Sandia National Laboratories

SAND2012-0754P

National Nuclear Security Administration

Next-Generation Capabilities for Large-Scale Scientific Visualization

SIAM Conference on Parallel Processing for Scientific Computing

February 15, 2012

Kenneth Moreland, Nathan Fabian
Sandia National Laboratories

Berk Geveci, Utkarsh Ayachit
Kitware Inc.

James Ahrens
Los Alamos National Laboraotry

**** Release Markings ****

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.

Collaborators

- **Sandia National Laboratories**
  - Kenneth Moreland
  - Nathan Fabian
  - David Thompson
  - Ron Oldfield
- **Los Alamos National Laboratories**
  - James Ahrens
  - Jonathan Woodring
- **Oak Ridge National Laboratory**
  - Scott Klasky
  - Norbert Podhorszki
- **Argonne National Laboratory**
  - Venkatram Vishwanath
  - Mark Hereld
  - Michael E. Papka
- **Kitware, Inc.**
  - Berk Geveci
  - Utkarsh Ayachit
  - Andrew C. Bauer
  - Pat Marion
  - Sebastien Jourdain
  - David DeMarle
- **University of Colorado at Boulder**
  - Michel Rasquin
  - Kenneth E. Jansen
- **Rutgers University**
  - Ciprian Docan
  - Manish Parashar

Outline (Hide)

- Overview history: Onyx to Exascale
- ParaView
  - General capabilities
  - Remote visualization
- In situ
  - Workflow, In transit
- Exascale challenges
  - What changes
  - Dax

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

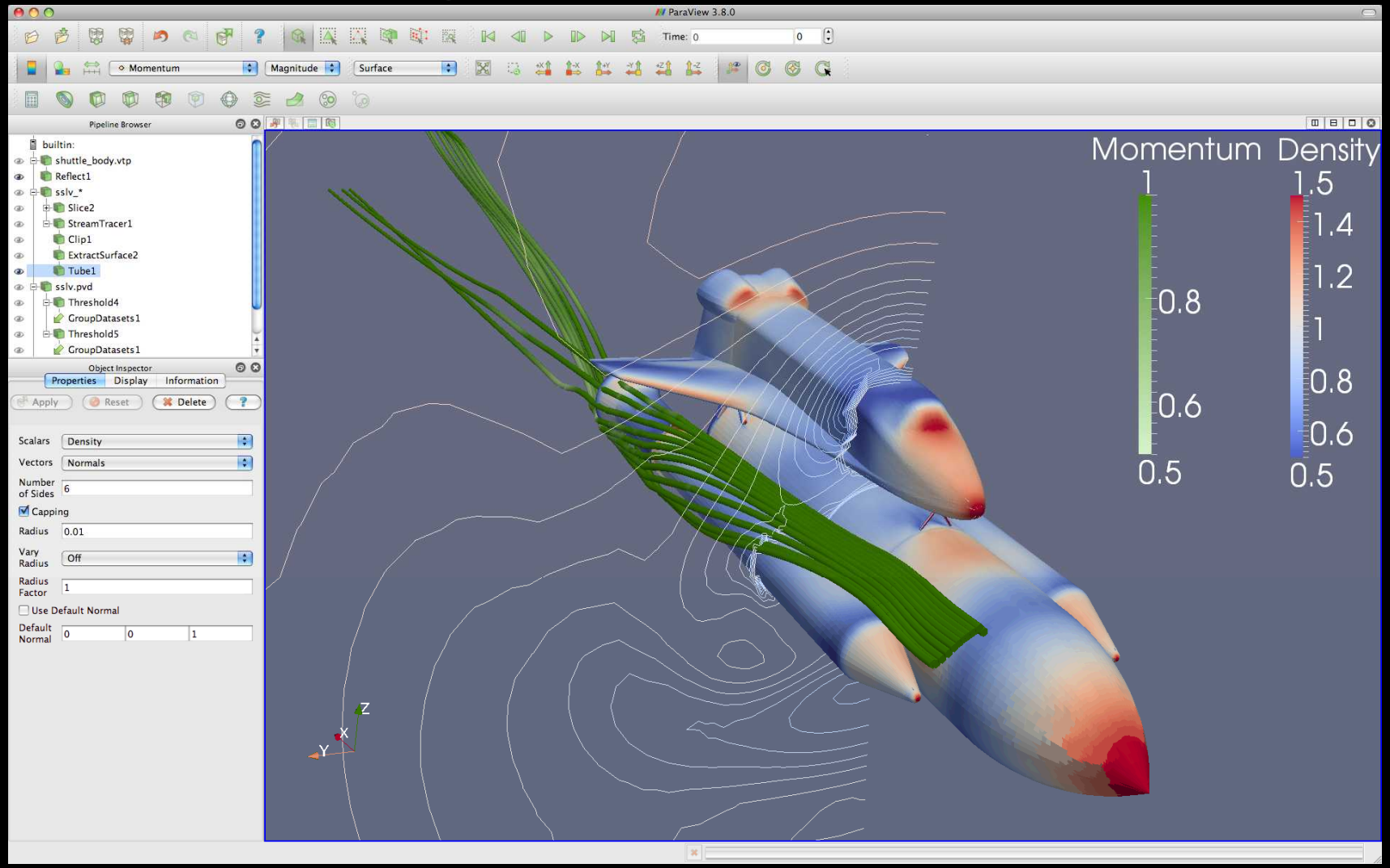
2020's

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

Exascale Platform

ParaView: an End User Tool for Large-Scale Visualization

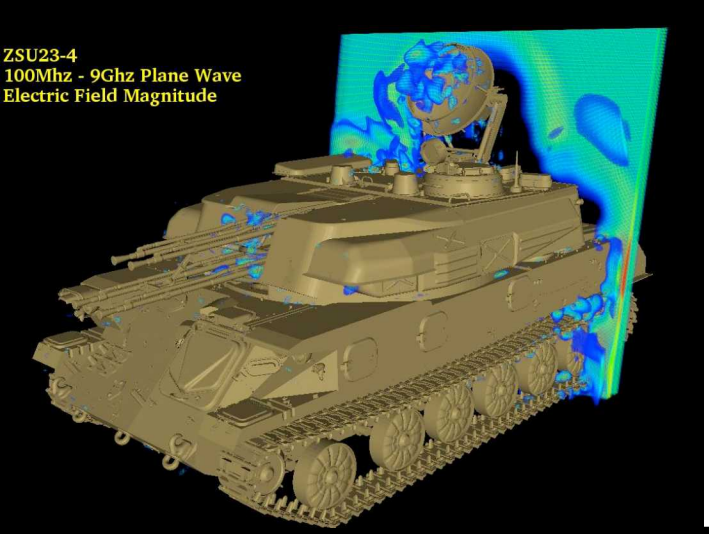
Sandia National Laboratories



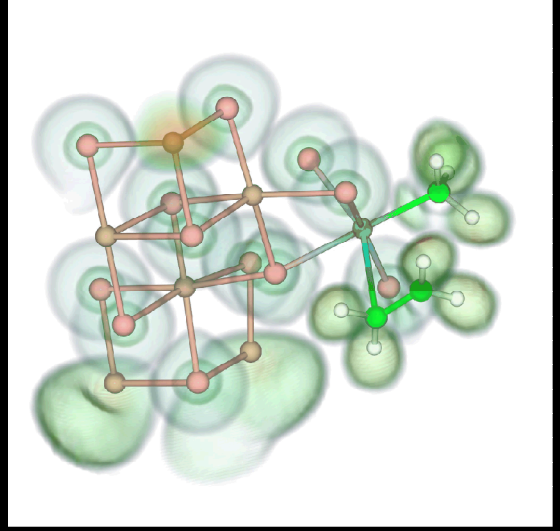
http://paraview.org

HPC wire Editors' Choice Awards 2010

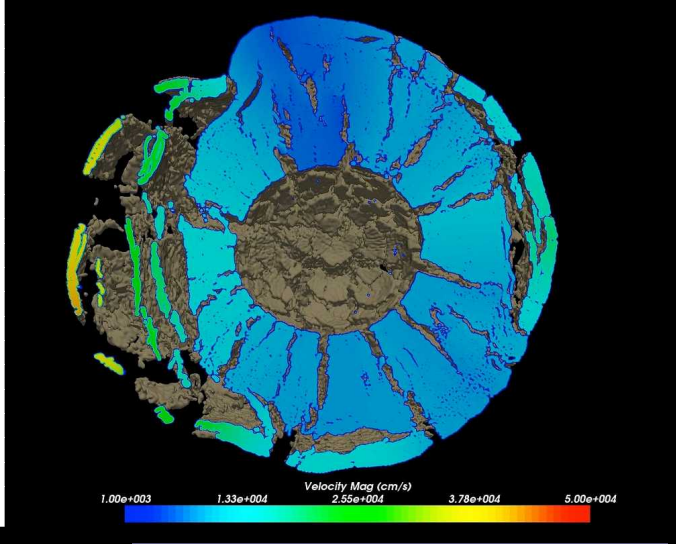
HPC wire Readers' Choice Awards 2010

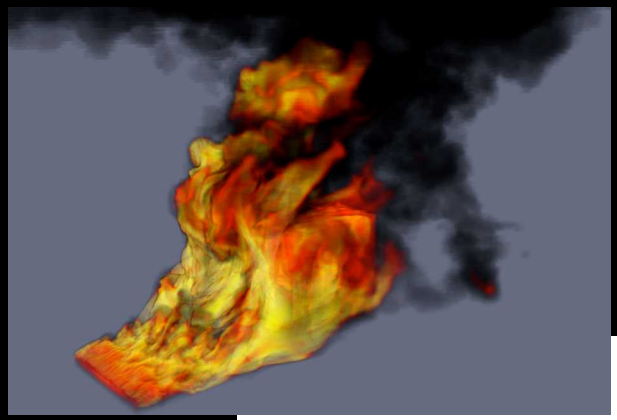


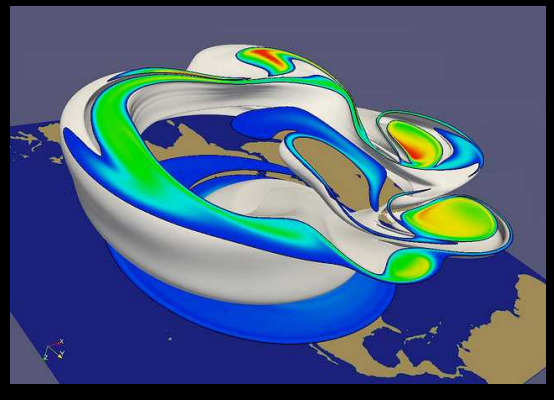
Jerry Clarke, US Army Research Laboratory

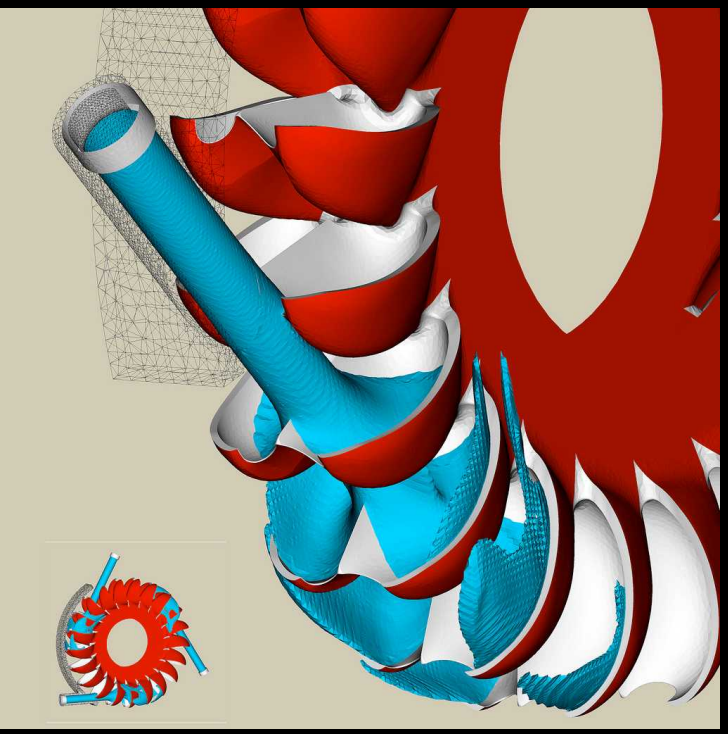


Swiss National Supercomputing Centre

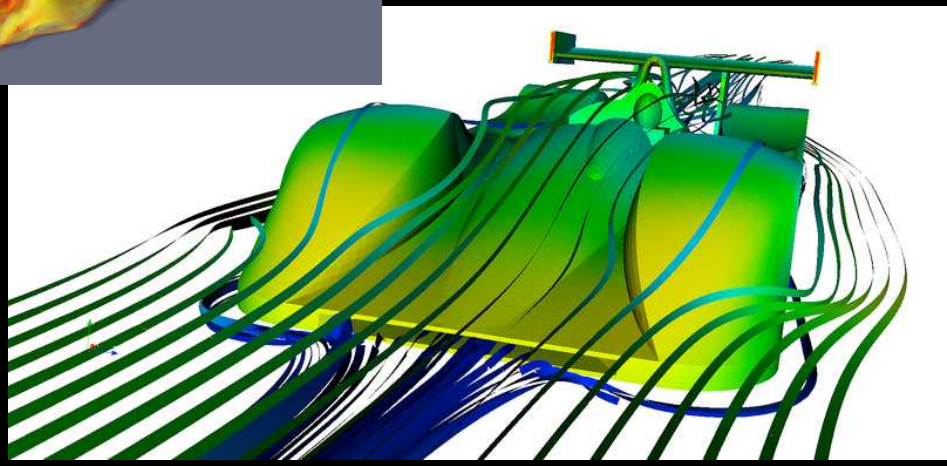








Swiss National Supercomputing Centre



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

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

	2010	"2018"	Factor Change
System Peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

DOE Exascale Initiative Roadmap, Architecture and Technology Workshop, San Diego, December, 2009.
Reported in *The Opportunities and Challenges of Exascale Computing*, Summary Report of the ASCAC Subcommittee, Fall 2010.

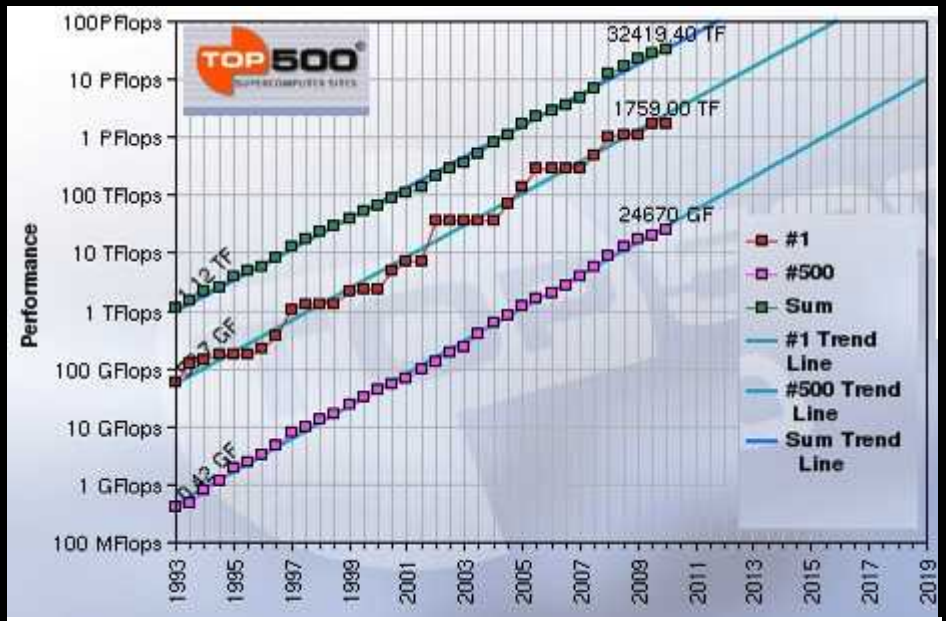
Slide of Doom

	2010	"2018"	Factor Change
System Peak	2 Pf/s	1 Ef/s	
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	

DOE Exascale Initiative Roadmap, Architecture and Technology Workshop, San Diego, December, 2009.
Reported in *The Opportunities and Challenges of Exascale Computing*, Summary Report of the ASCAC Subcommittee, Fall 2010.

Extreme scale computing

- Trends
  - More FLOPS
  - More concurrency
  - Comparatively less storage, I/O bandwidth
- ASCI purple (49 TB/140 GB/s)
- JaguarPF (300 TB/200 GB/s)
  - Most people get < 5 GB/sec at scale



Systems	2009	2011	2015	2018
System Peak Flops/s	2 Peta	20 Peta	100-200 Peta	1 Exa
System Memory	0.3 PB	1 PB	5 PB	10 PB
Node Performance	125 GF	200 GF	400 GF	1-10 TF
Node Memory BW	25 GB/s	40 GB/s	100 GB/s	200-400 GB/s
Node Concurrency	12	32	O(100)	O(1000)
Interconnect BW	1.5 GB/s	10 GB/s	25 GB/s	50 GB/s
System Size (Nodes)	18,700	100,000	500,000	O(Million)
Total Concurrency	225,000	3 Million	50 Million	O(Billion)
Storage	15 PB	30 PB	150 PB	300 PB
I/O	0.2 TB/s	2 TB/s	10 TB/s	20 TB/s
MTTI	Days	Days	Days	O(1Day)
Power	6 MW	~10 MW	~10 MW	~20 MW

ILLUSTRATION: A. TOMEY

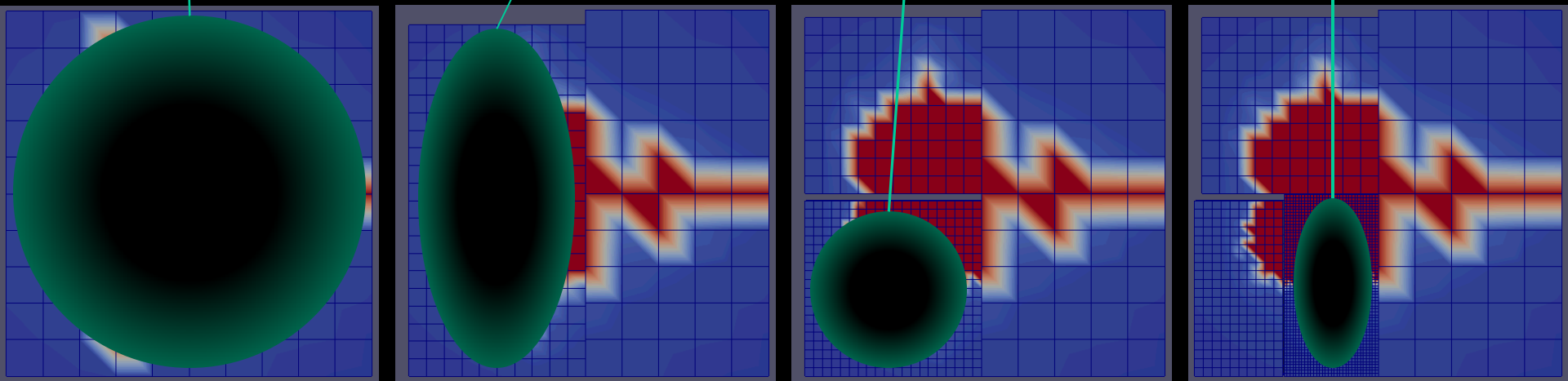
From J. Dongarra, "Impact of Architecture and Technology for Extreme Scale on Software and Algorithm Design," Cross-cutting Technologies for Computing at the Exascale, February 2-5, 2010.

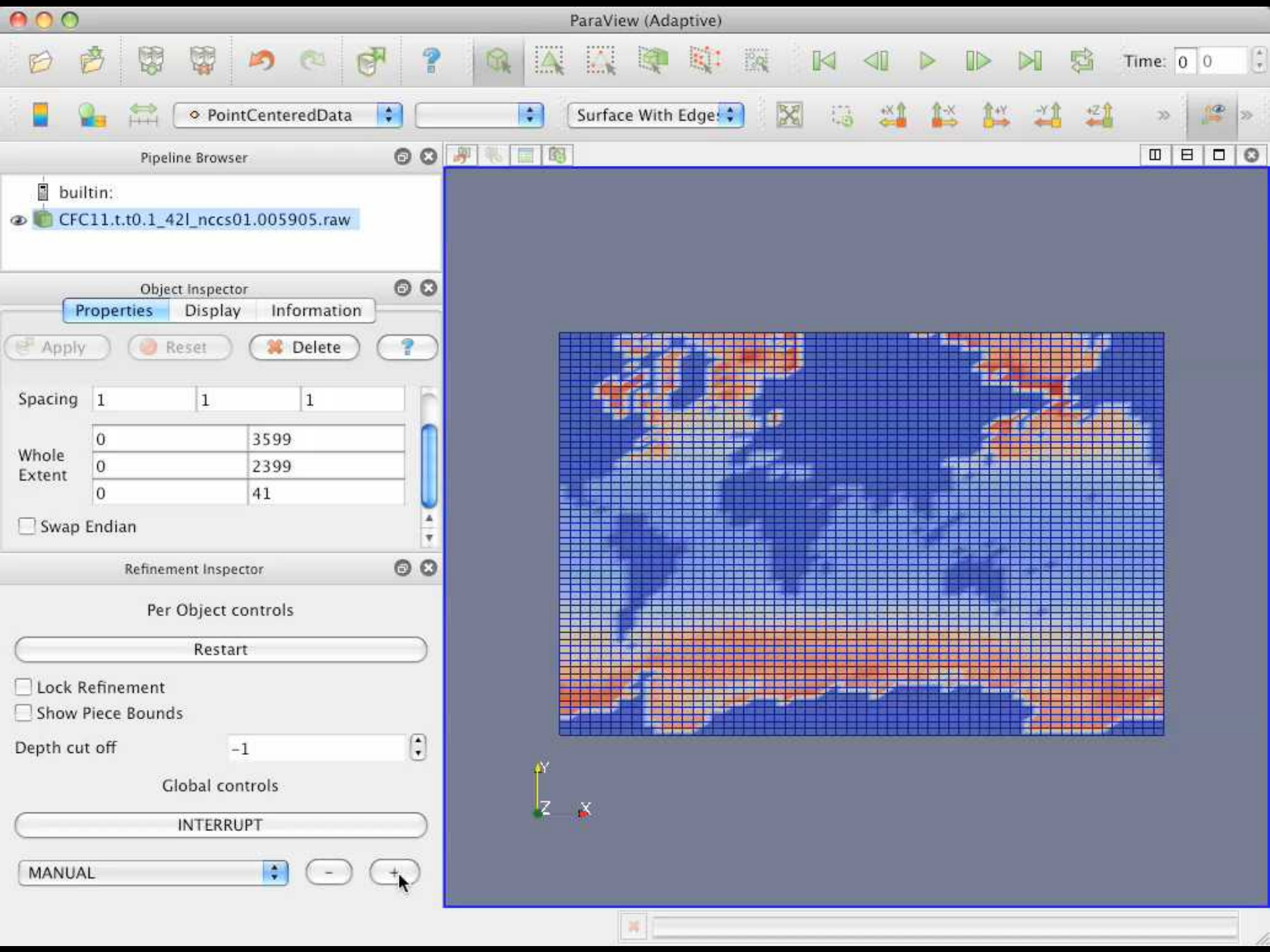
Prioritized Adaptive Streaming

Sandia National Laboratories

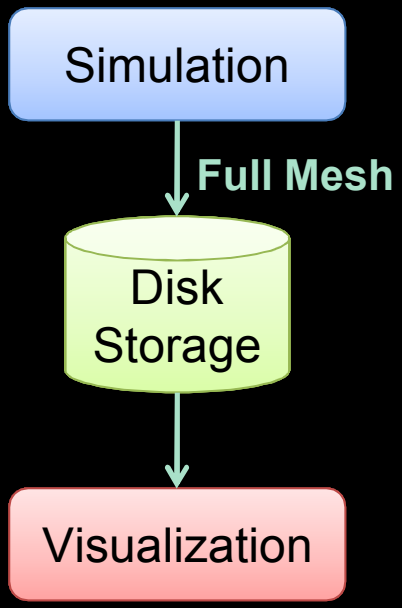
Fine

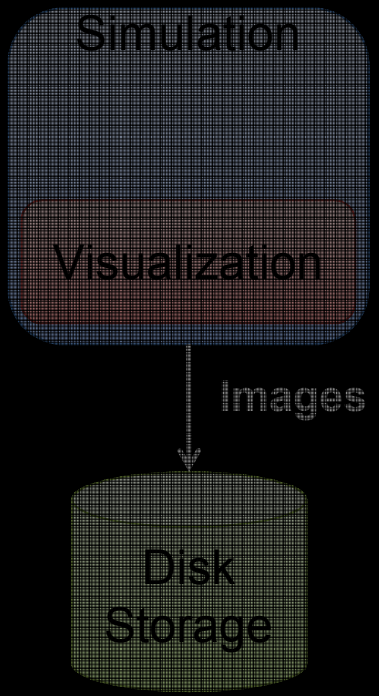
Coarse

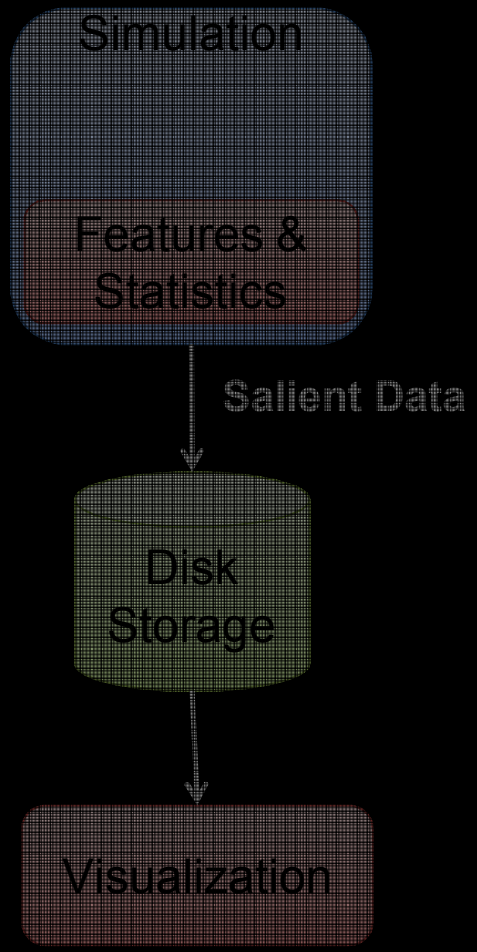



ParaView (Adaptive)
Time: 0 0
PointCenteredData
Surface With Edges
Pipeline Browser
builtin:
CFC11.t.t0.1_42l_nccs01.005905.raw
Object Inspector
Properties
Display
Information
Apply
Reset
Delete
Spacing 1 1 1
Whole Extent 0 3599 0 2399 0 41
Swap Endian
Refinement Inspector
Per Object controls
Restart
Lock Refinement
Show Piece Bounds
Depth cut off -1
Global controls
INTERRUPT
MANUAL
-
+
Y
Z
X

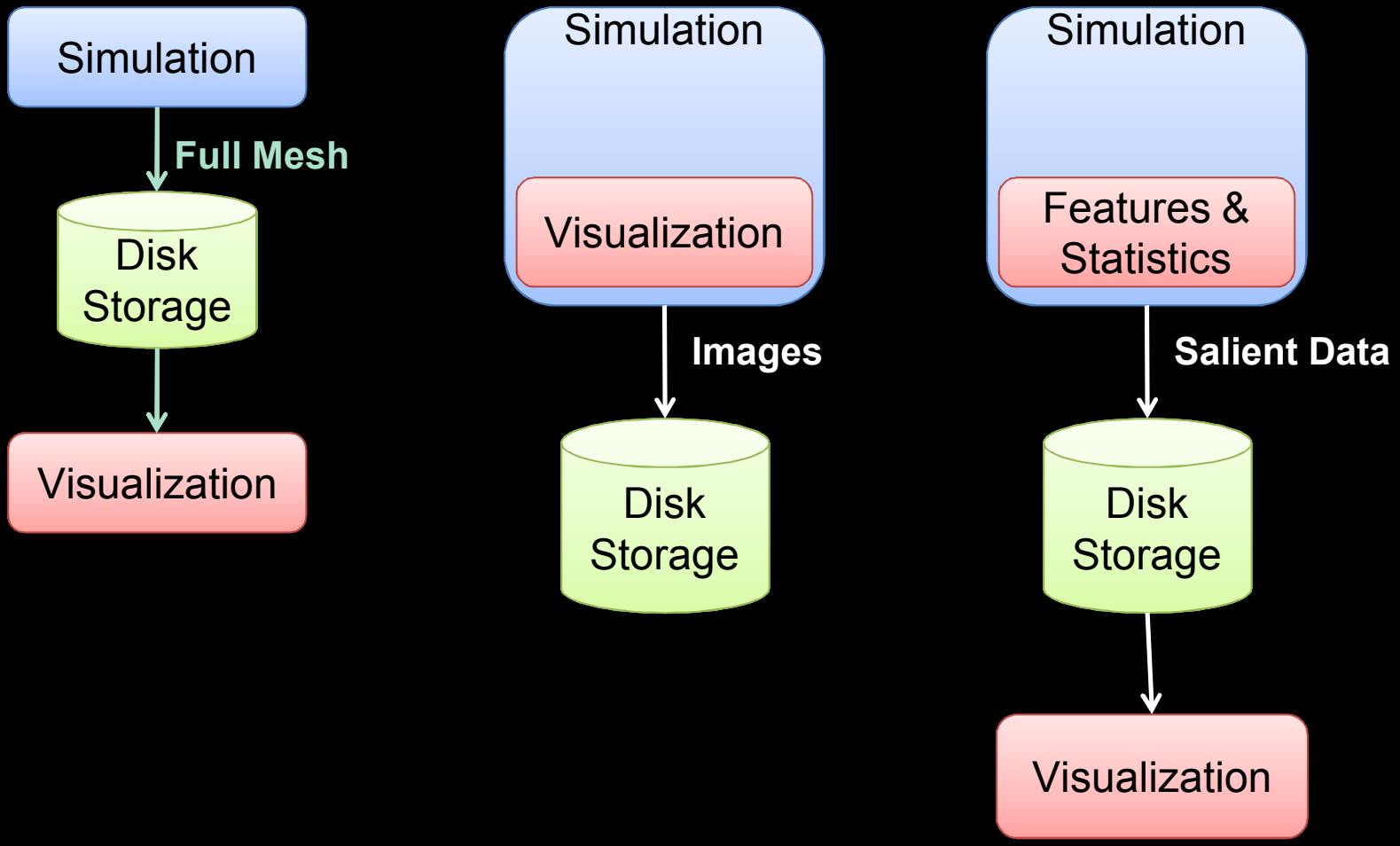
Diskless Visualization

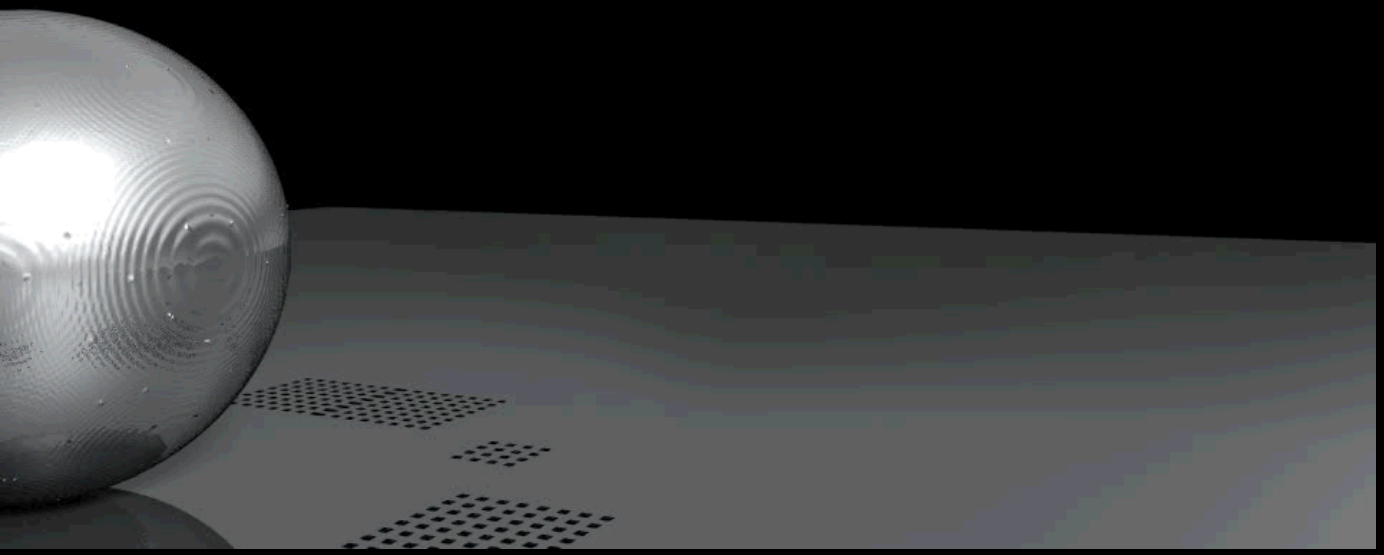




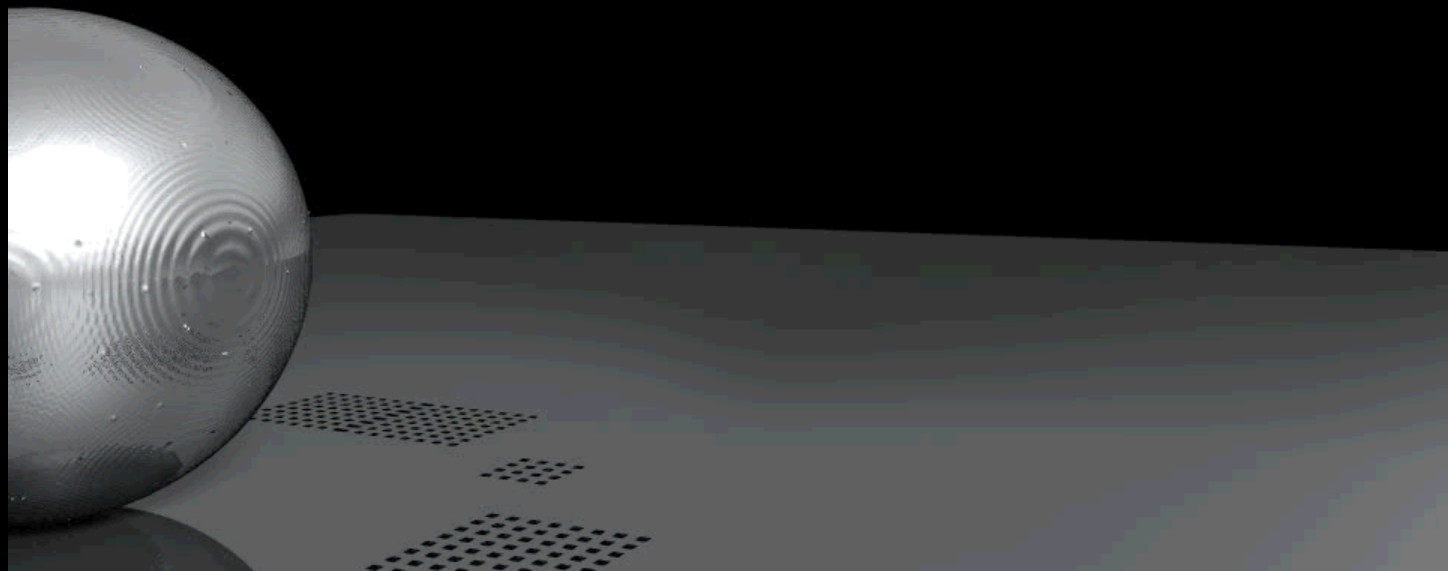


Diskless Visualization

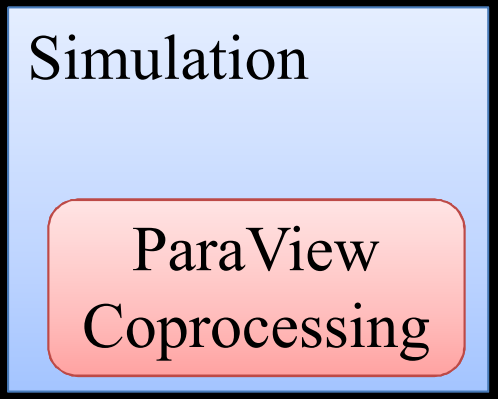


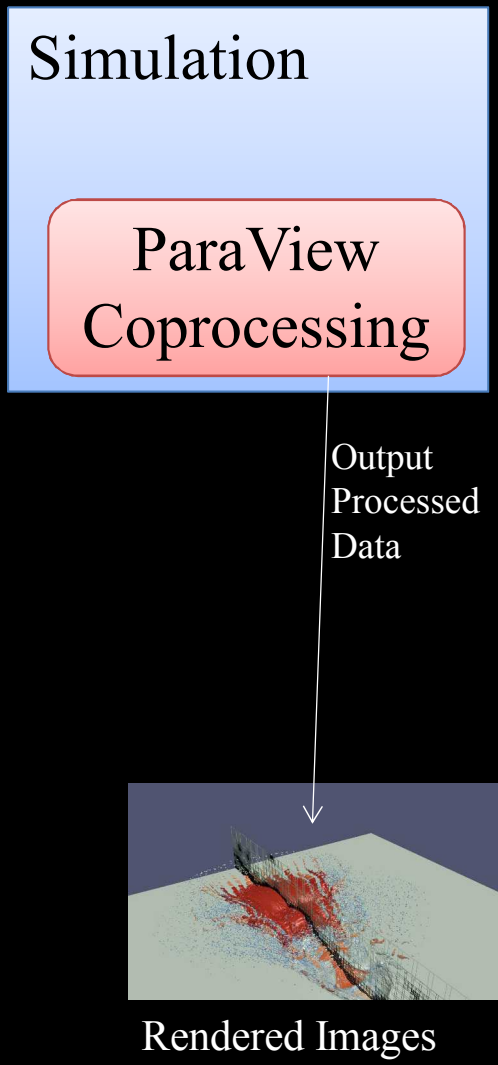


1 mesh per 100 steps
~1% write time



1 image per 1 step
~0.1% write time

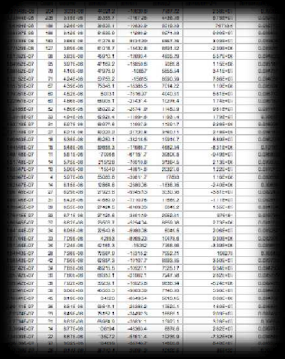

Simulation
ParaView
Coprocessing


Simulation
ParaView
Coprocessing
Output
Processed
Data
Rendered Images

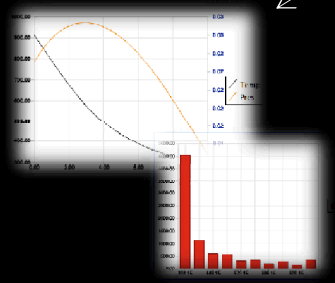
Simulation

ParaView Coprocessing

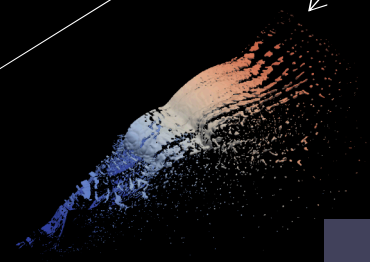
Output Processed Data



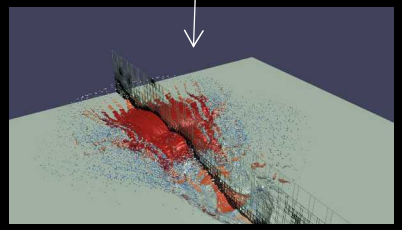
Statistics



Line Series



Polygonal Surfaces Field Data



Rendered Images

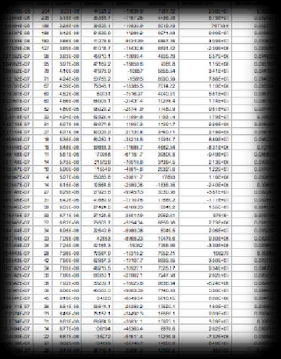
```
# Create the reader and set the filename.
reader = servermanager.sources.Reader(FileNames=path)
view = servermanager.CreateRenderView()
repr = servermanager.CreateRepresentation(reader, view)
reader.UpdatePipeline()
dataInfo = reader.GetDataInformation()
pDinfo = dataInfo.GetPointDataInformation()
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.

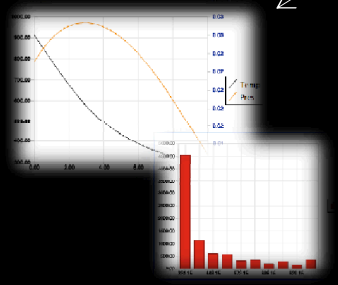
Simulation

ParaView Coprocessing

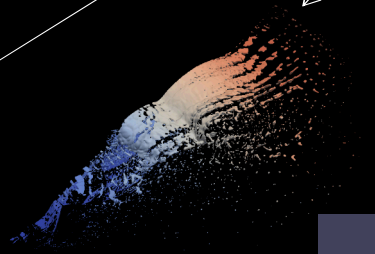
Output Processed Data



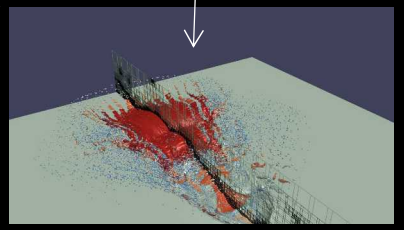
Statistics



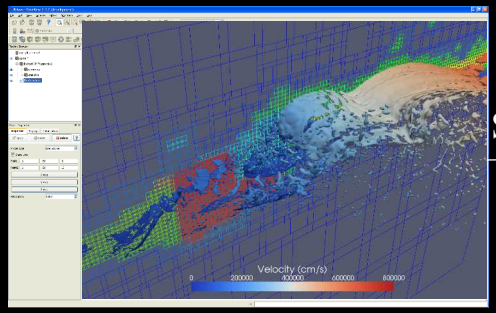
Line Series



Polygonal Surfaces Field Data



Rendered Images



Script Export

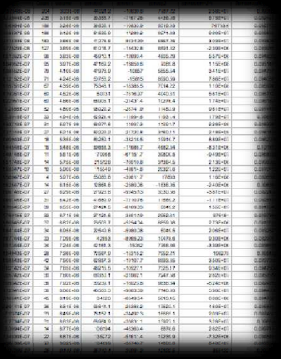
```
# Create the reader and set the filename.
reader = servermanager.sources.Reader(FileNames=path)
view = servermanager.CreateRenderView()
repr = servermanager.CreateRepresentation(reader, view)
reader.UpdatePipeline()
dataInfo = reader.GetDataInformation()
pDinfo = dataInfo.GetPointDataInformation()
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.

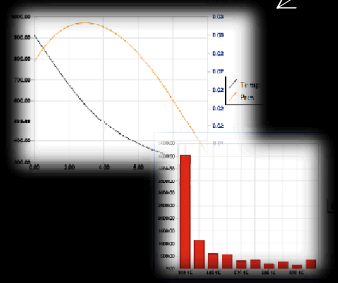
Simulation

ParaView Coprocessing

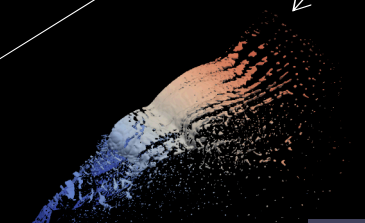
Output Processed Data



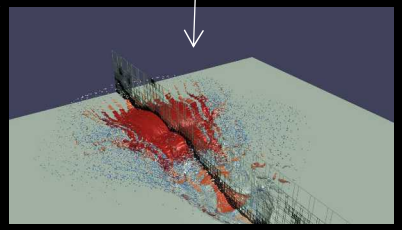
Statistics



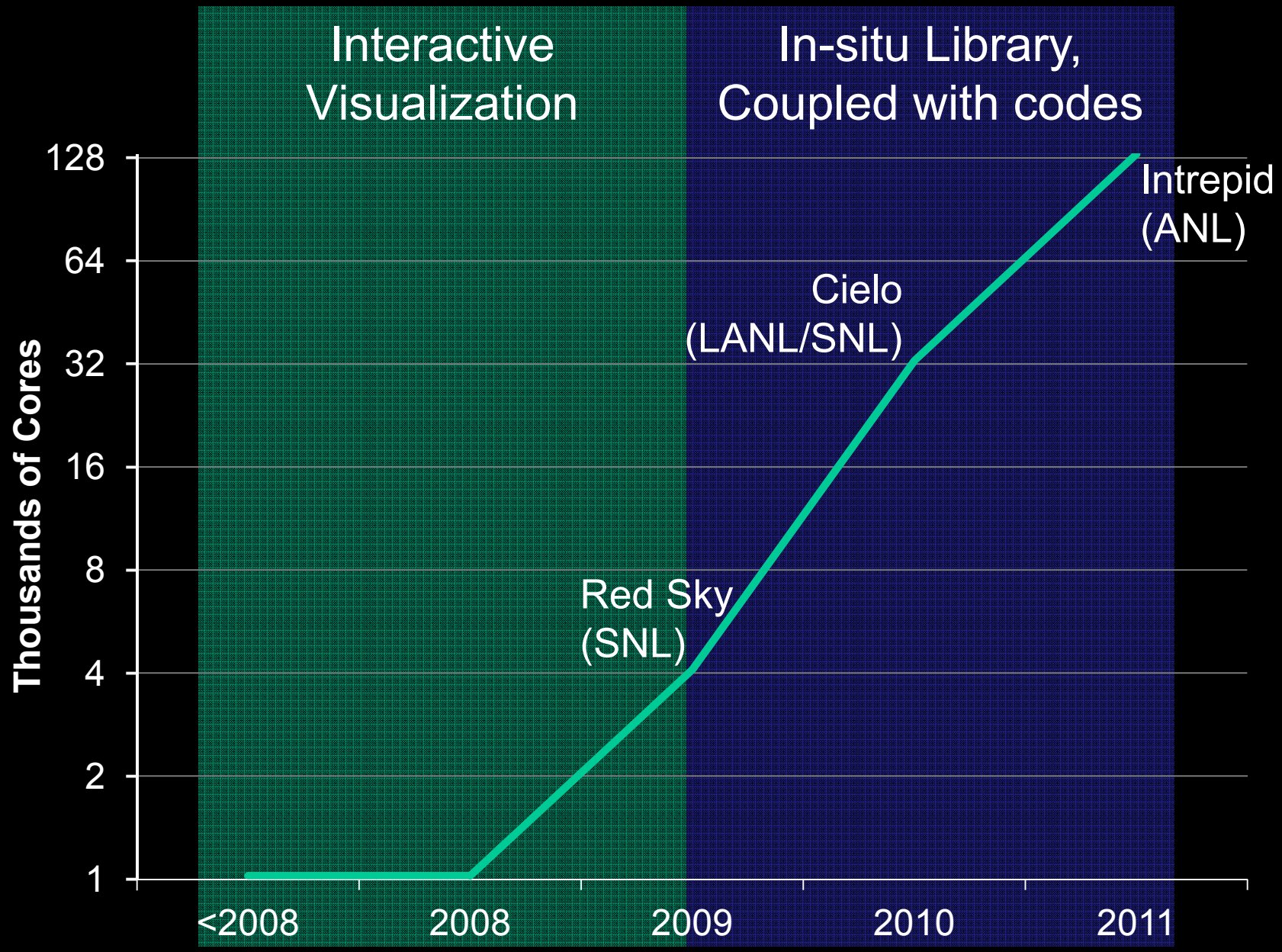
Line Series



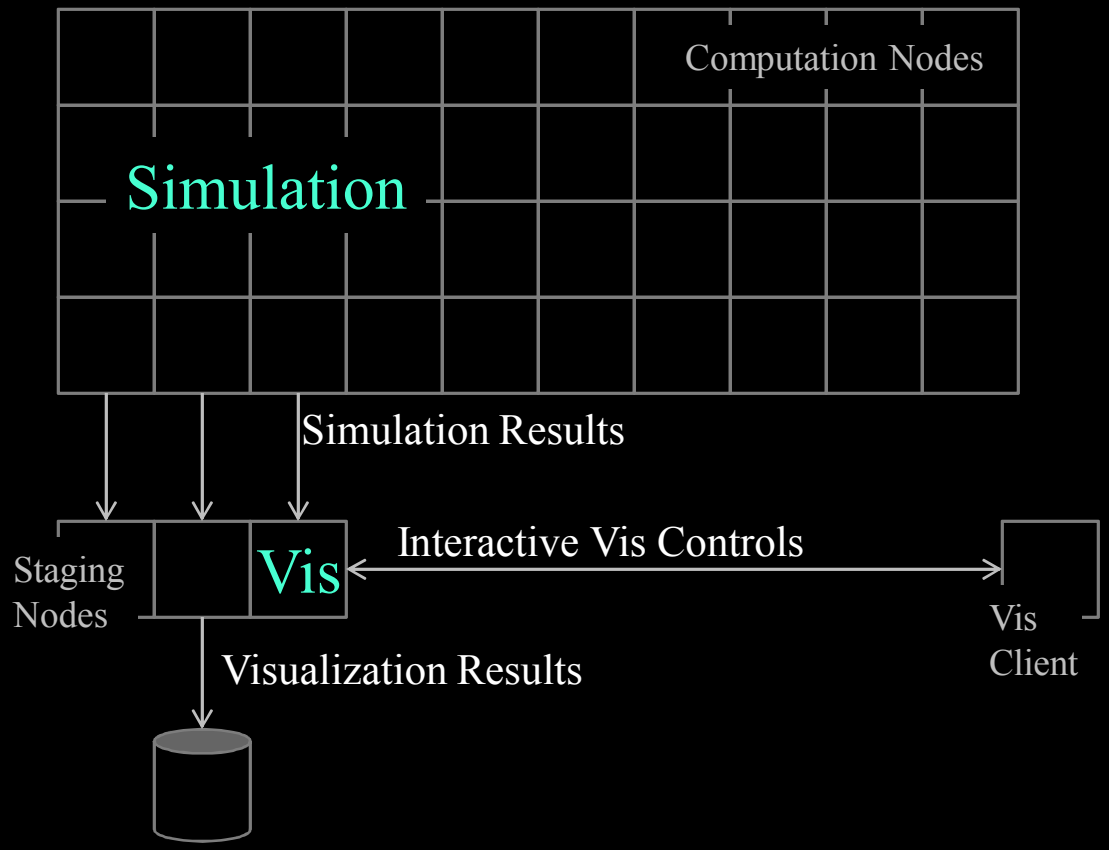
Polygonal Surfaces Field Data



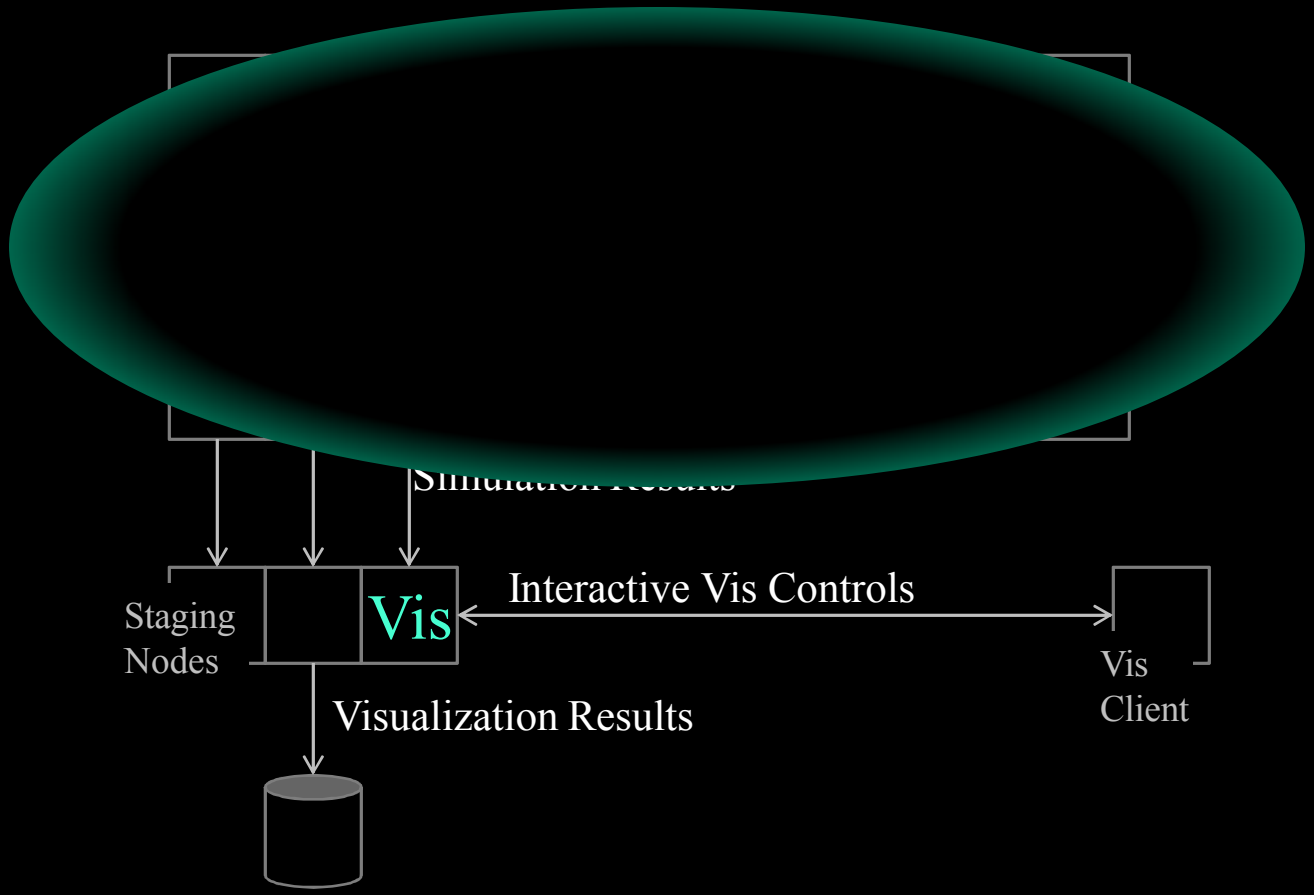
Rendered Images


Interactive Visualization
In-situ Library, Coupled with codes
Thousands of Cores
128
64
32
16
8
4
2
1
<2008
2008
2009
2010
2011
Red Sky (SNL)
Cielo (LANL/SNL)
Intrepid (ANL)

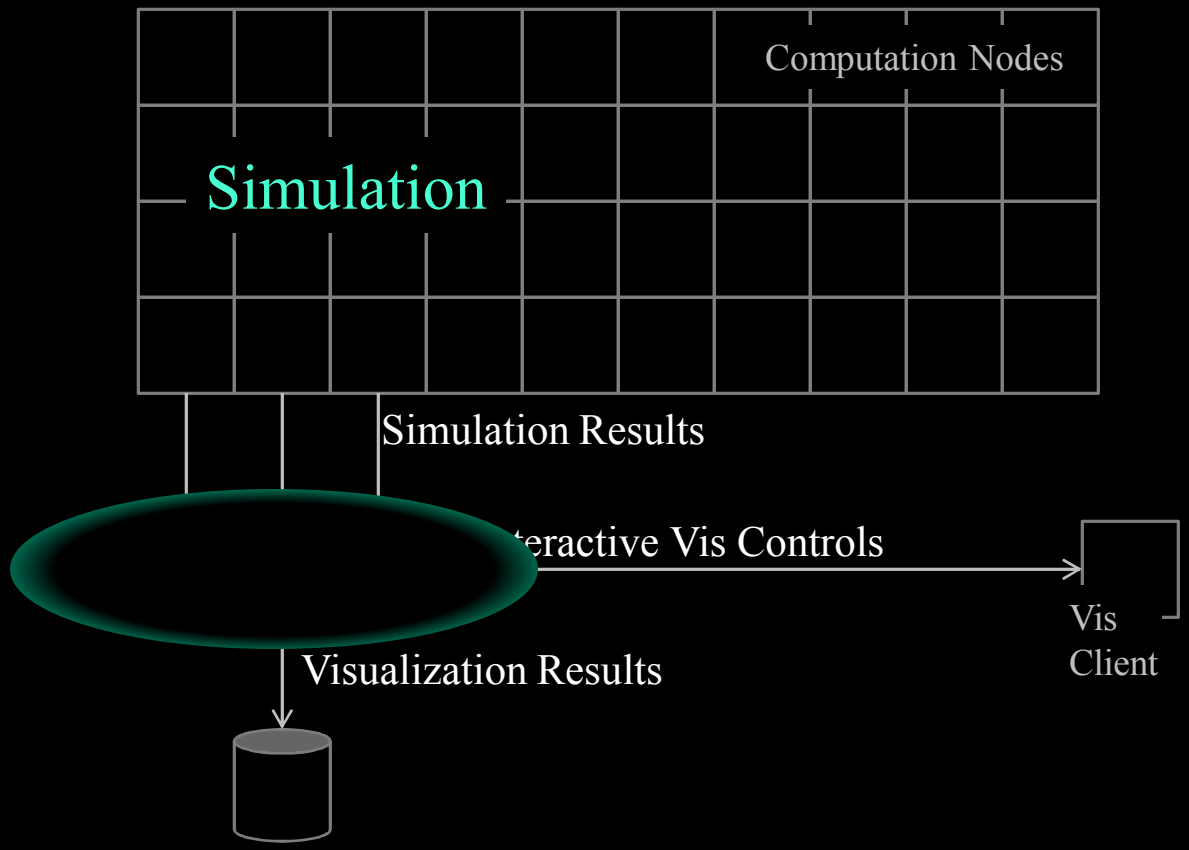
In Transit Visualization



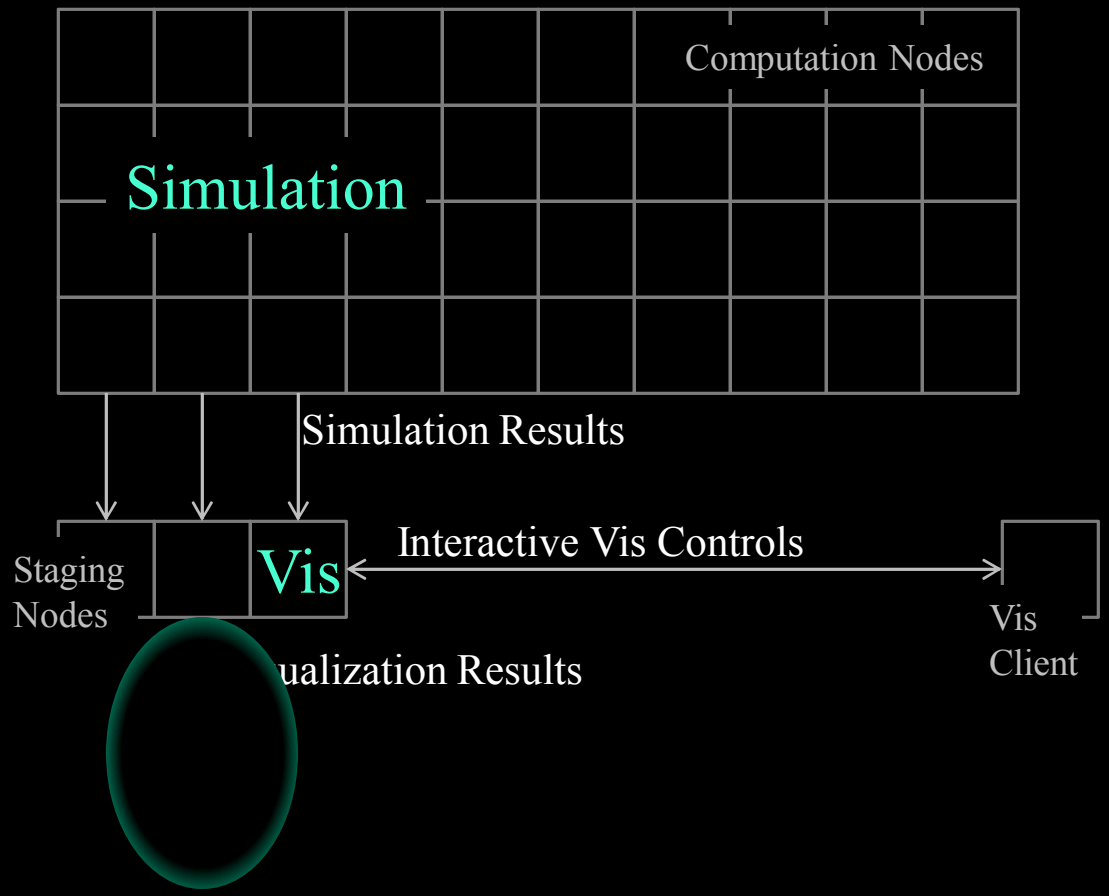
In Transit Visualization



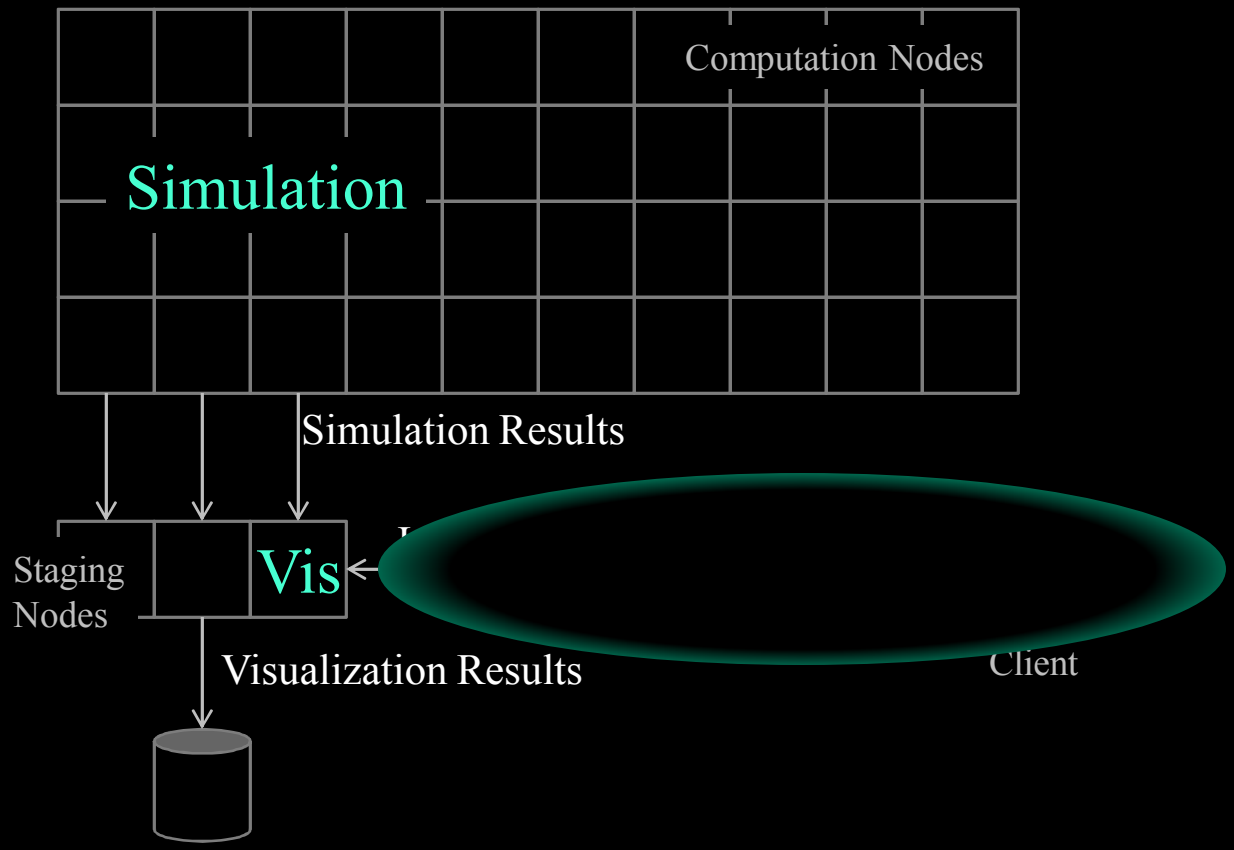
In Transit Visualization

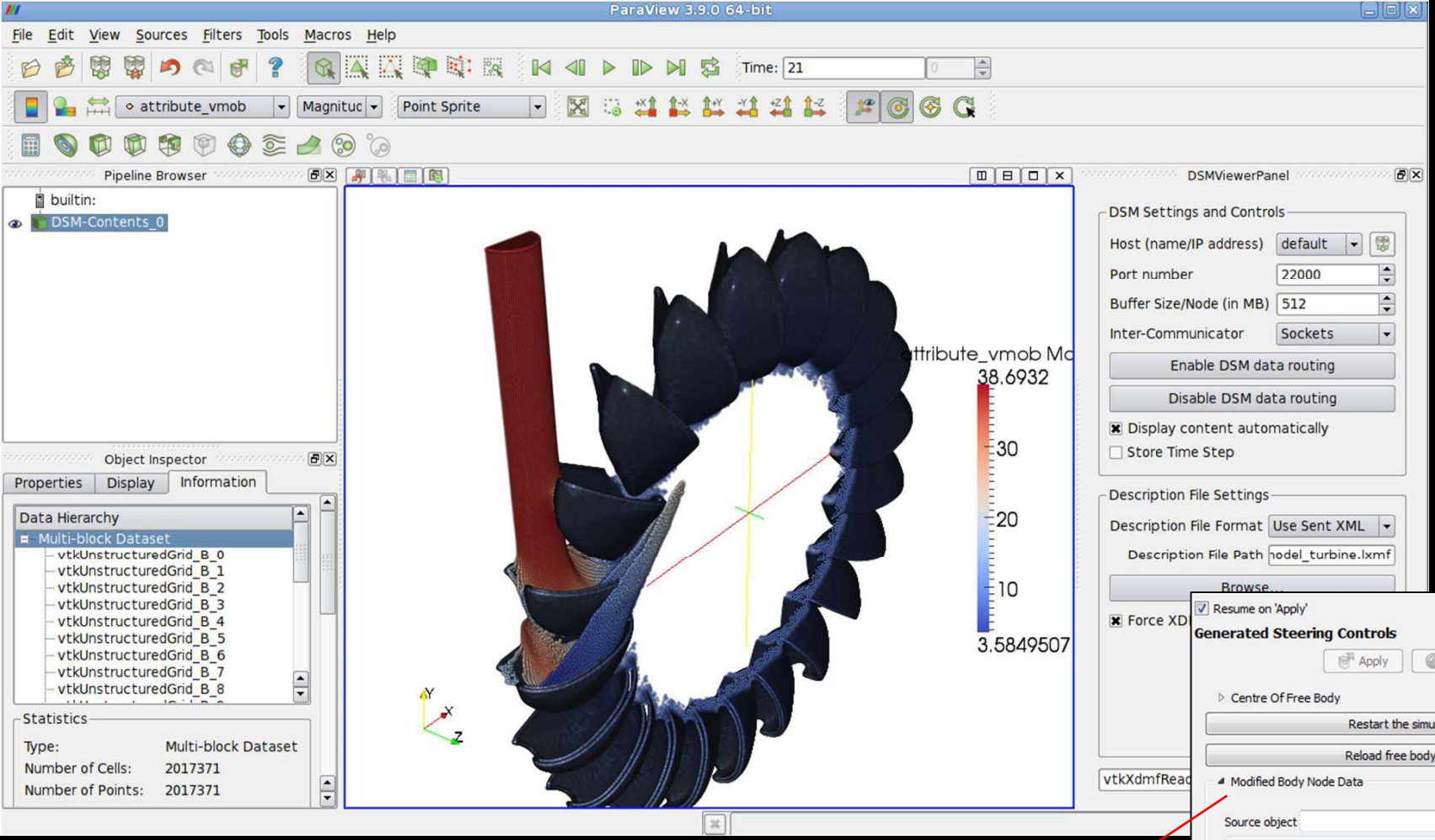


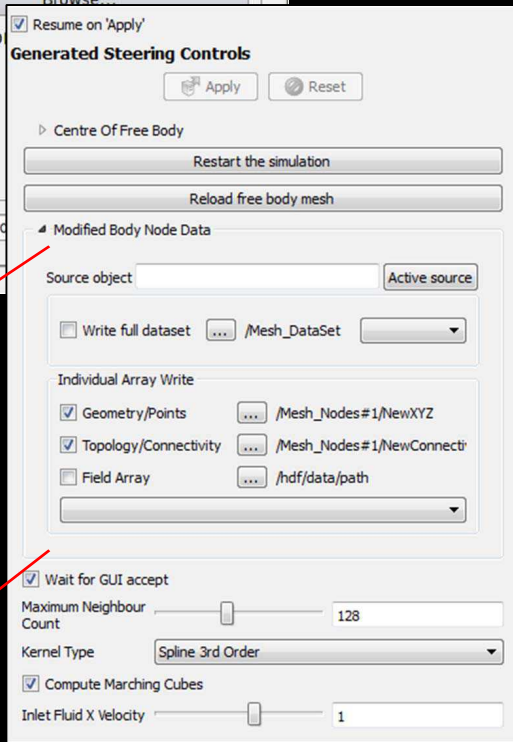
In Transit Visualization



In Transit Visualization

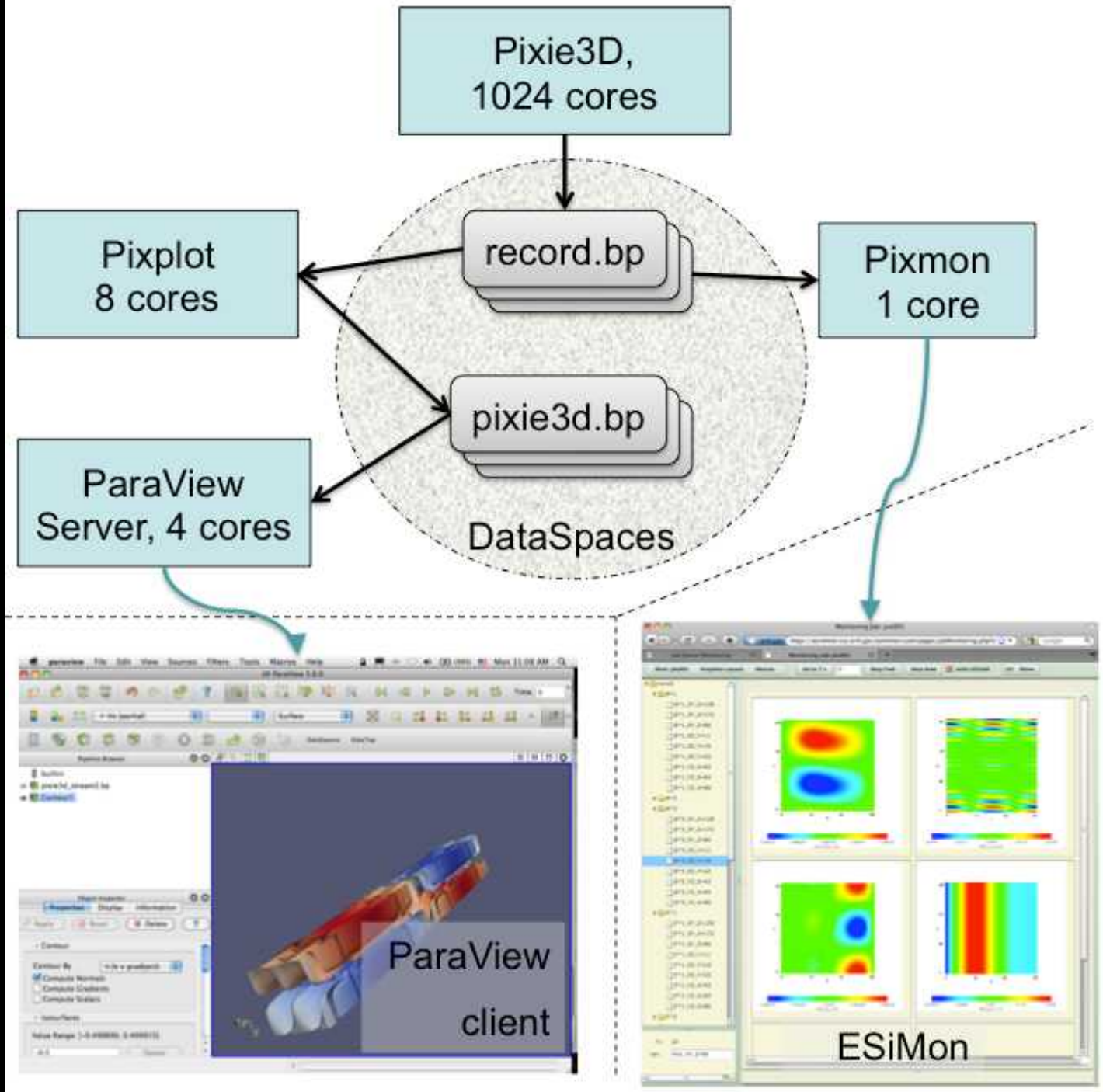







```
<DataExportProperty name="ModifiedBodyNodes"
                    command="SetSteeringArray"
                    label="Modified Body Node Data">
  <DataExportDomain name="data_export"
                    full_path="/Mesh_DataSet"
                    geometry_path="/Mesh_Nodes#1/NewXYZ"
                    topology_path="/Mesh_Nodes#1/NewCo..."
                    command_property="ReloadFreeBodyMesh" />
</DataExportProperty>
```

Biddiscombe, et al, "Parallel Computational Steering and Analysis for HPC Applications using a ParaView Interface and the HDF5 DSM Virtual File Driver."


Pixie3D,
1024 cores
Pixplot
8 cores
record.bp
Pixmon
1 core
pixie3d.bp
ParaView
Server, 4 cores
DataSpaces
ParaView
client
ESiMon

Slide of Doom

	2010	"2018"	Factor Change
System Peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

DOE Exascale Initiative Roadmap, Architecture and Technology Workshop, San Diego, December, 2009.
Reported in *The Opportunities and Challenges of Exascale Computing*, Summary Report of the ASCAC Subcommittee, Fall 2010.

Slide of Doom

	2010	"2018"	Factor Change
System Peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

DOE Exascale Initiative Roadmap, Architecture and Technology Workshop, San Diego, December, 2009.
Reported in *The Opportunities and Challenges of Exascale Computing*, Summary Report of the ASCAC Subcommittee, Fall 2010.

Naïve Scaling

	2010	"2018"	Factor Change
System Memory	0.3 PB	10 PB	33
Total Concurrency	225 K	1 B	4,444

MPI Only?

Vis object code + state: 20MB

On Jaguar: 20MB × 200,000 processes = 4TB

On Exascale: 20MB × 1 billion processes = 20PB !

DOE Exascale Initiative Roadmap, Architecture and Technology Workshop, San Diego, December, 2009.
Reported in *The Opportunities and Challenges of Exascale Computing*, Summary Report of the ASCAC Subcommittee, Fall 2010.

Naïve Scaling

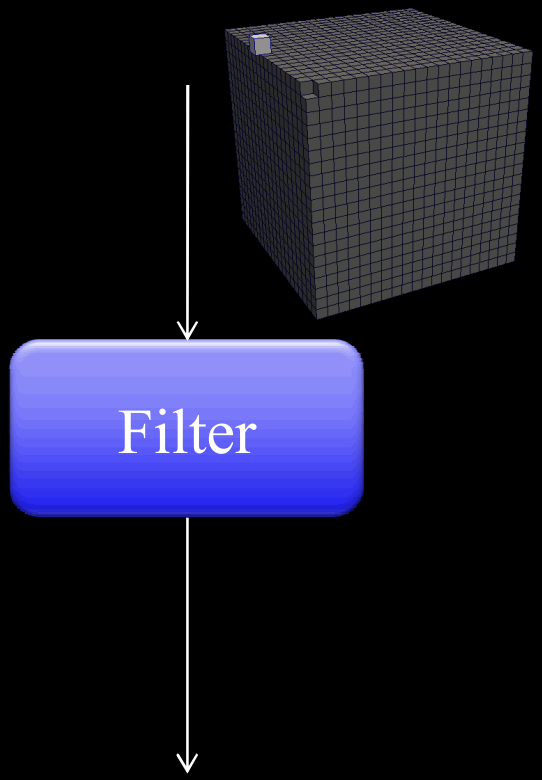
	2010	"2018"	Factor Change
System Memory	0.3 PB	10 PB	33
Total Concurrency	225 K	1 B	4,444

Visualization pipeline too heavyweight?

- On Jaguar: 1 trillion cells → 5 million cells/thread
- On Exascale: 30 trillion cells → 30K cells/thread

DOE Exascale Initiative Roadmap, Architecture and Technology Workshop, San Diego, December, 2009.
Reported in *The Opportunities and Challenges of Exascale Computing*, Summary Report of the ASCAC Subcommittee, Fall 2010.

Revisiting the Pipeline



- Lightweight Object
- Serial Execution
- No explicit partitioning
- No access to larger structures
- No state
- Changing programming models

Dax Makes it Easy

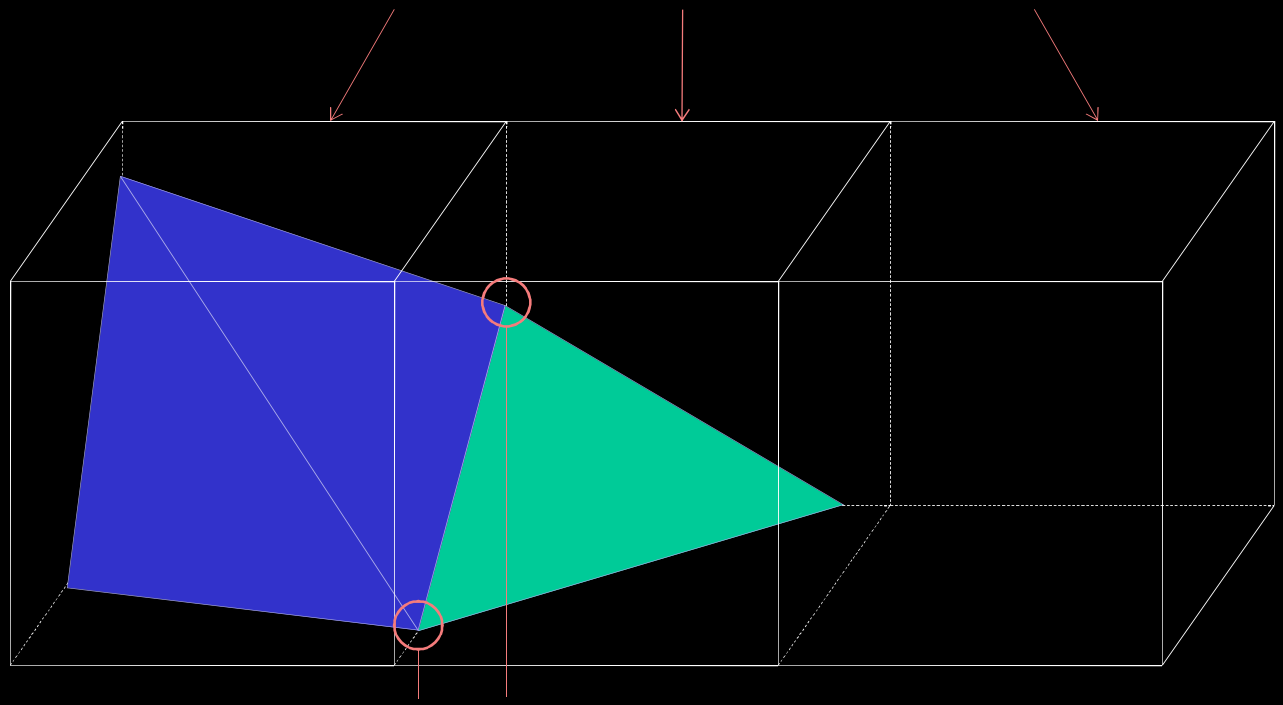
```
int vtkCellDerivatives::RequestData(...)
{
  ...[allocate output arrays]...
  ...[validate inputs]...
  for (cellId=0; cellId < numCells; cellId++)
    {
    ...[update progress]...
    input->GetCell(cellId, cell);
    subId
      = cell->GetParametricCenter(pcoords);
    inScalars->GetTuples(cell->PointIds,
                         cellScalars);
    scalars = cellScalars->GetPointer(0);
    cell->Derivatives(subId,
                      pcoords,
                      scalars,
                      1,
                      derivs);
    outGradients->SetTuple(cellId, derivs);
    }
  ...[cleanup]...
}
```

```
DAX_WORKLET void CellGradient(...)
{


  dax::exec::Cell cell(work);
  dax::Vector3 parametric_cell_center
    = dax::make_Vector3(0.5, 0.5, 0.5);


  dax::Vector3 value = cell.Derivative(
    parametric_cell_center,
    points,
    point_attribute,
    0);
  cell_attribute.Set(work, value);


}
```

Threaded Programming is Hard
Example: Marching Cubes

Sandia National Laboratories

Easy because cubes can be processed in parallel, right?



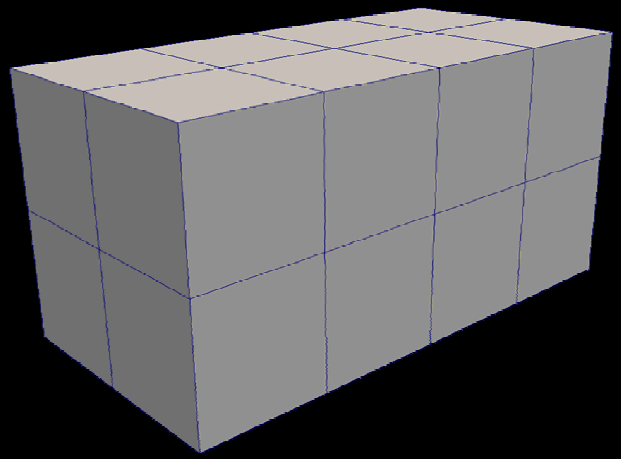
How do you resolve coincident points?
How do you capture topological connections?

How do you pack the results?

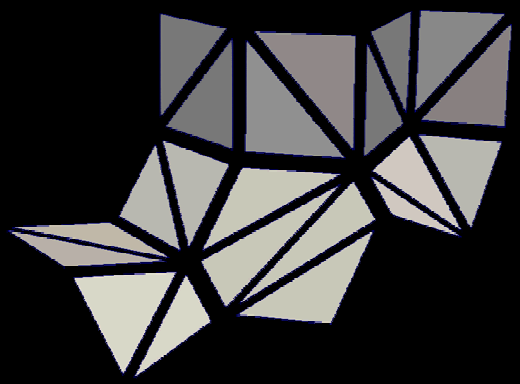
Work in Progress: Connectivity-Based Algorithms

Sandia National Laboratories

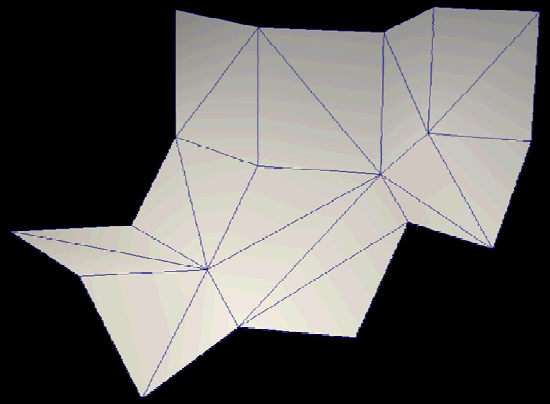
Input Volume



Triangle Soup

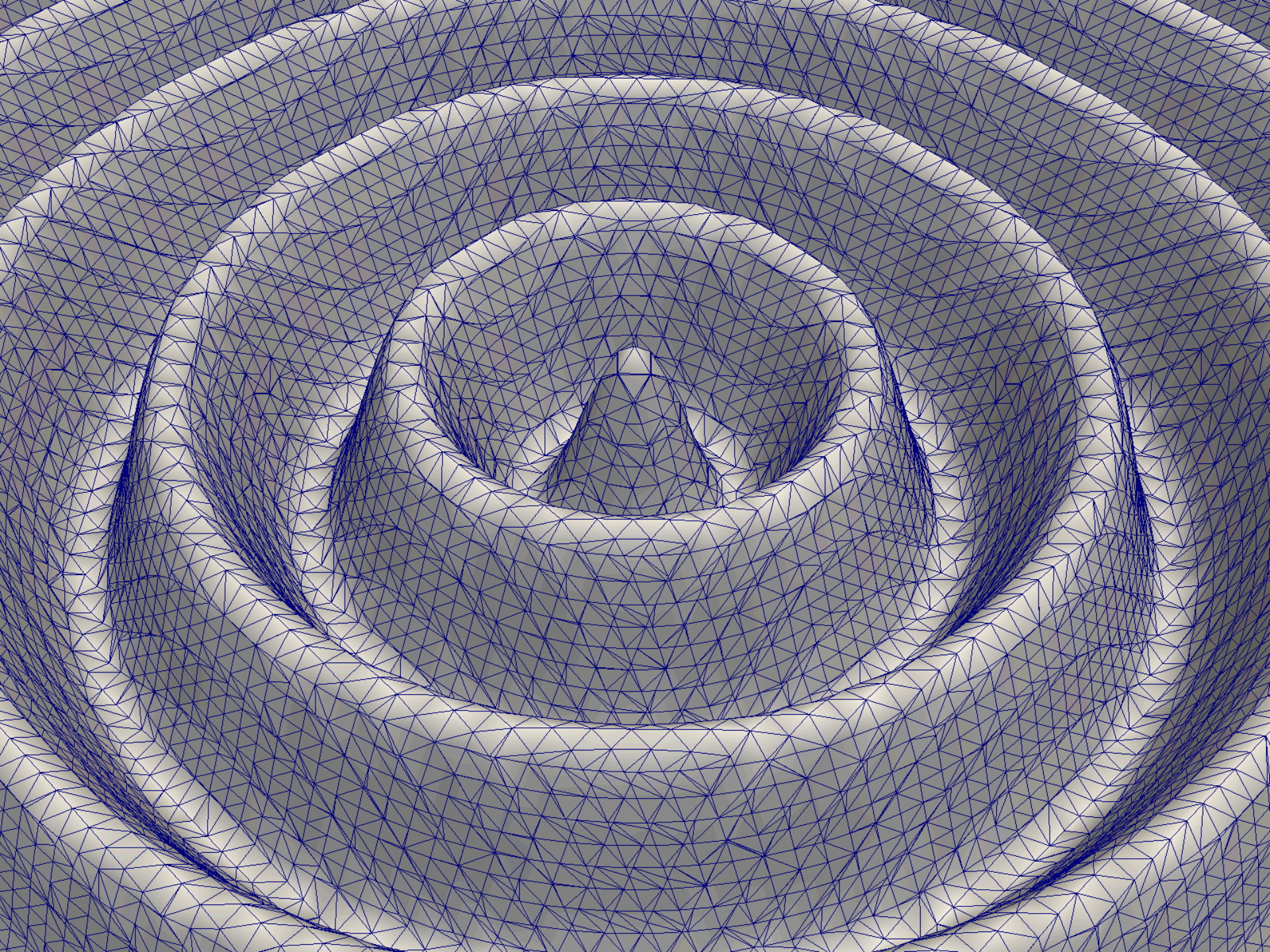


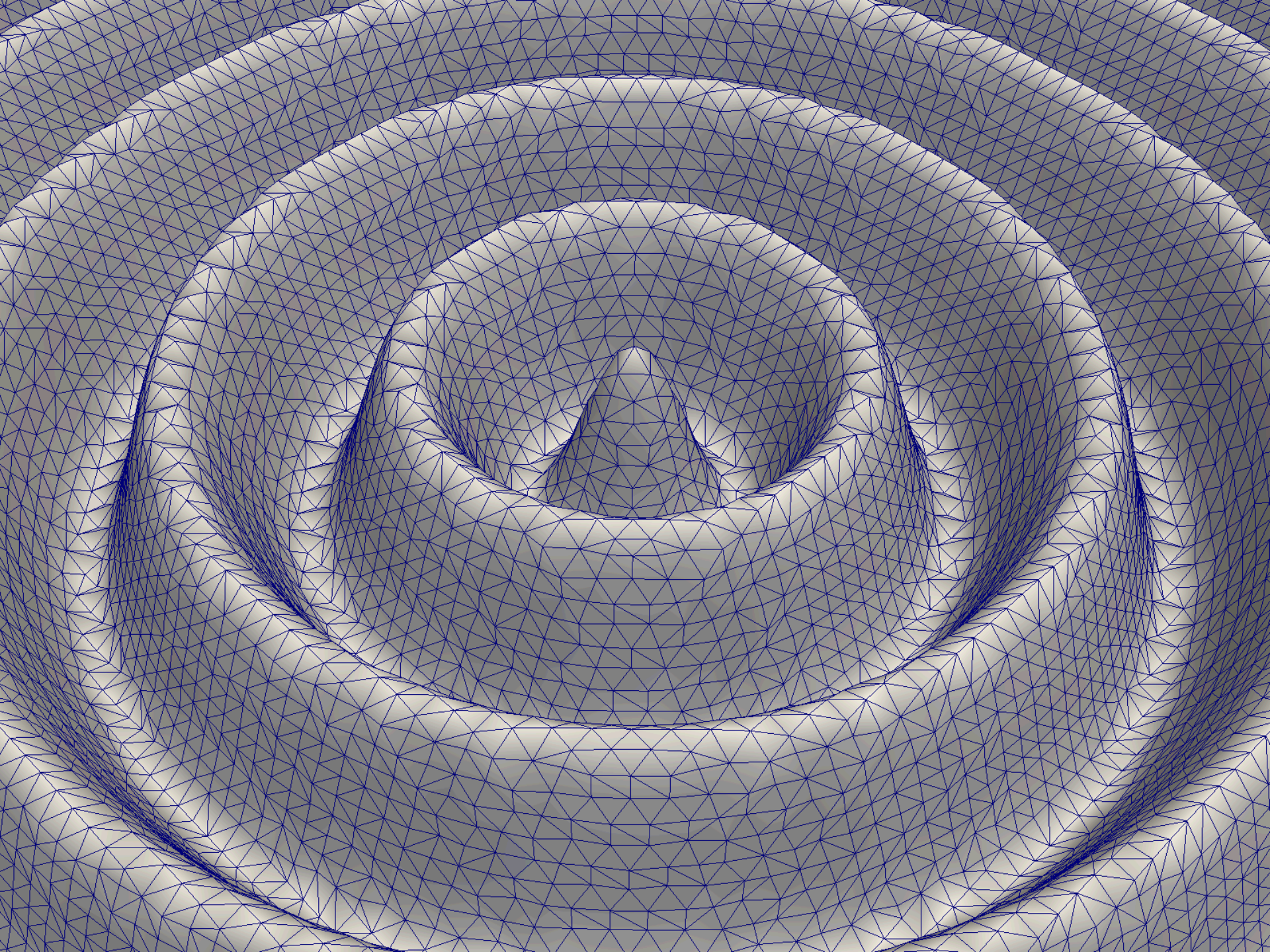
Manifold Surface



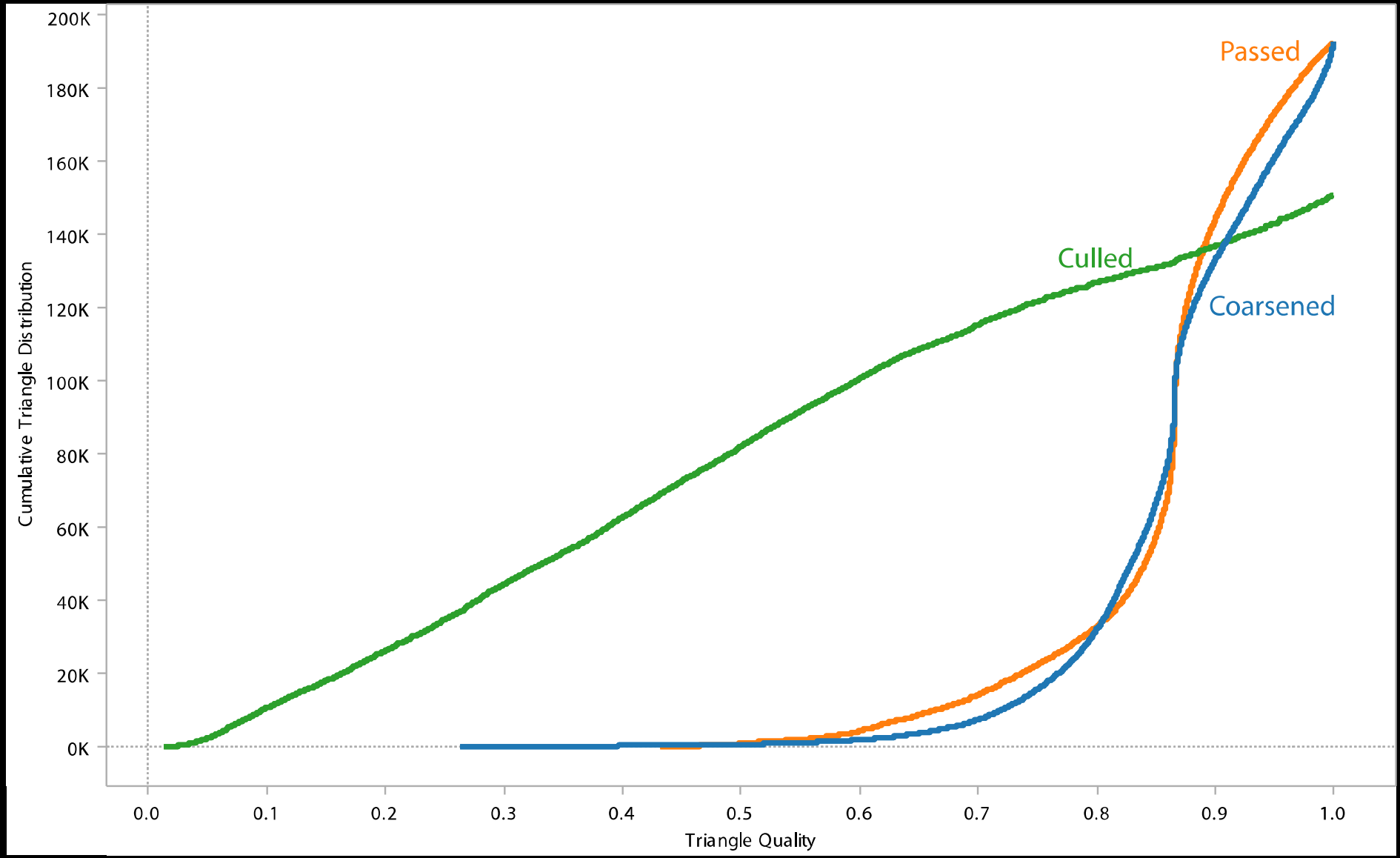
Generate Triangles
(Local)

Connect Surface
(Communicative)

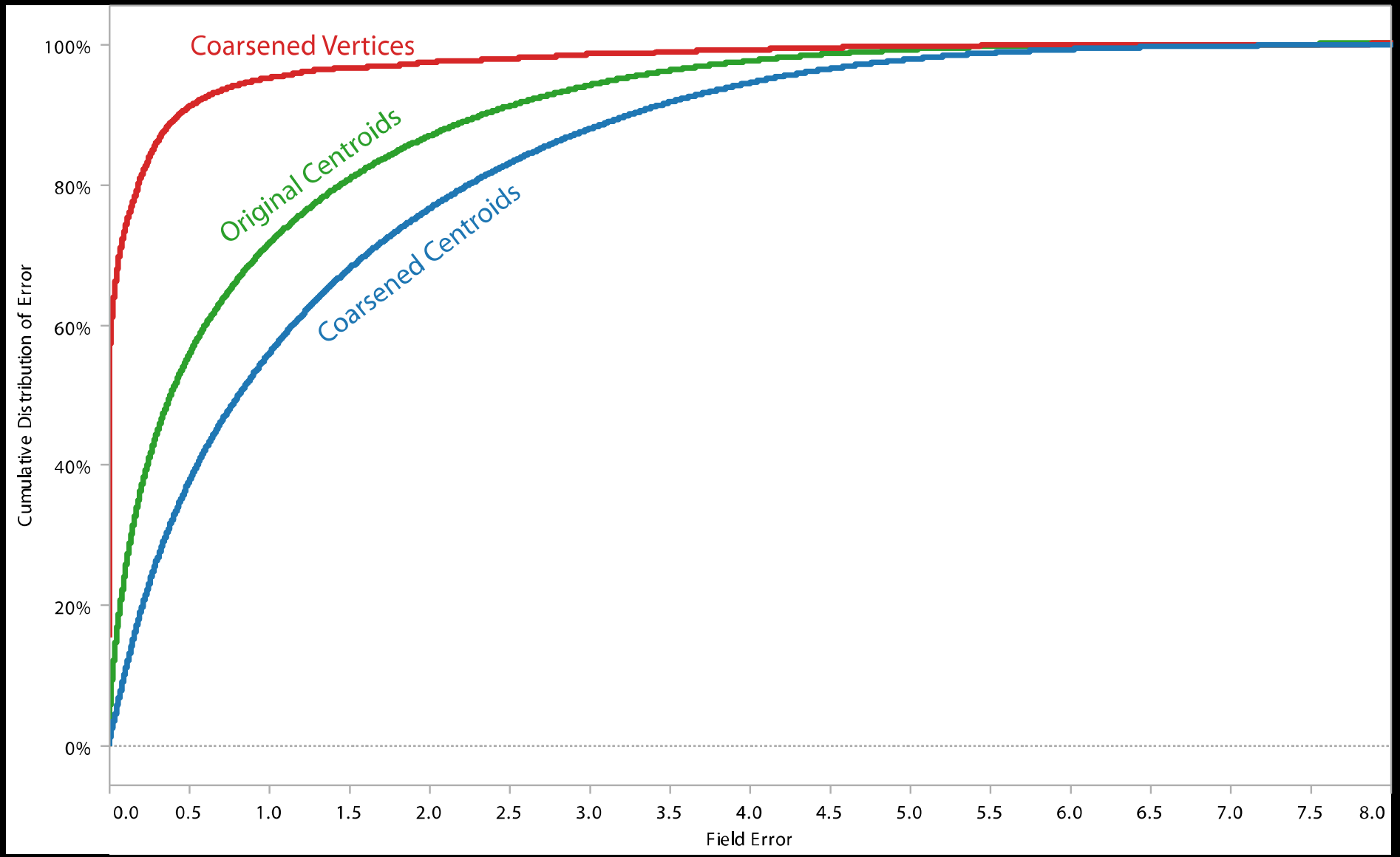




Coarsened Triangle Quality



Coarsened Mesh Error



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

Questions?