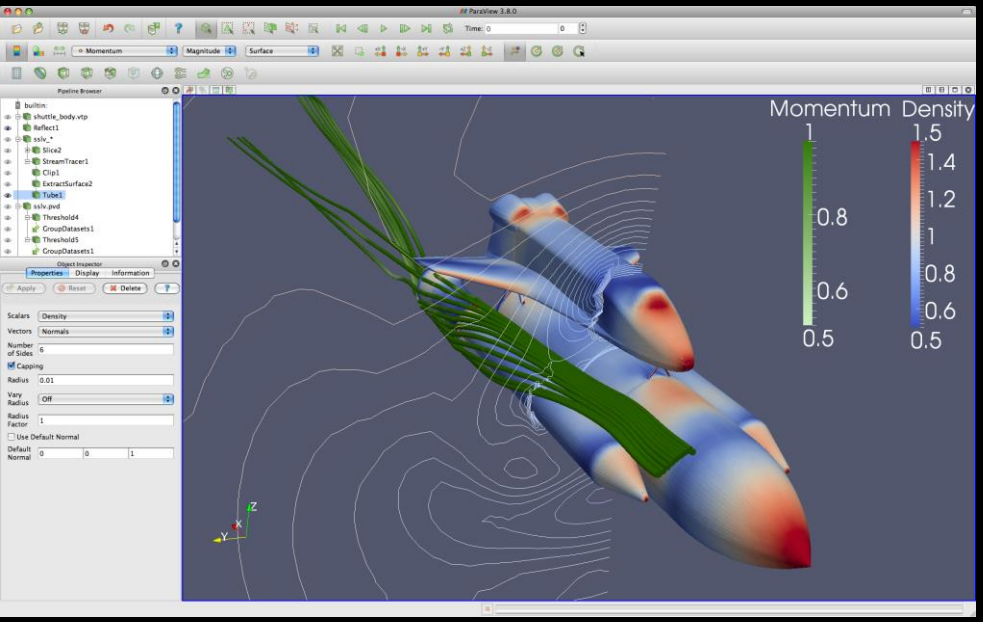
U.S. DEPARTMENT OF ENERGY

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. 2014-3581P

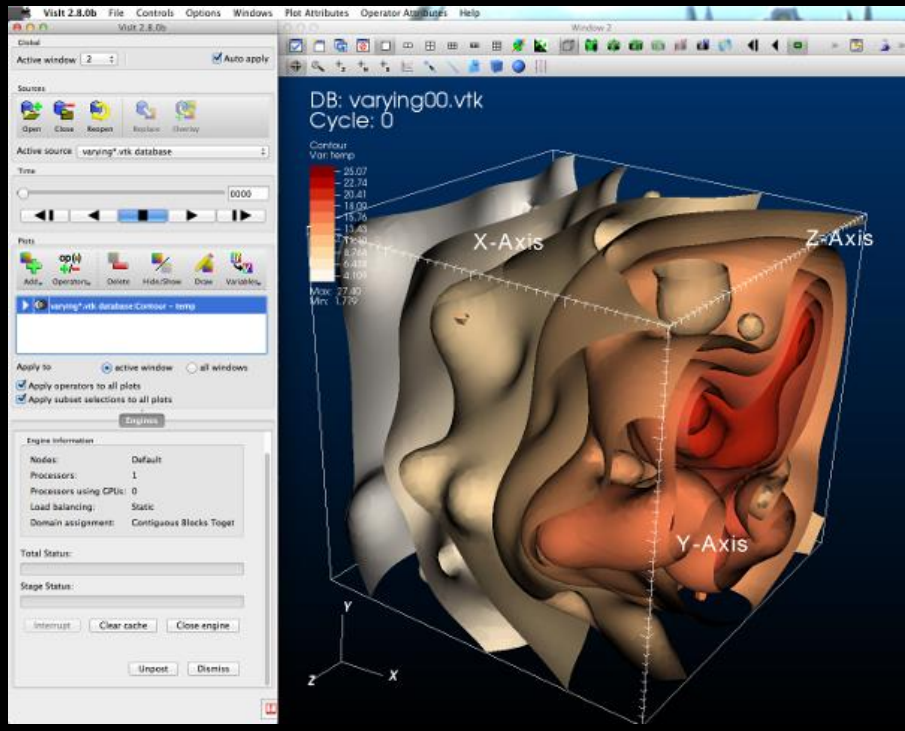
What I Do

Sandia National Laboratories



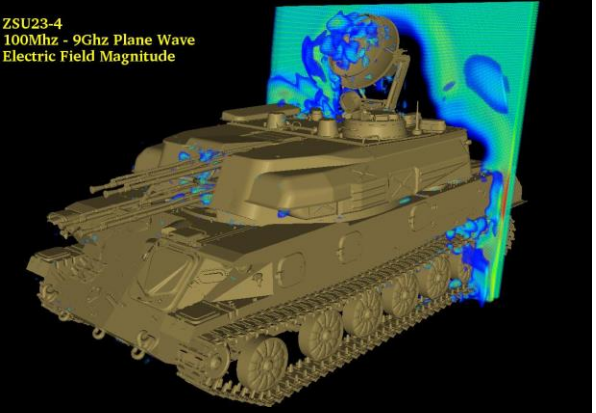
ParaView

http://paraview.org

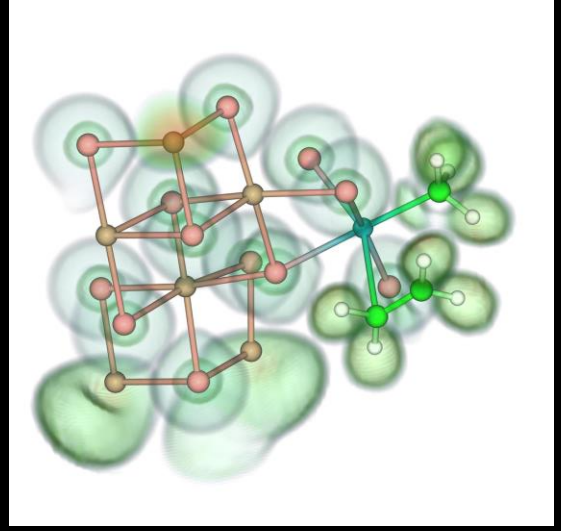


VisIt

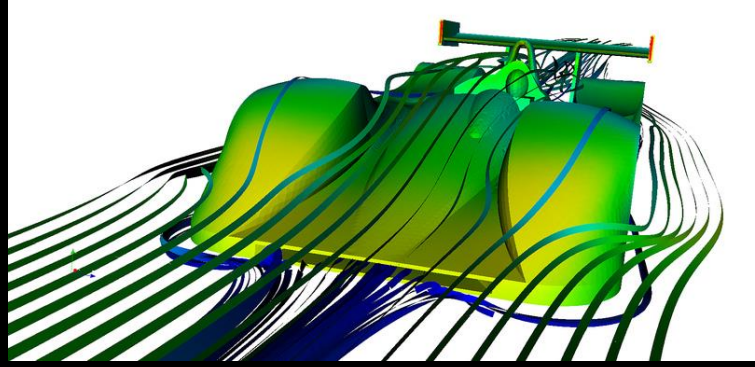
http://visit.llnl.gov/



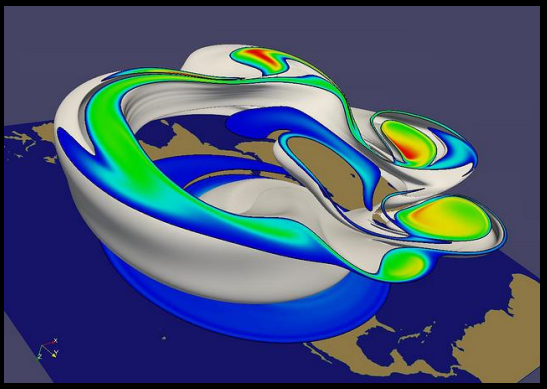
Jerry Clarke, US Army Research Laboratory

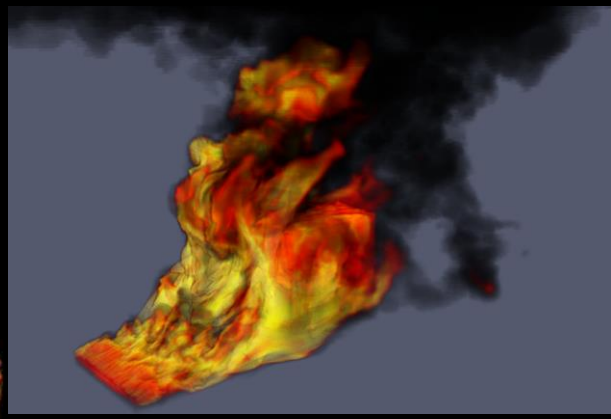


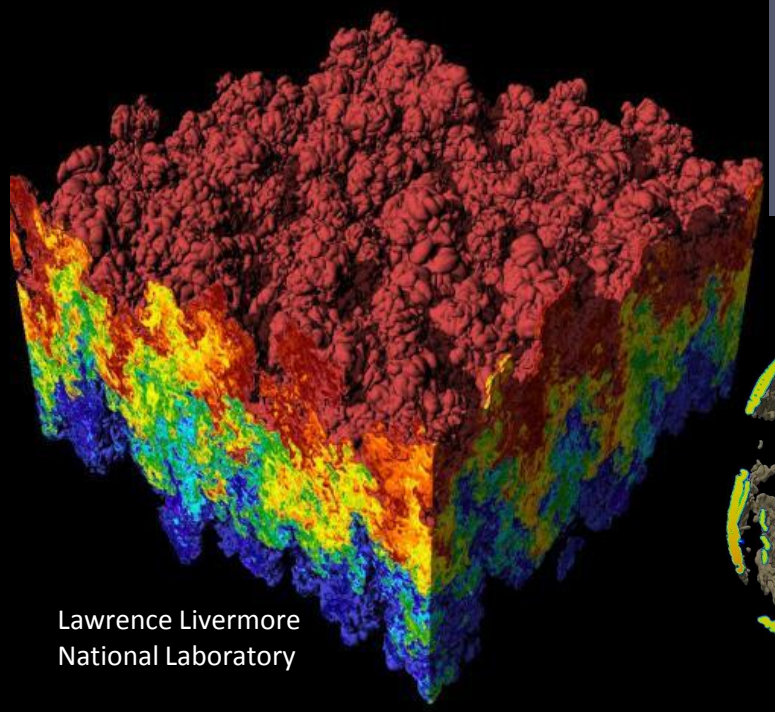
Swiss National Supercomputing Centre



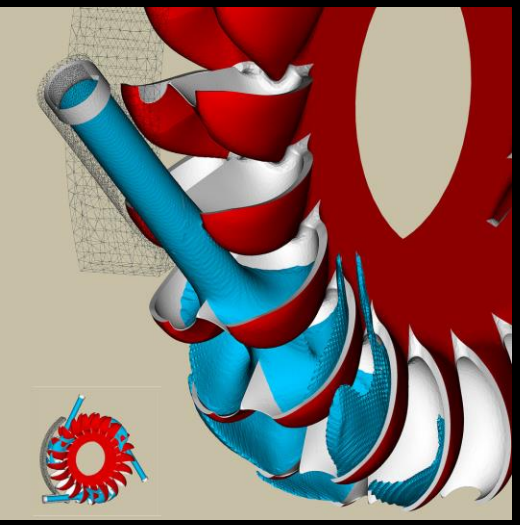
Renato N. Elias, NACAD/COPPE/UFRJ, Rio de Janerio, Brazil



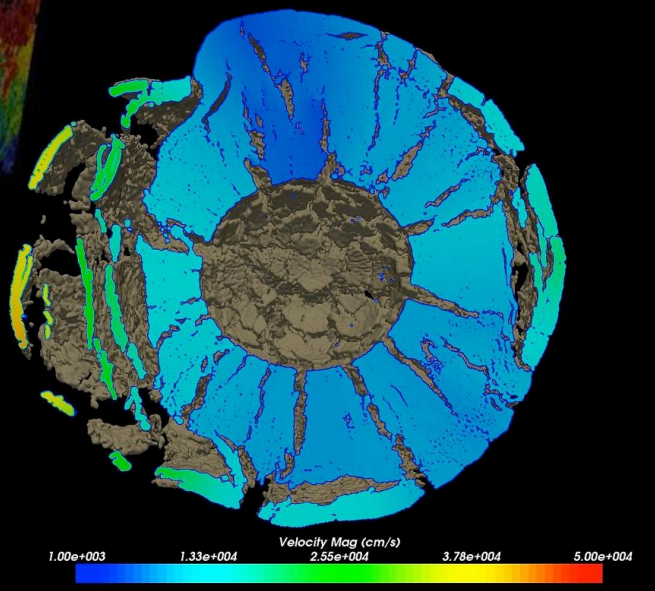




Lawrence Livermore
National Laboratory



Swiss National Supercomputing Centre



nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE

SKYFALL

The trajectory, origin and airburst behaviour of the Chelyabinsk fireball

1990's

Before consumer market impacted graphics. Single memory/multipipe machines (SGI)

2000's

- 2005: Specialized Vis cluster - Distributed memory, commodity clusters tightly coupled with compute platform.
  - Specialized HW (graphics cards, I/O, memory)
- 2010: Distributed memory capability clusters – 'Running on the platform'
  - Highly constrained memory, no specialized HW

RedRoSE/BlackRoSE

Cielo

2020's

- Constrained by power
  - Need: Coupled analysis
- Exascale breaks everything
  - New programming models
  - Billion-way parallelism

Exascale Platform

Slide of Doom

System Parameter	2011	"2018"		Factor Change
System Peak	2 PetaFLOPS	1 ExaFLOP		500
Power	6 MW	≤ 20 MW		3
System Memory	0.3 PB	32 – 64 PB		100 – 200
Total Concurrency	225K	1B × 10	1B × 100	40,000 – 400,000
Node Performance	125 GF	1 TF	10 TF	8 – 80
Node Concurrency	12	1,000	10,000	83 – 830
Network BW	1.5 KB/s	100 GB/s	1000 GB/s	66 – 660
System Size (nodes)	18,700	1,000,000	100,000	50 – 500
I/O Capacity	15 PB	300 – 1000 PB		20 – 67
I/O BW	0.2 TB/s	20 – 60 TB/s		100 – 300

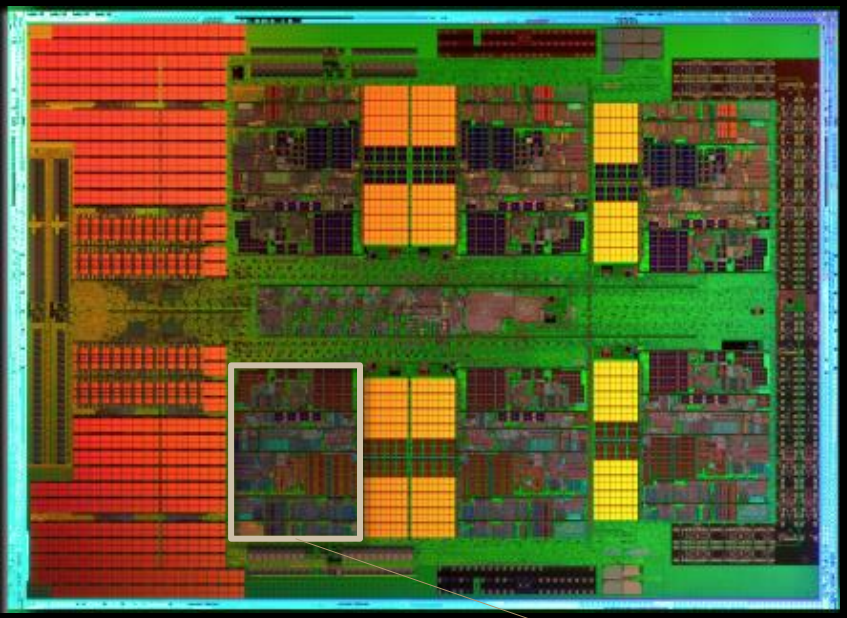
Extreme Scale is Threads, Threads, Threads!

Sandia National Laboratories

	Jaguar – XT5	Titan – XK7	Exascale*
Cores	224,256	299,008 and 18,688 gpu	1 billion
Concurrency	224,256 way	70 – 500 million way	10 – 100 billion way
Memory	300 Terabytes	700 Terabytes	128 Petabytes

- To succeed at extreme scale, you need to consider the finest possible level of concurrency
  - Expect each thread to process exactly one element
  - Disallow communication among threads

*Source: Scientific Discovery at the Exascale, Ahern, Shoshani, Ma, et al.

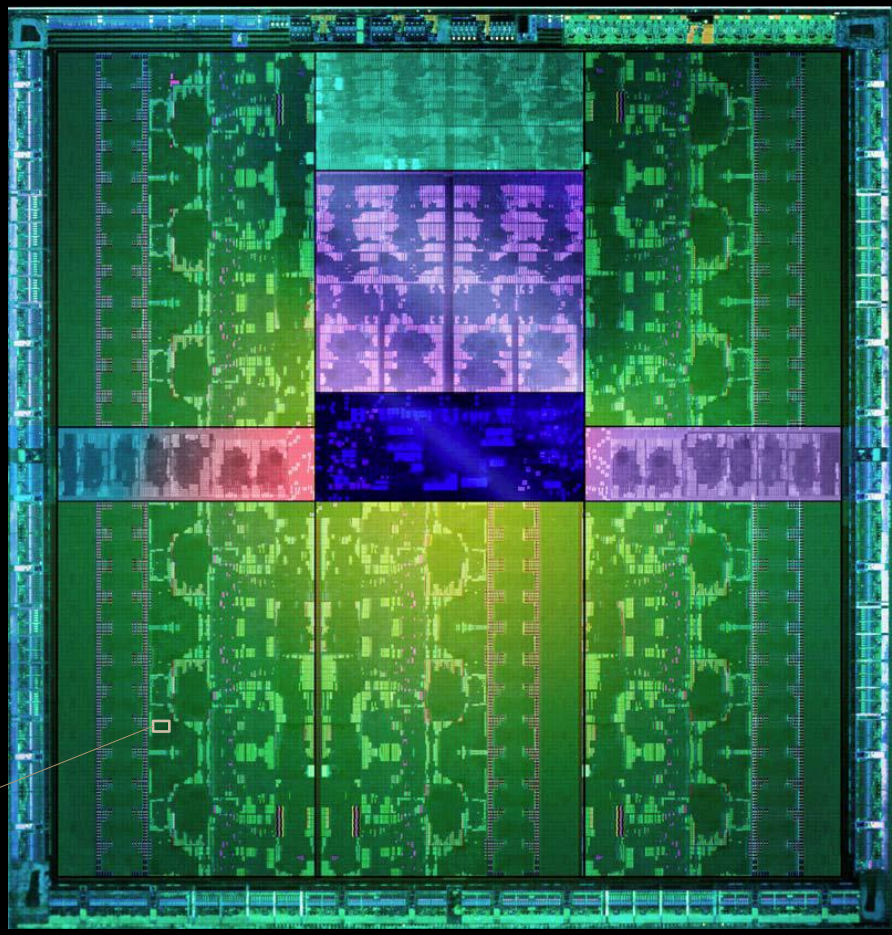


AMD x86

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

1 x86 core

1 Kepler "core"



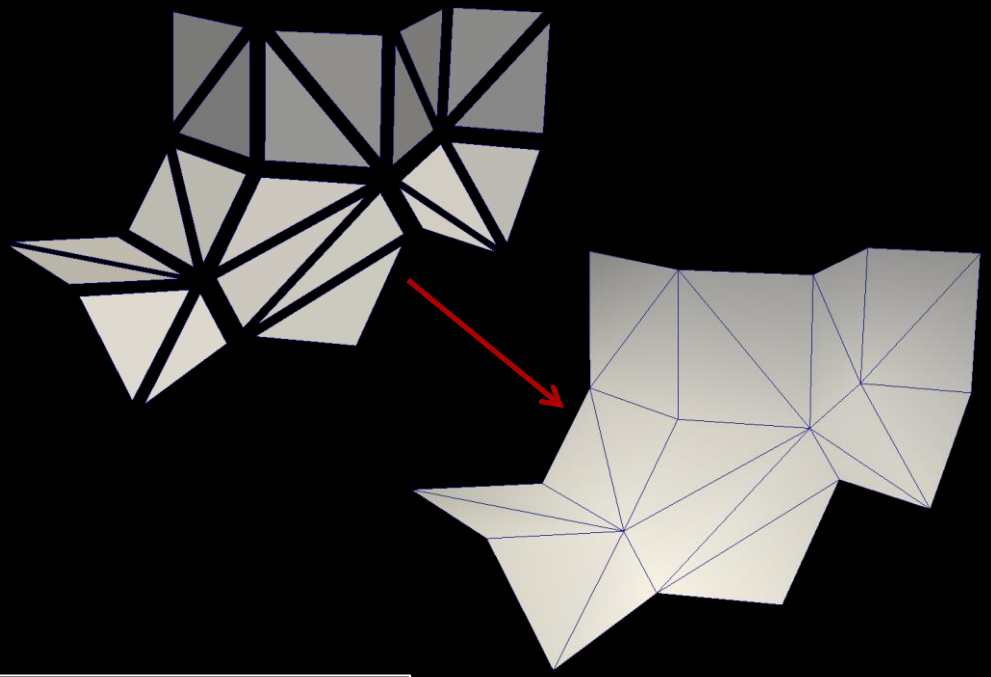
NVIDIA GPU

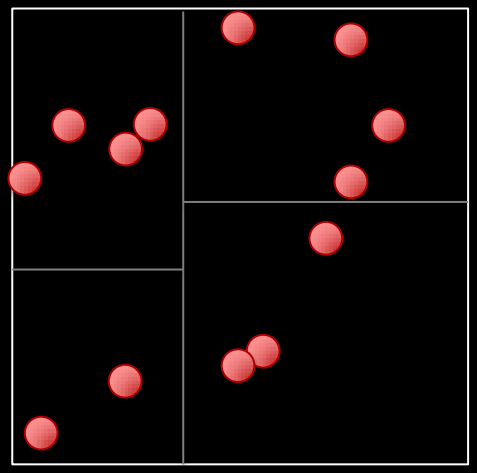
2,880 cores collected in 15 SMX
Shared PC, Cache, Mem Fetches
Reduced control logic
MPI-Only not feasible

Challenges in Massively-Threaded Vis

Sandia National Laboratories

- Connectivity-based representations (nodes, elements, fields, etc.)
- Irregular output sizes (sparse array compaction)
- Adjacency traversal (nodes to elements or elements to nodes)
- Connection finding (element soup to manifold structure)
- Search structures





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

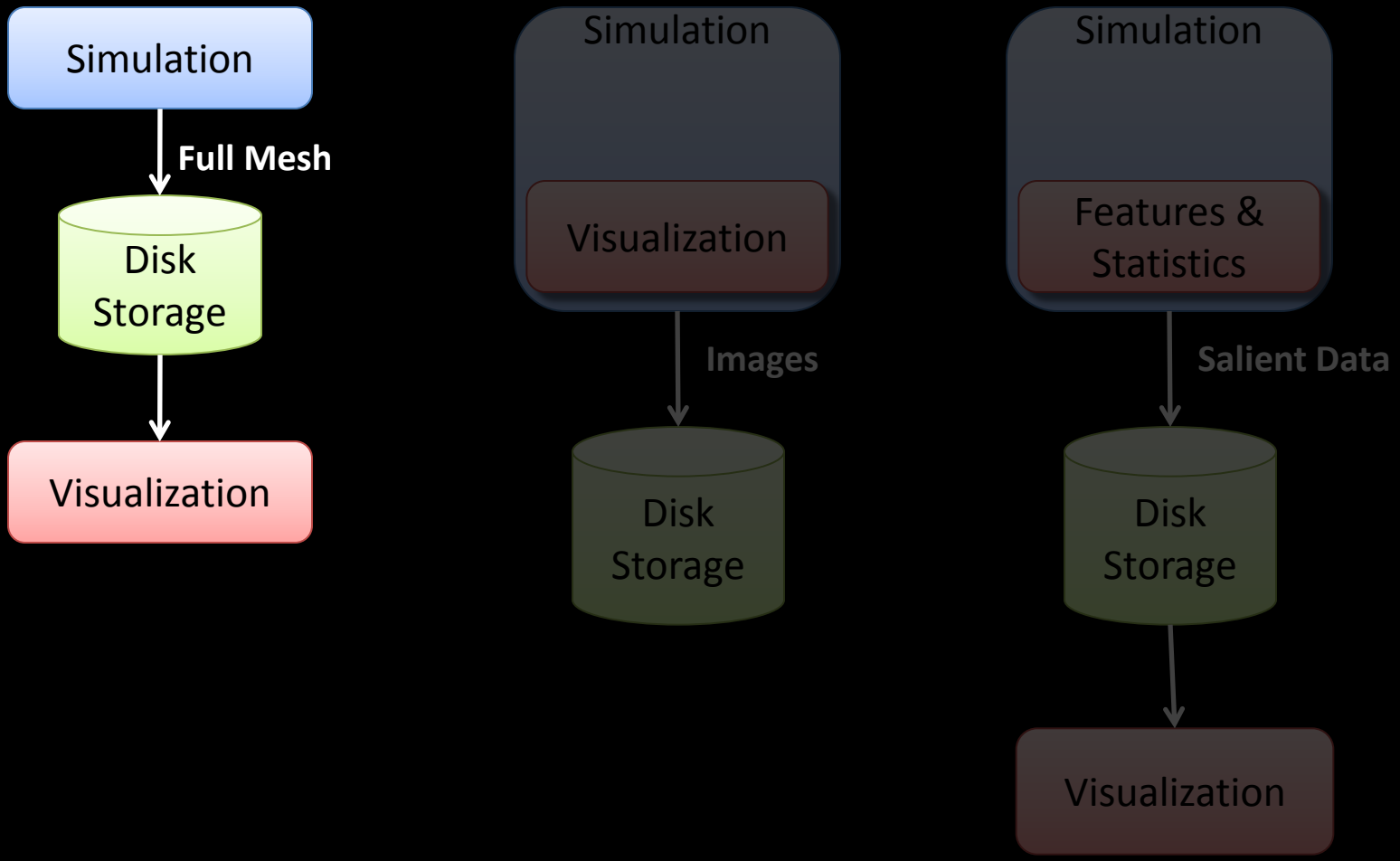
Sandia/Los Alamos National Laboratories

	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×**

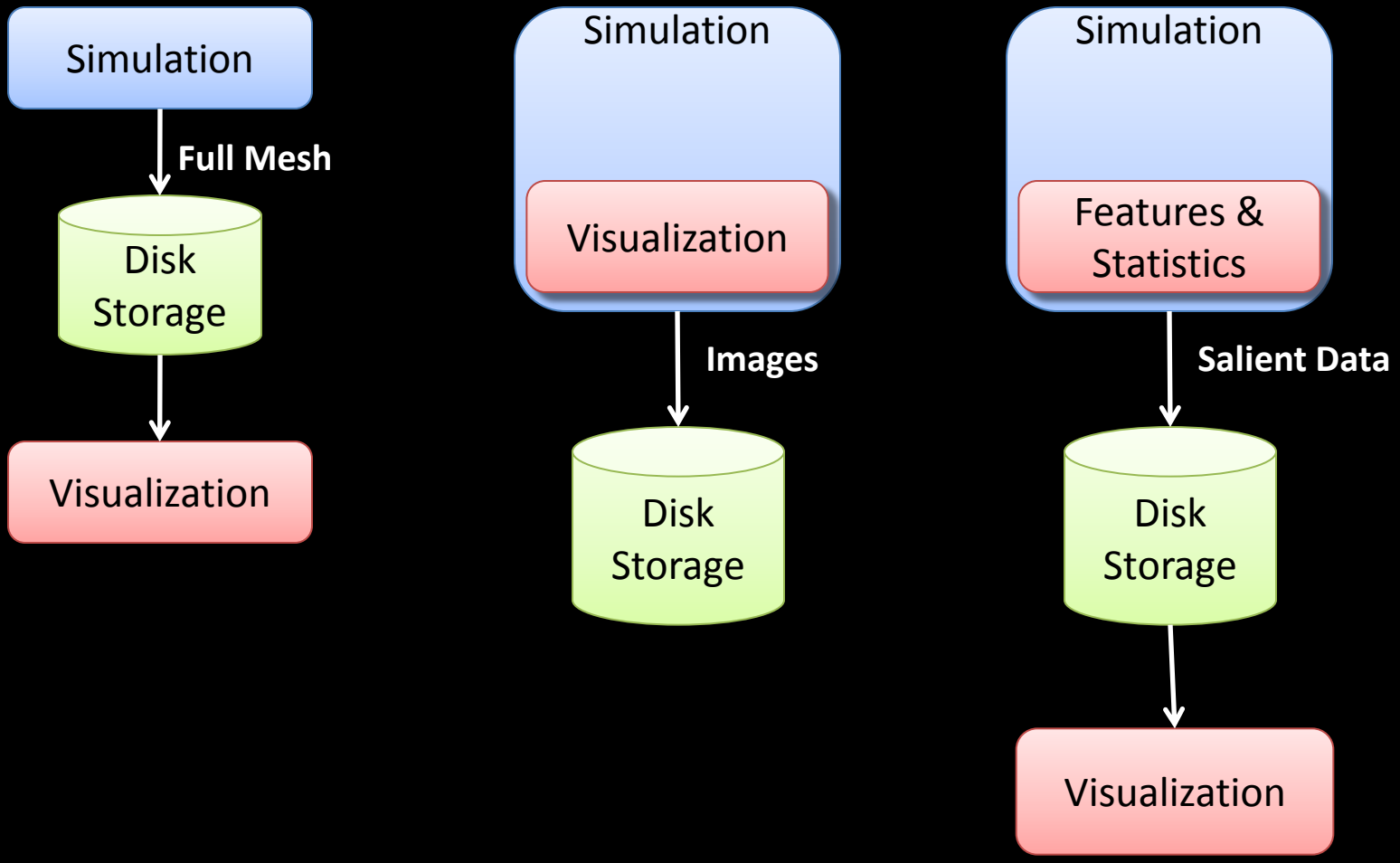
https://cfwebprod.sandia.gov/cfdocs/CCIM/docs/033768p.pdf
http://www.lanl.gov/orgs/hpc/cielo/

Diskless Visualization

Sandia National Laboratories



Diskless Visualization



The Catalyst In Situ Library

Sandia National Laboratories

Catalyst

catalyst.paraview.org

Apps | Vis | Wikis | Misc | Papers | Bookmarks | Sandia Email | Dropbox | Detexify | ExtremeScaleOct201

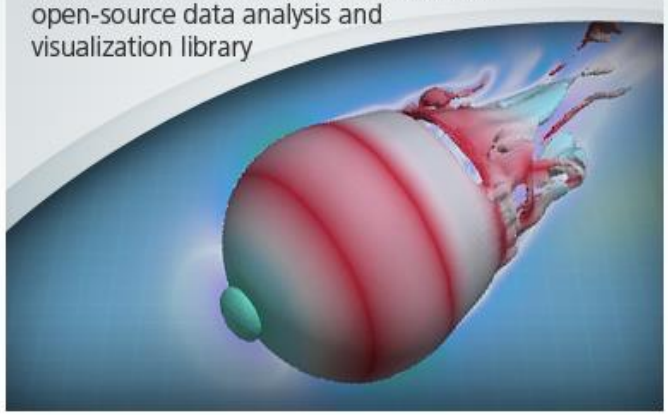
CATALYST
scalable in-situ analysis

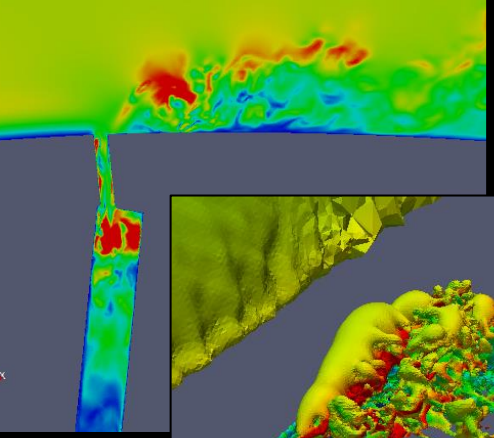
PROJECT RESOURCES HELP OPEN SOURCE

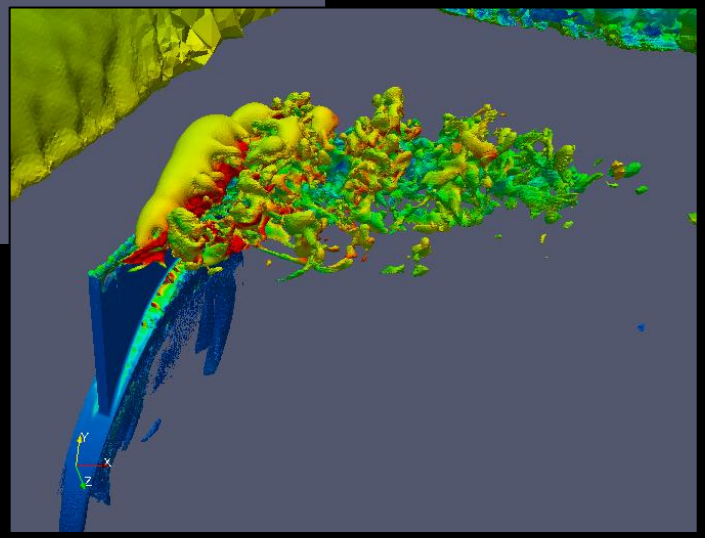
With the processing power of supercomputers rapidly growing, simulation codes continue to generate more and more data at a faster rate. While this growth enables more accurate and/or complex simulations to be run, it also makes examining the results a much more difficult task. This is mainly due to the 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.

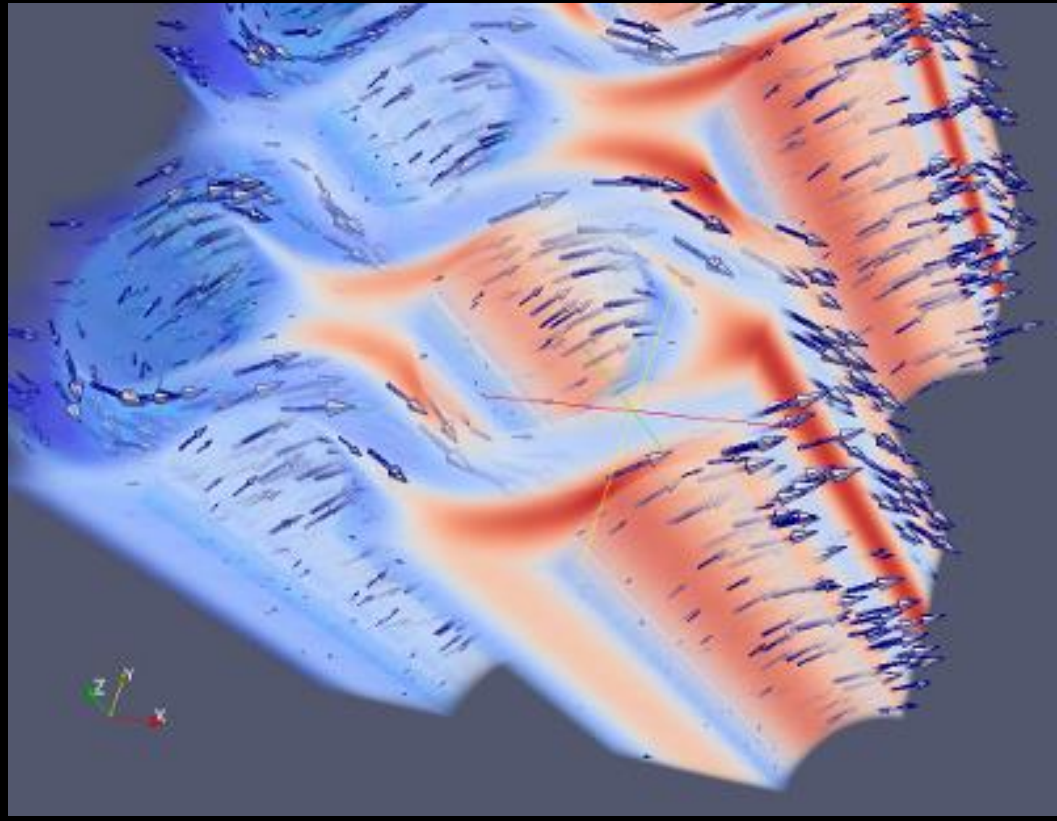
Introducing Catalyst
tightly couple simulation codes with this open-source data analysis and visualization library

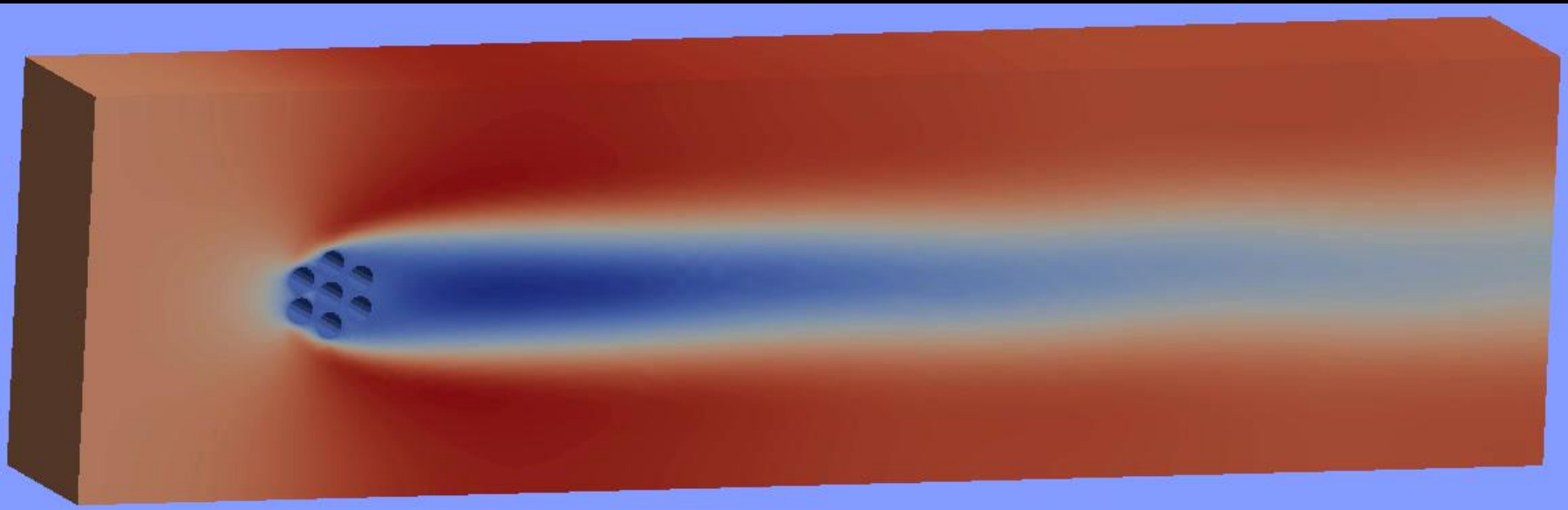


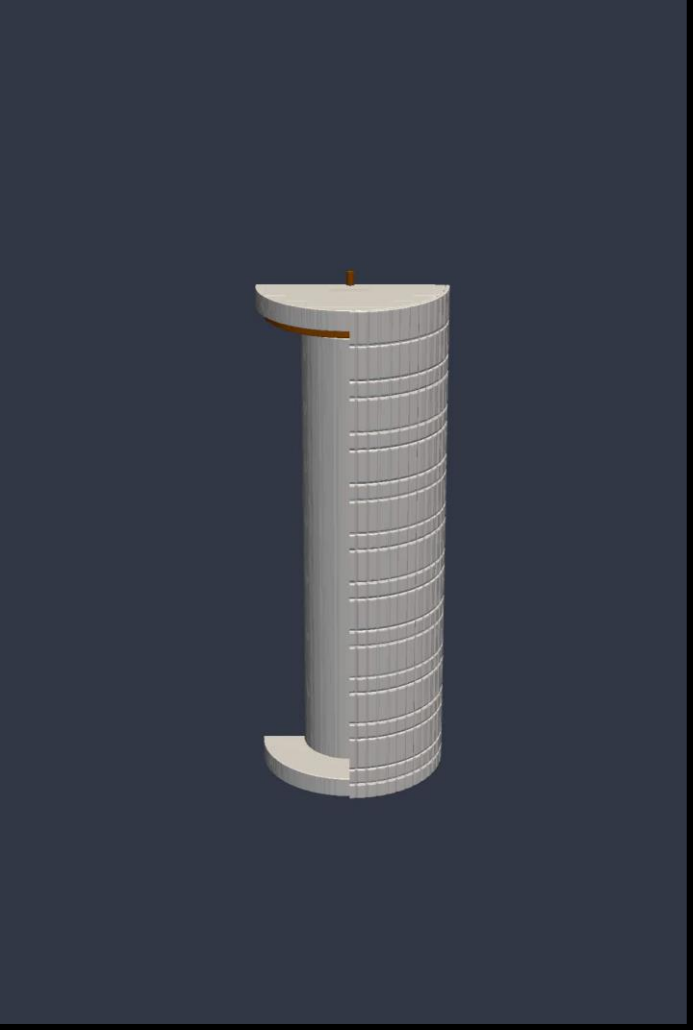


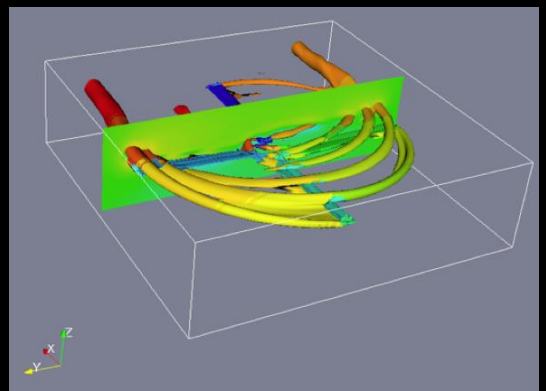


Lorendeau, Fournier, and Ribes. “In-Situ visualization in fluid mechanics using Catalyst: a case study for Code_Saturne.”

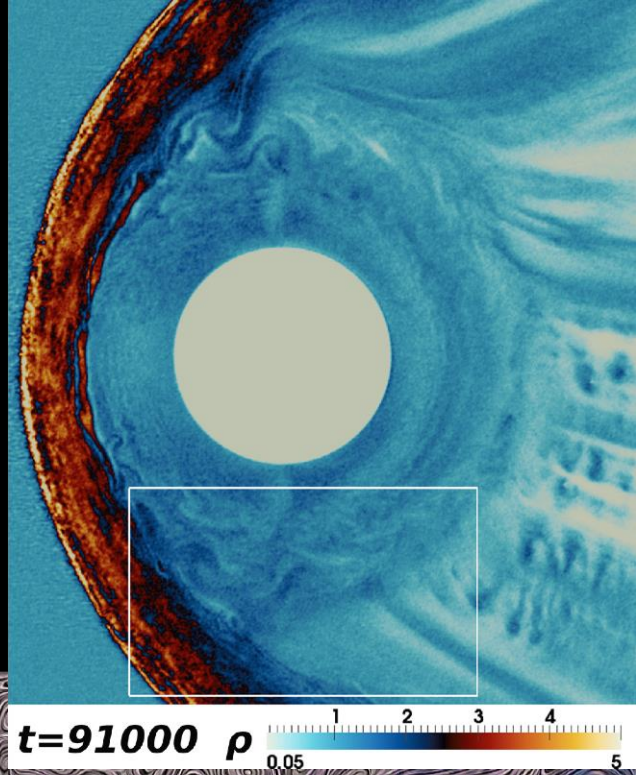


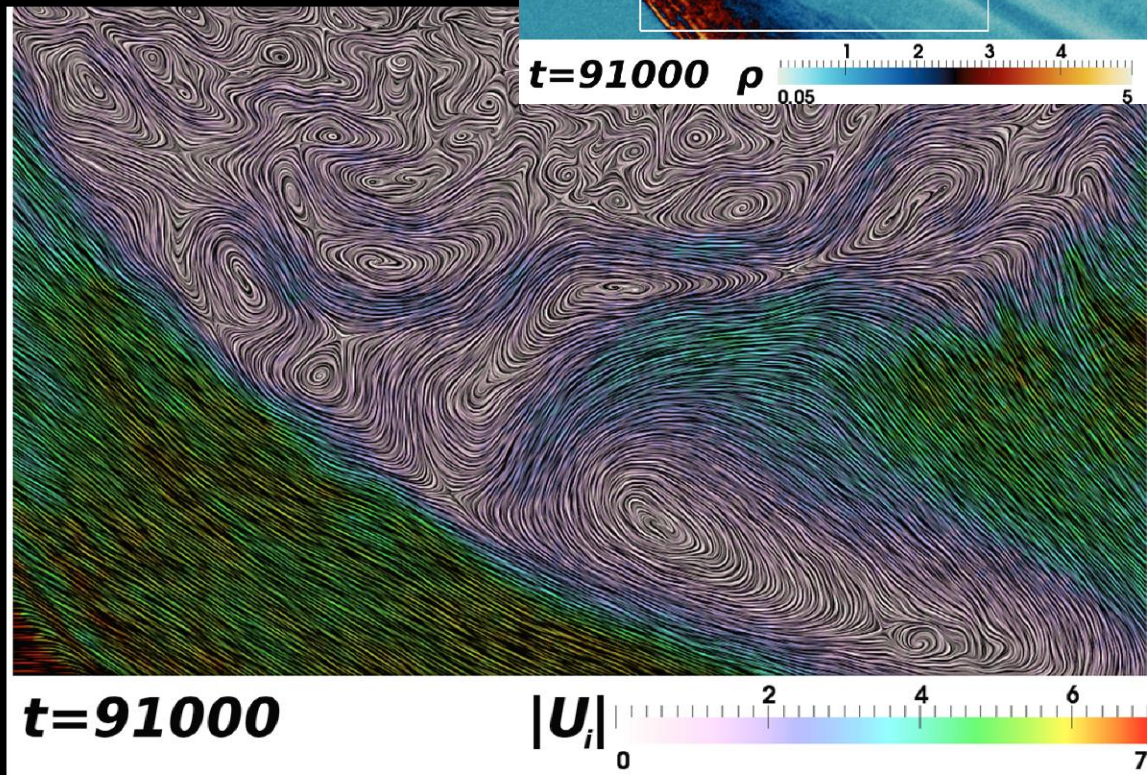






Krimabadi, O'Leary, Tatineni, Loring, Majumdar, and Geveci. "In-Situ Visualization for Global Hybrid Systems."





All I Want for a Commercialized December Holiday is My Two Front Teeth And…

Sandia National Laboratories

- An abstract I/O layer for job-to-job communication
  - In the past, leveraged the file system
  - Want the same mechanism for runtime transfers or burst buffer access
- Improved HPC Job Schedulers
  - Currently designed to optimize for homogeneous jobs
  - Want to launch jobs independently, have them communicate
  - Want to divvy up heterogeneous resources on each node
- Virtual Machines
  - What if instead of a lightweight Unix we had a Type 1 hypervisor?
  - Access to MPI fabric and accelerators (like CUDA/Xeon Phi)
  - Simplify deployment: Compile on VM on desktop, deploy to compute node
  - Simplify distribution: Make VM image, distribute to customers on platform X
  - Customize OS support: lightweight OS to full Linux distribution
  - Implement Resiliency? Capture state? Freeze and move at will?
  - Degrade hardware gracefully?