Exceptional service in the national interest

SAND2014-1199P

Sandia National Laboratories

Photos placed in horizontal position with even amount of white space between photos and header

Approaching Production In Situ Visualization for Extreme Scale Analysis

SIAM PP 2014

Kenneth Moreland	Sandia National Laboratories
Ron Oldfield	Sandia National Laboratories
Nathan Fabian	Sandia National Laboratories
Berk Geveci	Kitware, Inc.
Andrew Bauer	Kitware, Inc.
David Lonie	Kitware, Inc.

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

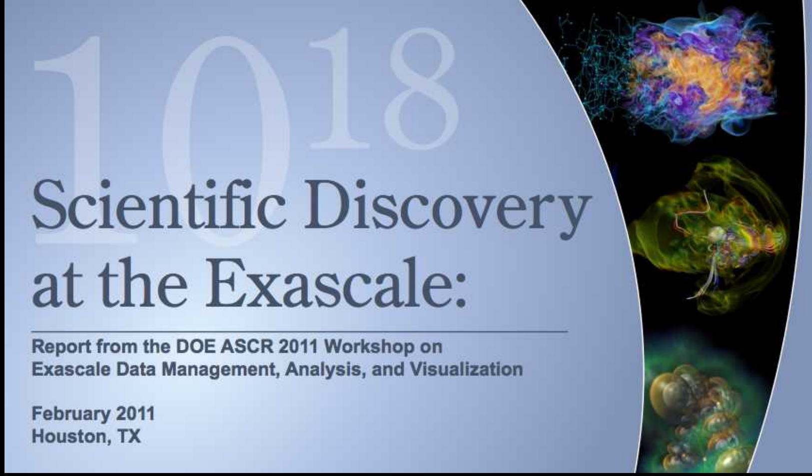
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Collaborators

- **Sandia National Laboratories**
  - Kenneth Moreland
  - Nathan Fabian
  - Ron Oldfield
- **Los Alamos National Laboratories**
  - James Ahrens
  - Jonathan Woodring
  - John Patchett
  - David Rogers
- **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 Thompson
  - David DeMarle
- **University of Colorado at Boulder**
  - Michel Rasquin
  - Kenneth E. Jansen
- **Rutgers University**
  - Ciprian Docan
  - Manish Parashar

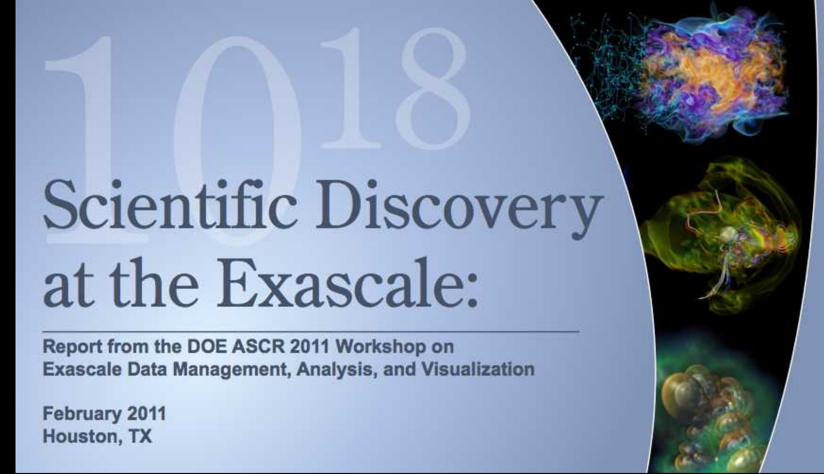
Sandia National Laboratories

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		10 – 30

Slide of Doom



System Parameter	2011	"2018"		Factor Change
System Peak	2 PetaFLOPS	1 ExaFLOP		
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		
I/O BW	0.2 TB/s	20 – 60 TB/s		

Extreme Scale Computing

- 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

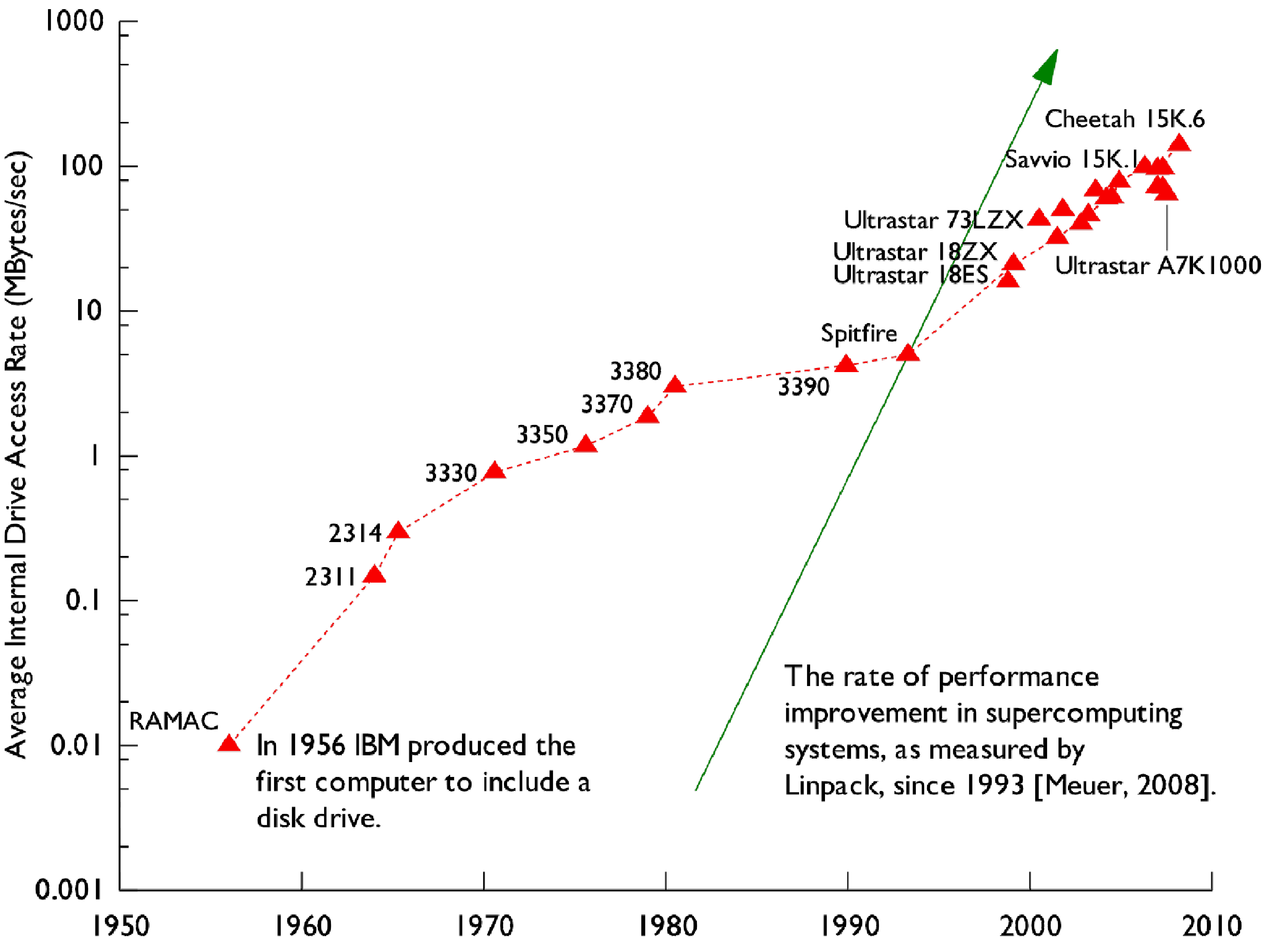
	System Peak	I/O BW
ASC Purple (2005)	100 TFLOPS	106 GB/s
Sequioa (2012)	20 PFLOPS	1 TB/s

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

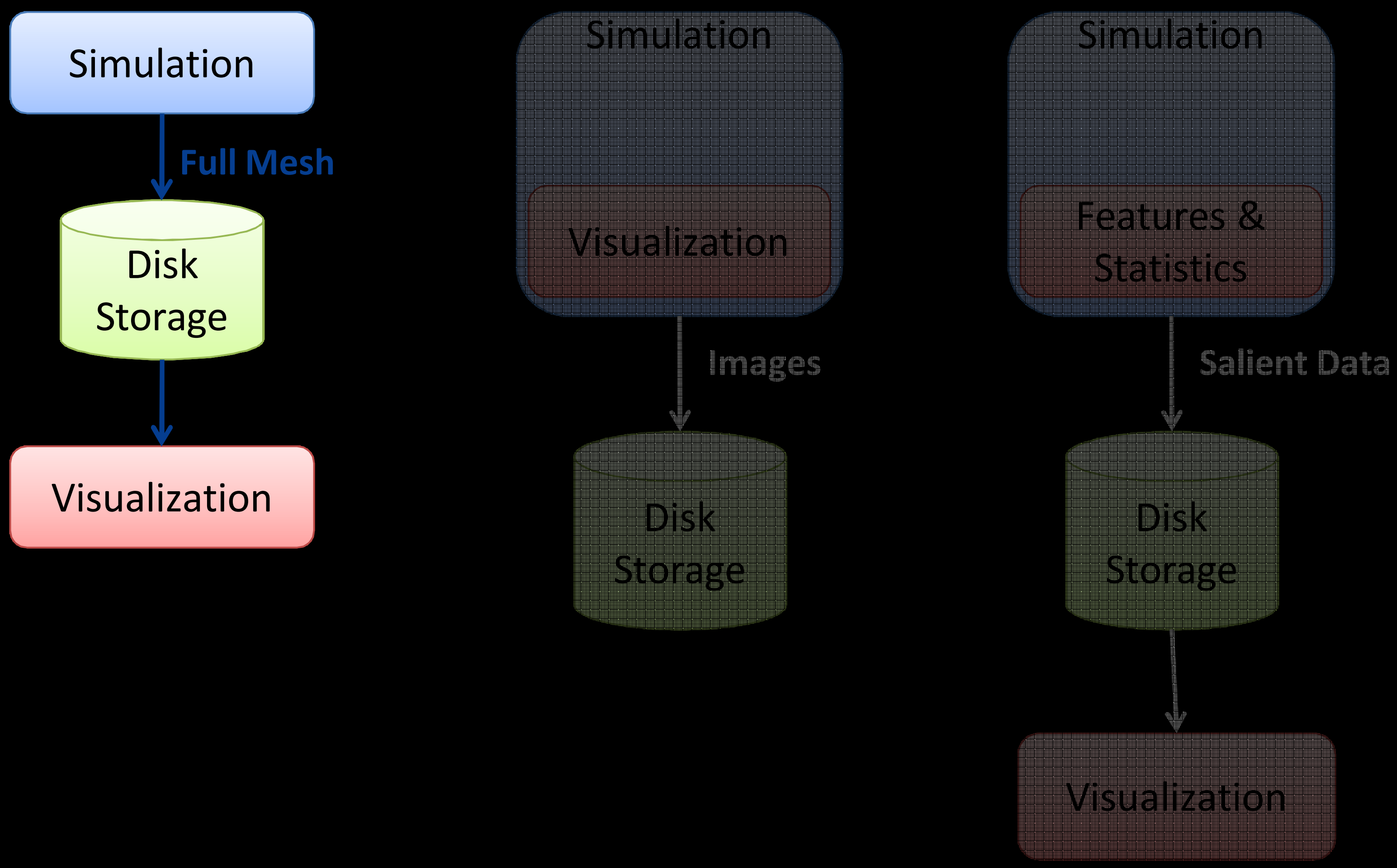
Sandia National Laboratories

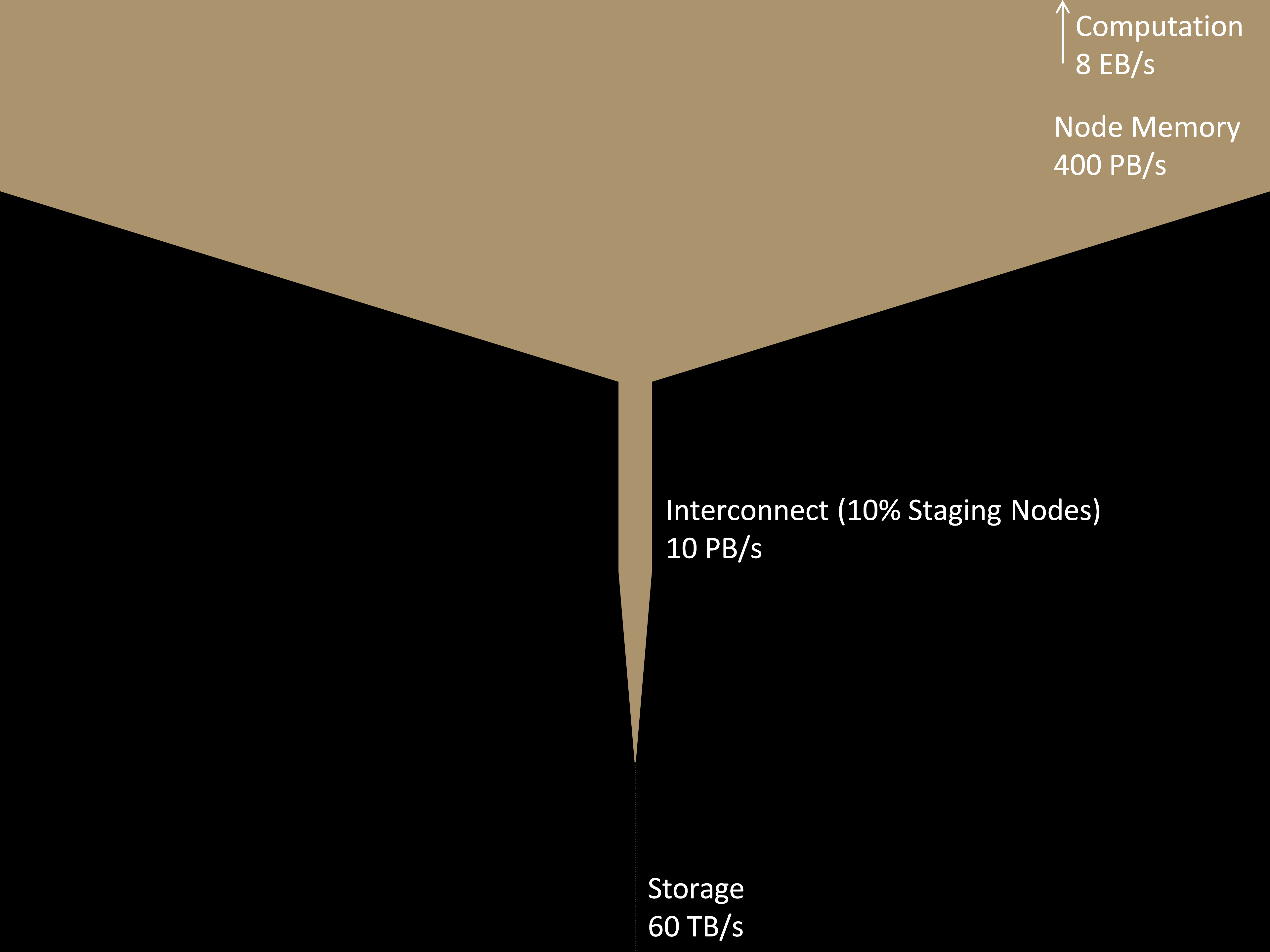
	System Peak	I/O BW
Red Storm (2003)	180 TFLOPS	100 GB/s
Cielo (2011)	1.4 PFLOPS	160 GB/s

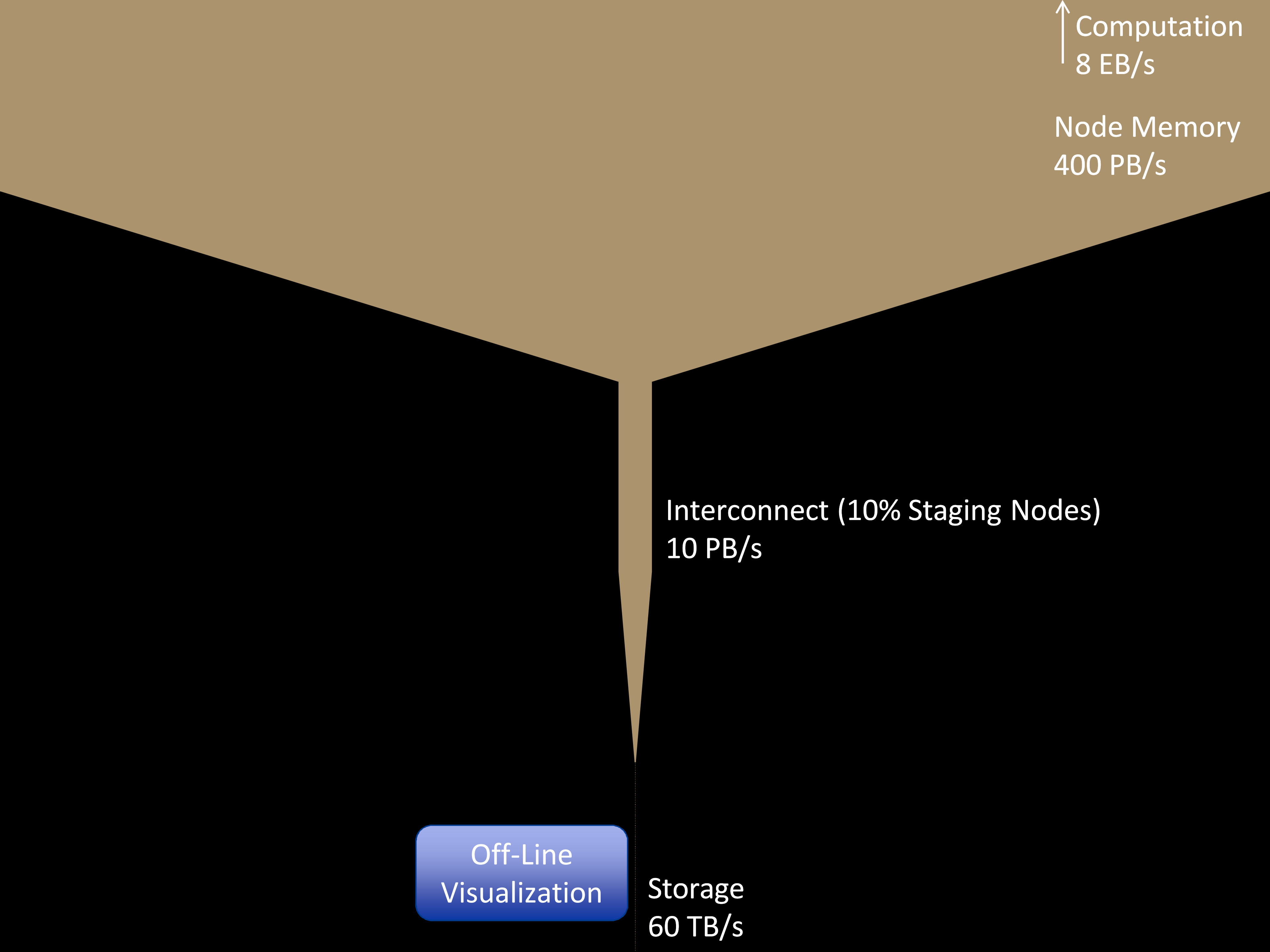
https://cfwebprod.sandia.gov/cfdocs/CCIM/docs/033768p.pdf
http://www.lanl.gov/orgs/hpc/cielo/


Average Internal Drive Access Rate (MBytes/sec)
1000
100
10
1
0.1
0.01
0.001
1950
1960
1970
1980
1990
2000
2010
RAMAC
2311
2314
3330
3350
3370
3380
3390
Spitfire
Ultrastar 18ES
Ultrastar 18ZX
Ultrastar 73LZX
Savvio 15K.1
Cheetah 15K.6
Ultrastar A7K1000
In 1956 IBM produced the first computer to include a disk drive.
The rate of performance improvement in supercomputing systems, as measured by Linpack, since 1993 [Meuer, 2008].

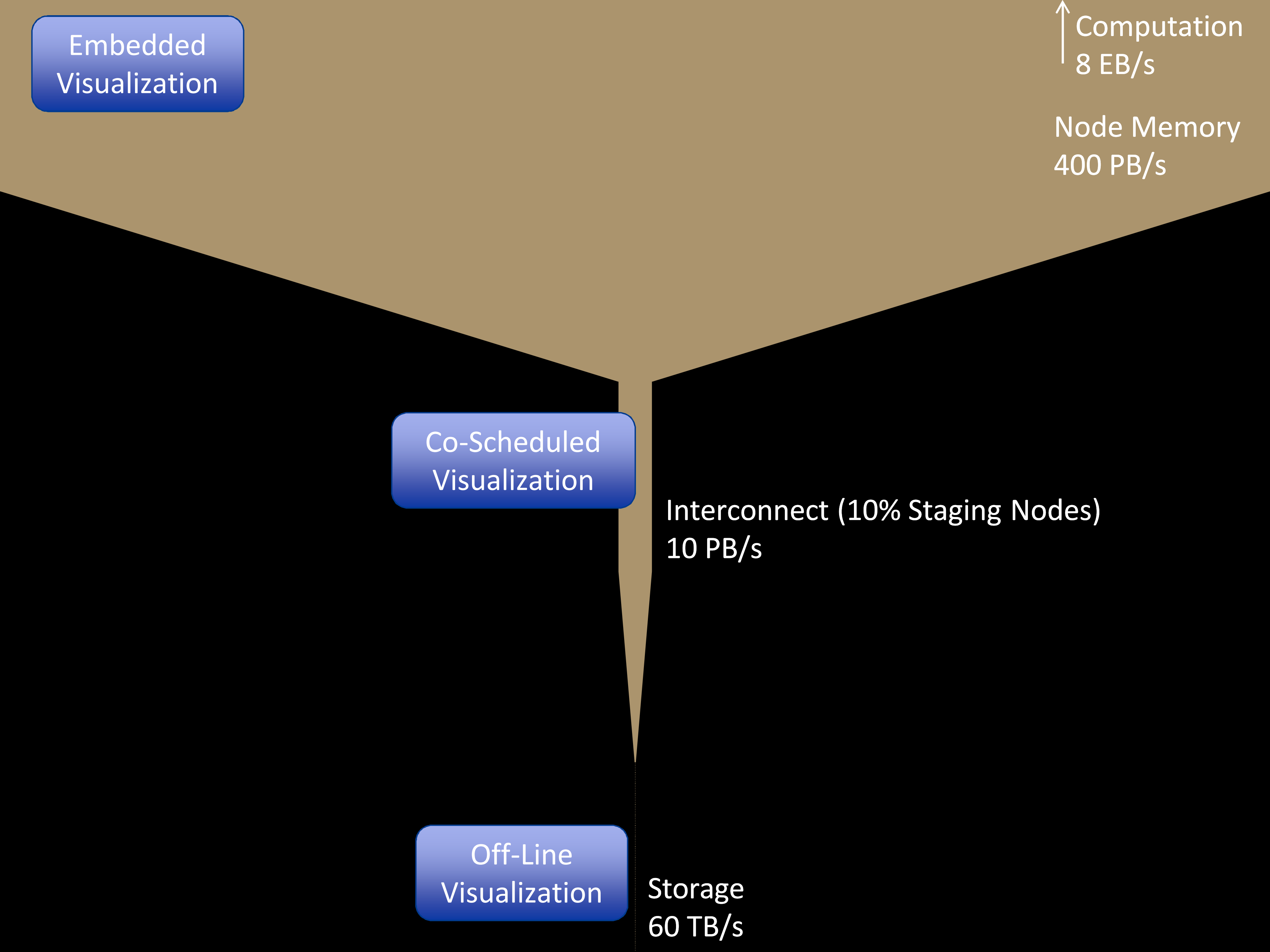
Diskless Visualization



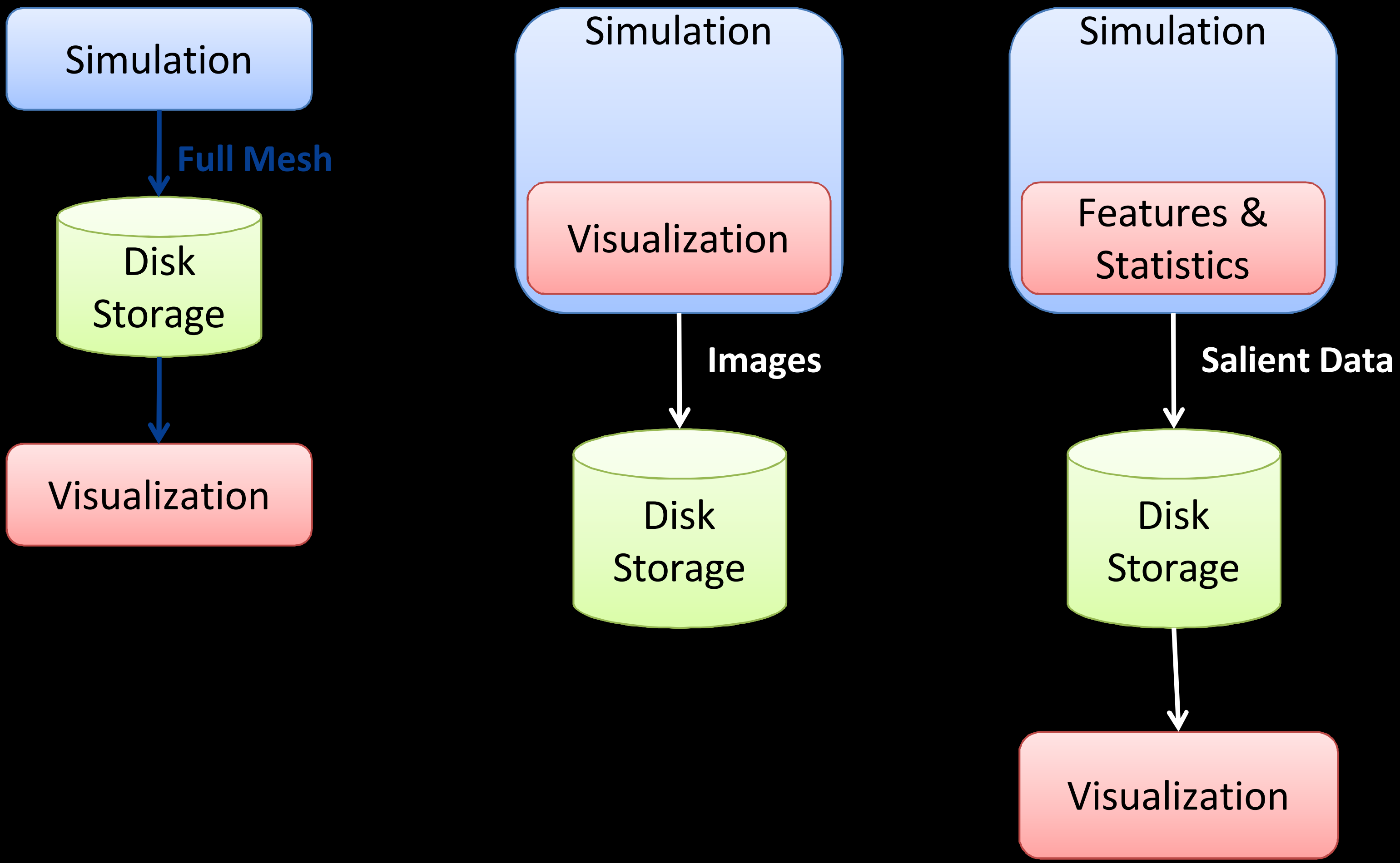

Computation
8 EB/s
Node Memory
400 PB/s
Interconnect (10% Staging Nodes)
10 PB/s
Storage
60 TB/s

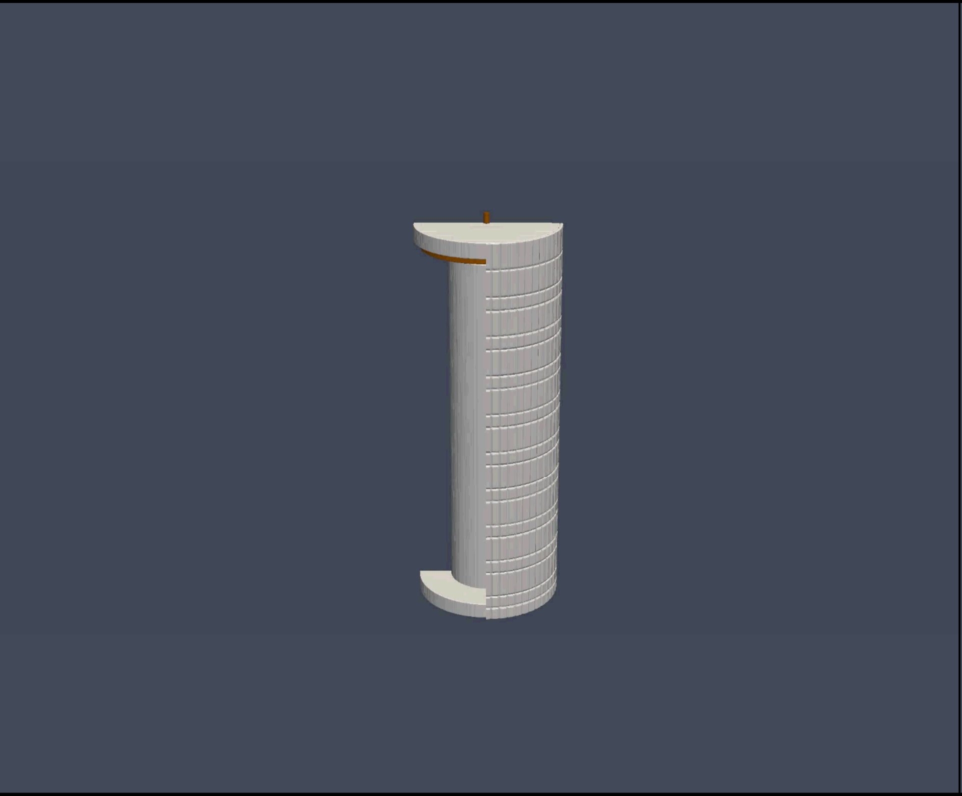


Computation
8 EB/s
Node Memory
400 PB/s
Interconnect (10% Staging Nodes)
10 PB/s
Off-Line
Visualization
Storage
60 TB/s

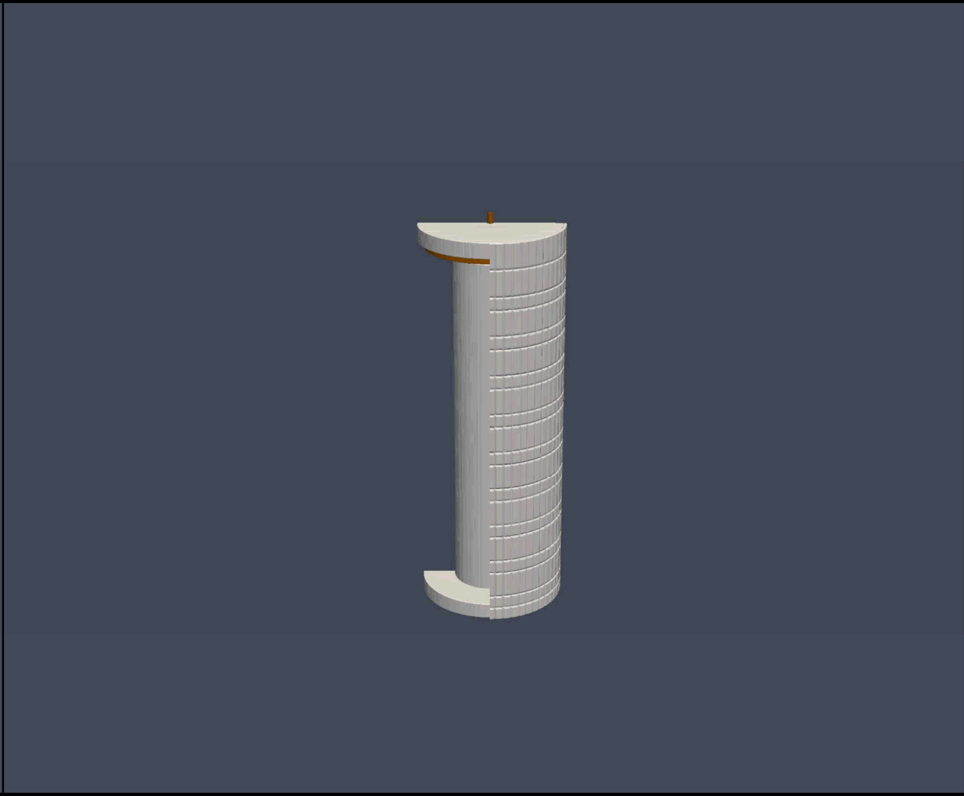

Embedded Visualization
Computation 8 EB/s
Node Memory 400 PB/s
Co-Scheduled Visualization
Interconnect (10% Staging Nodes) 10 PB/s
Off-Line Visualization
Storage 60 TB/s

Diskless Visualization





Limited off-line data access.



Full in situ data access.

The Perils of In Situ Integration

Simulation

Visualization

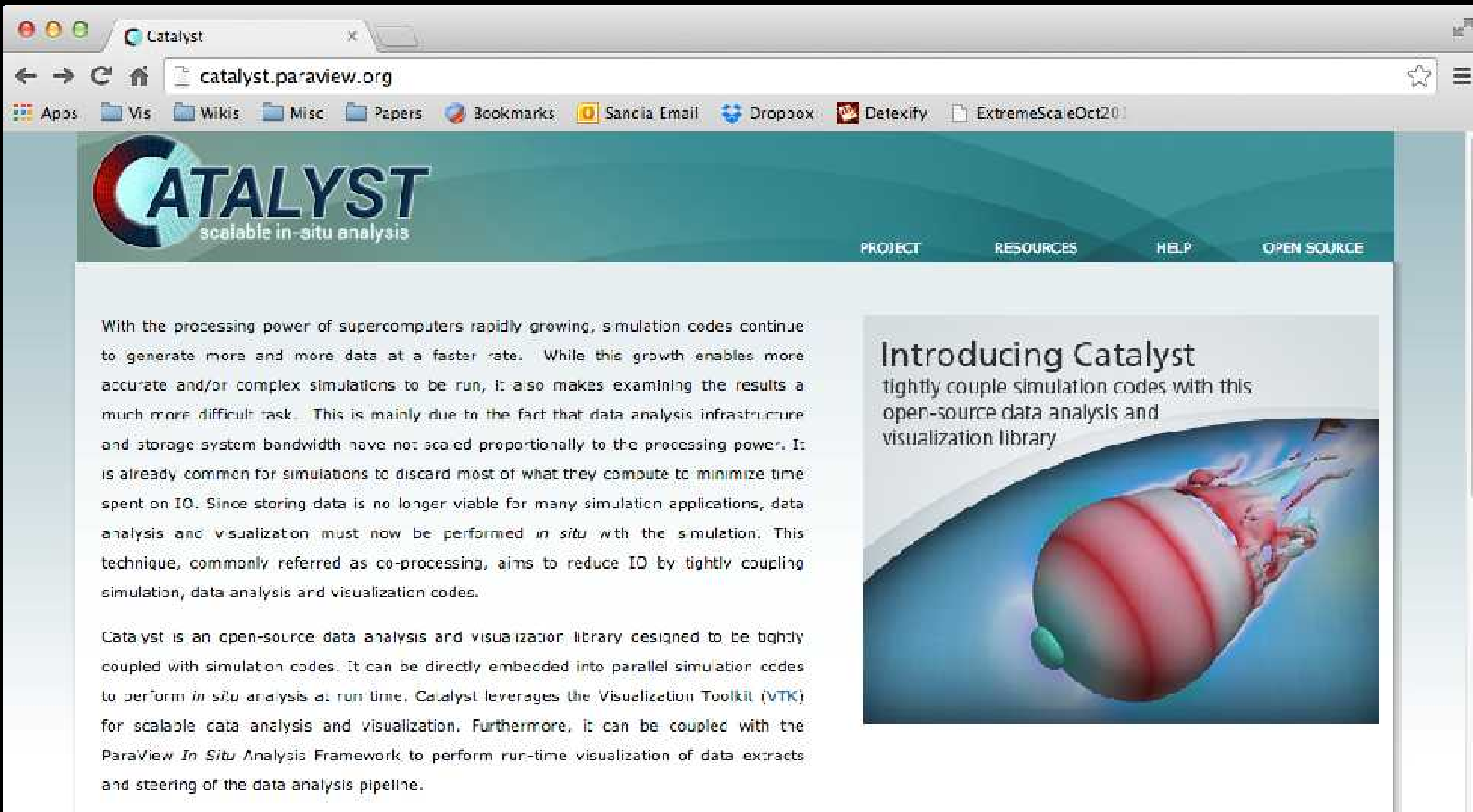
Source: Archive of Ridiculous Metaphors

Proper In Situ Integration

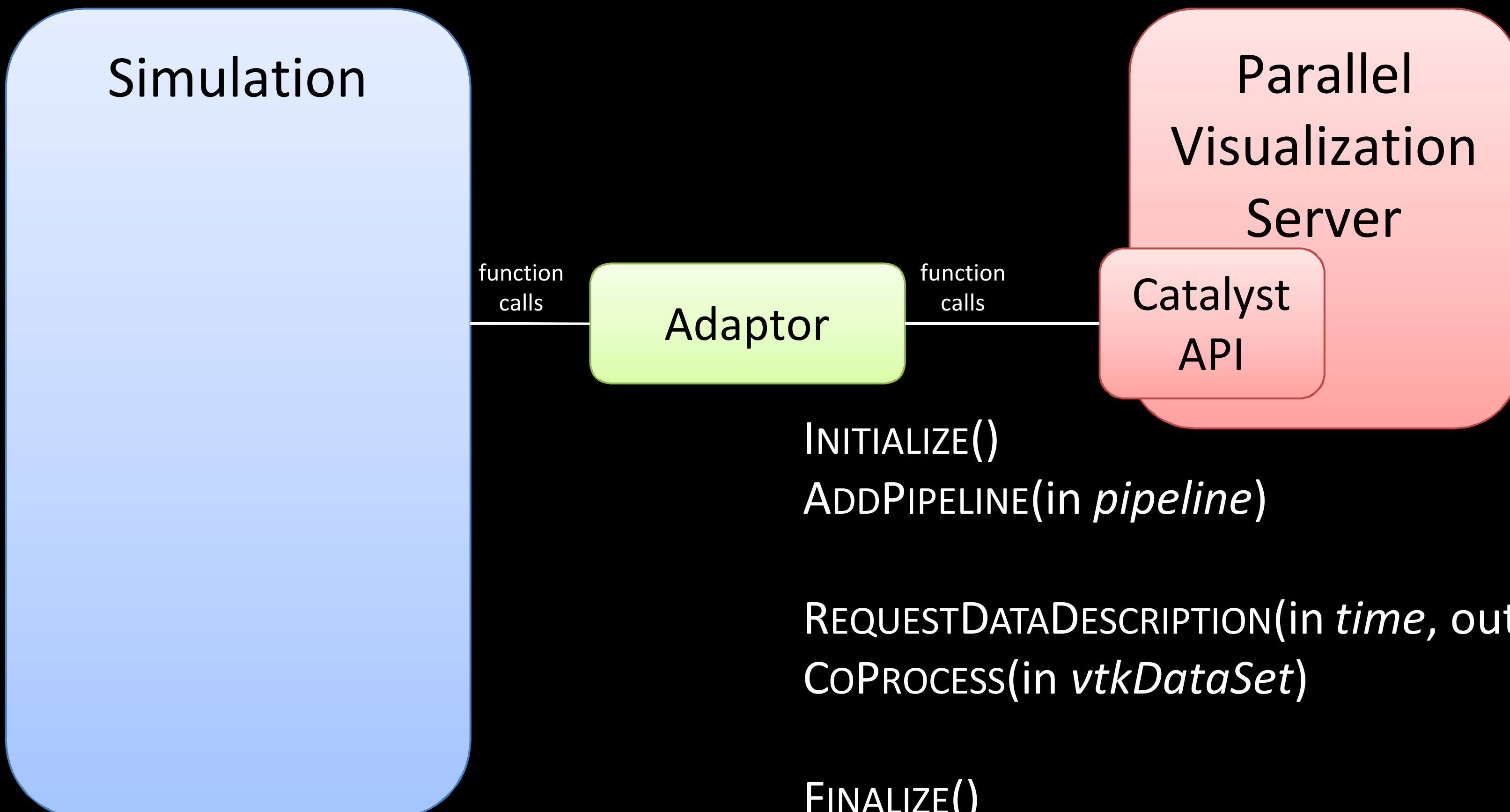
Source: Archive of Ridiculous Metaphors

The Catalyst In Situ Library

Sandia National Laboratories



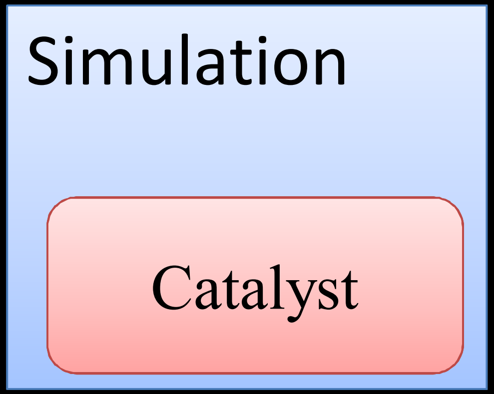
Catalyst-Simulation Integration



INITIALIZE()
ADDPIPELINE(in *pipeline*)

REQUESTDATADESCRIPTION(in *time*, out *fields*)
COPROCESS(in *vtkDataSet*)

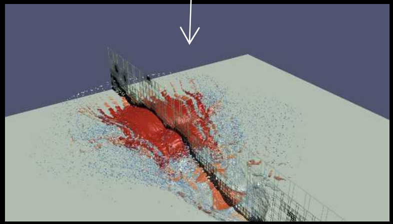
FINALIZE()


Simulation
Catalyst

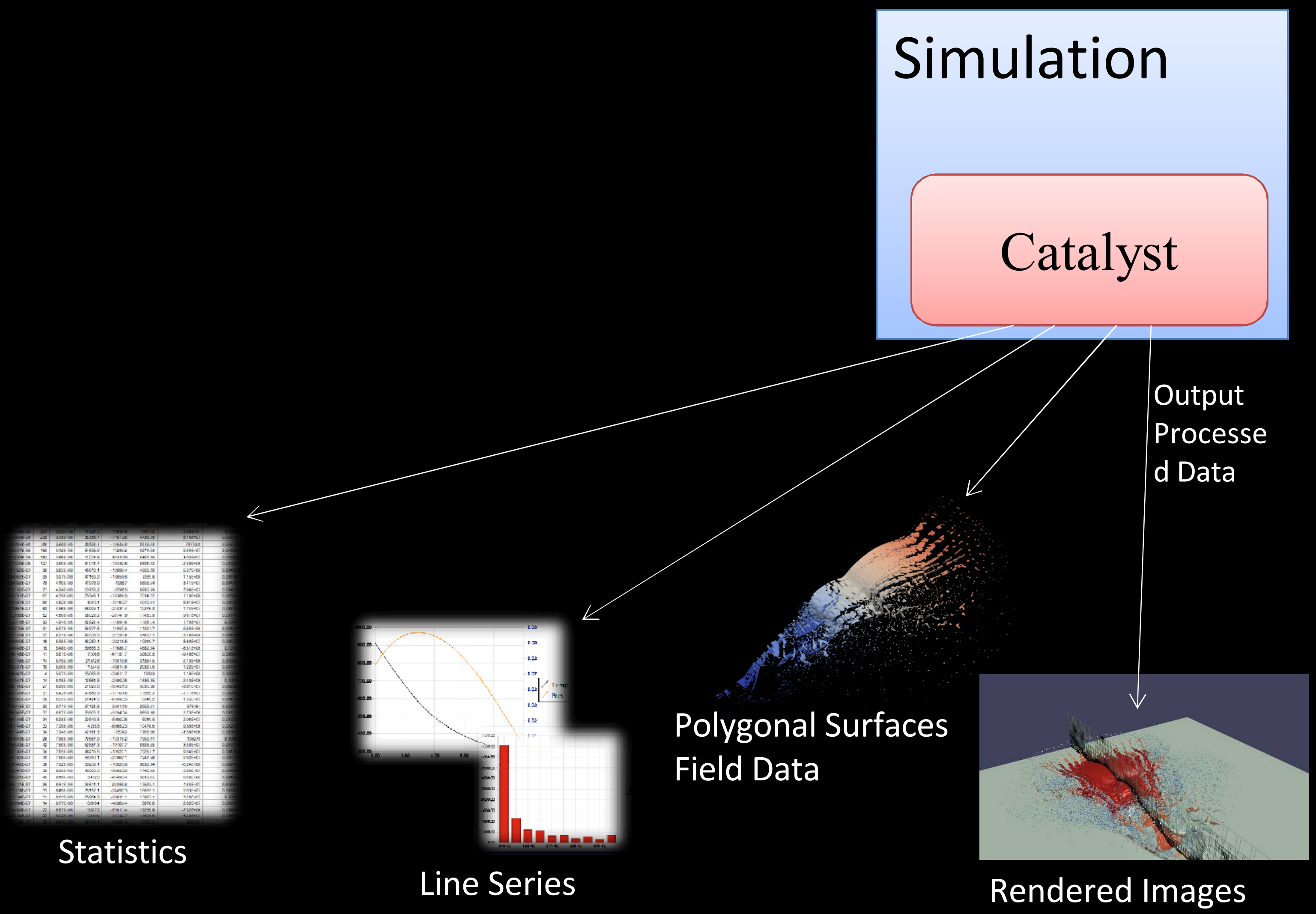
Simulation

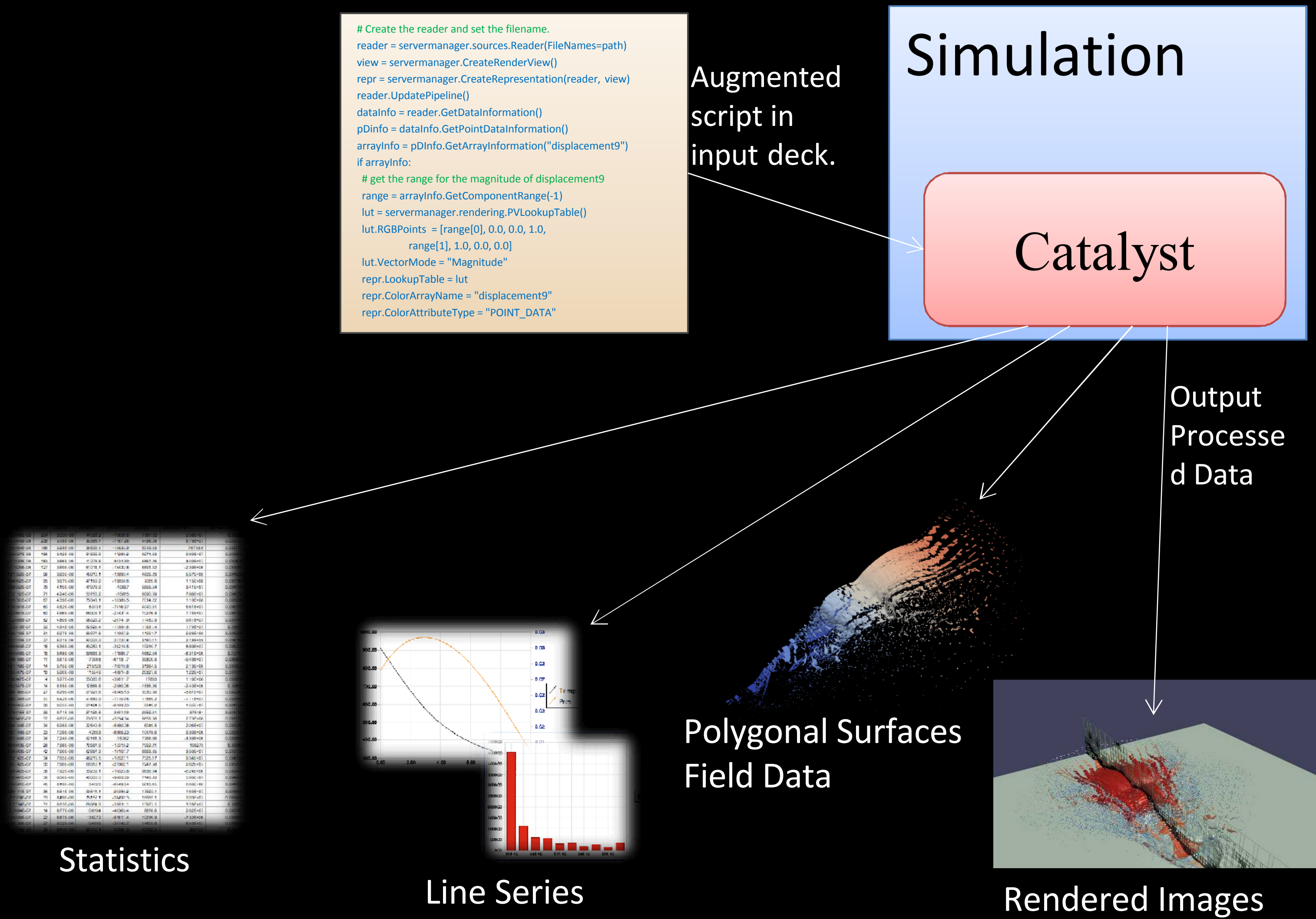
Catalyst

Output Processed Data

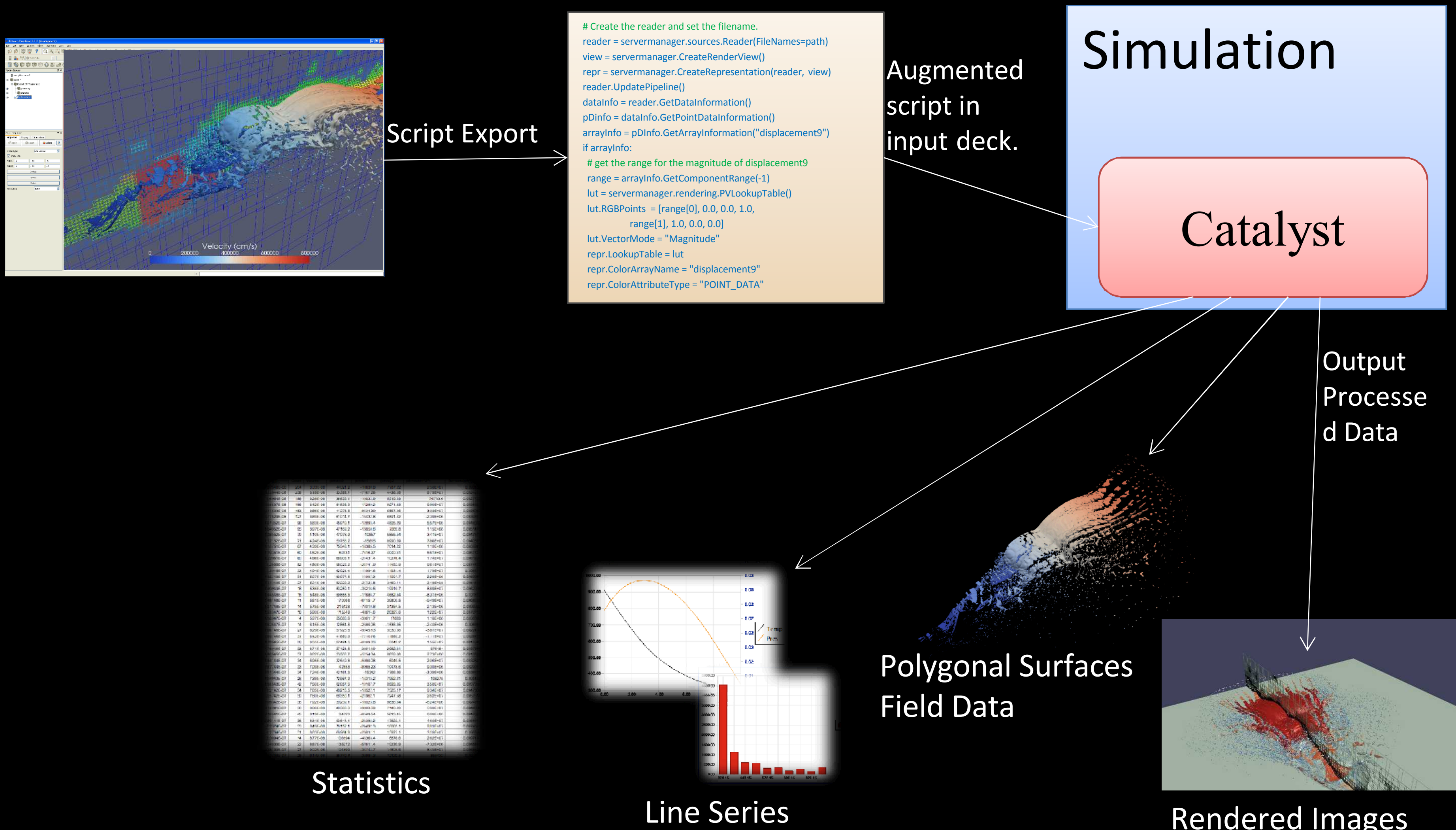


Rendered Images


Simulation
Catalyst
Output
Processe
d Data
Statistics
Line Series
Polygonal Surfaces
Field Data
Rendered Images

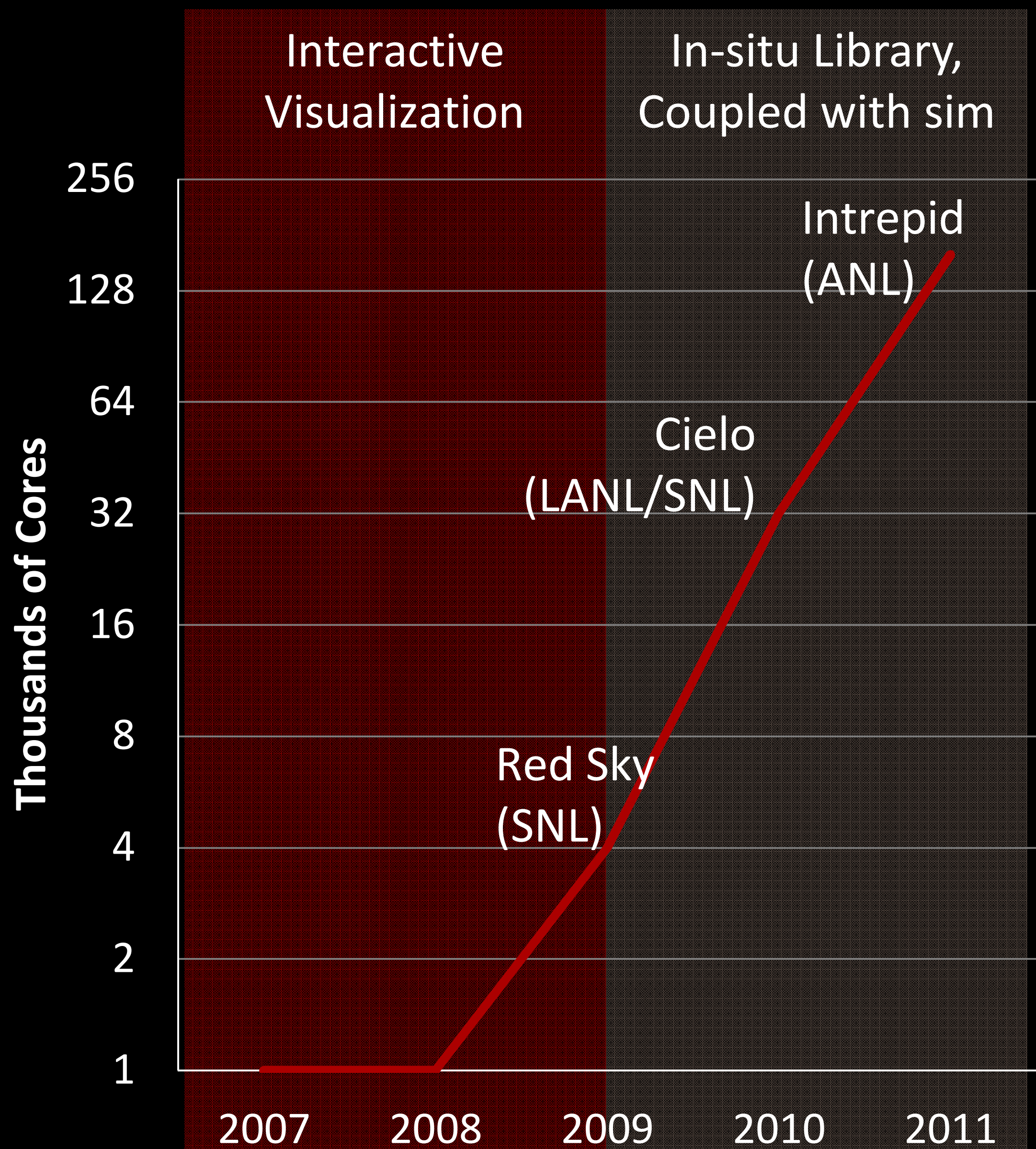


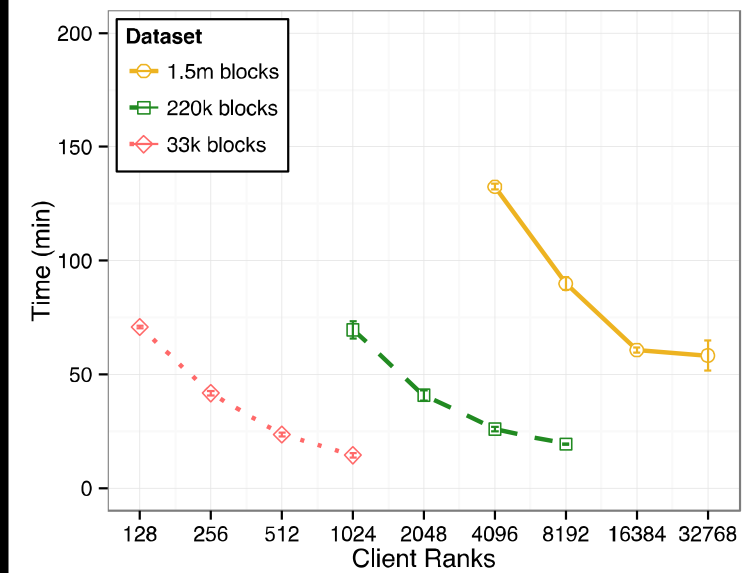
# Create the reader and set the filename.
reader = servermanager.sources.Reader(FileNames=path)
view = servermanager.CreateRenderView()
repr = servermanager.CreateRepresentation(reader, view)
reader.UpdatePipeline()
dataInfo = reader.GetDataInformation()
pDinfo = dataInfo.GetPointDataInformation()
arrayInfo = 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
Catalyst
Output Processed Data
Statistics
Line Series
Polygonal Surfaces
Field Data
Rendered Images

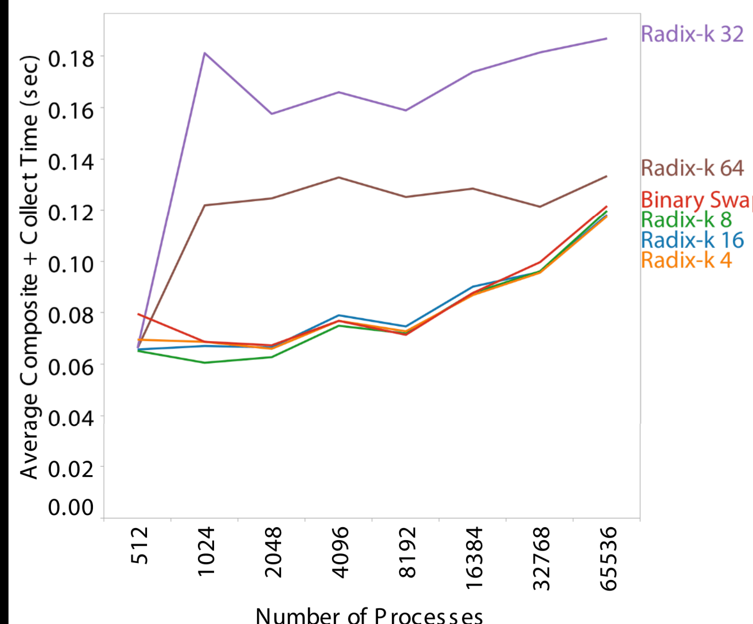


Script Export
# Create the reader and set the filename.
reader = servermanager.sources.Reader(FileNames=path)
view = servermanager.CreateRenderView()
repr = servermanager.CreateRepresentation(reader, view)
reader.UpdatePipeline()
dataInfo = reader.GetDataInformation()
pDinfo = dataInfo.GetPointDataInformation()
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
Catalyst
Output Processe d Data
Velocity (cm/s)
Statistics
Line Series
Polygonal Surfaces
Field Data
Rendered Images

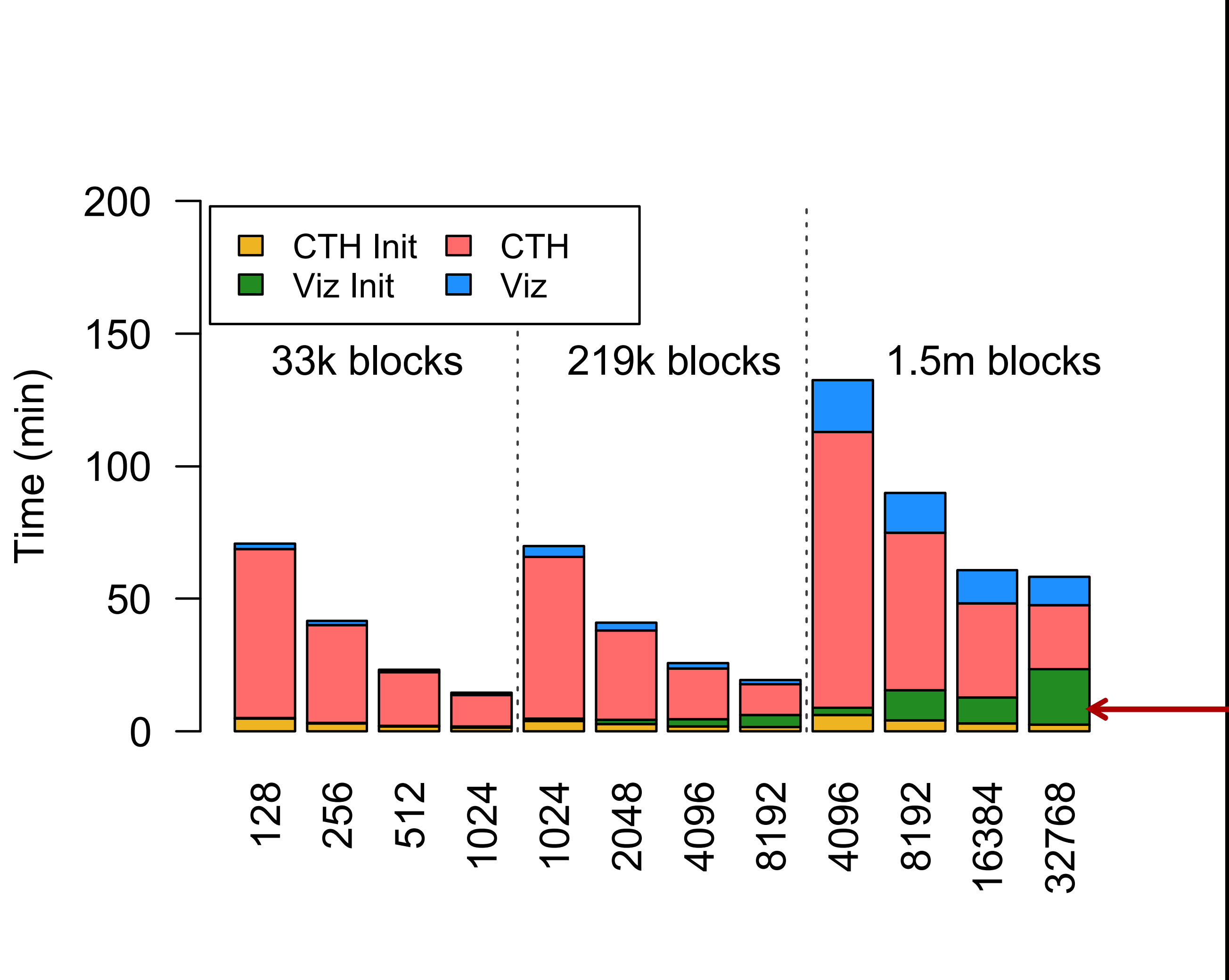
In Situ Library Challenges: Scaling







Example of Scaling Biting You



- Initialization time problematic
- Traced mostly back to dynamic elements (Python scripts) loading
- Solution: statically link scripts

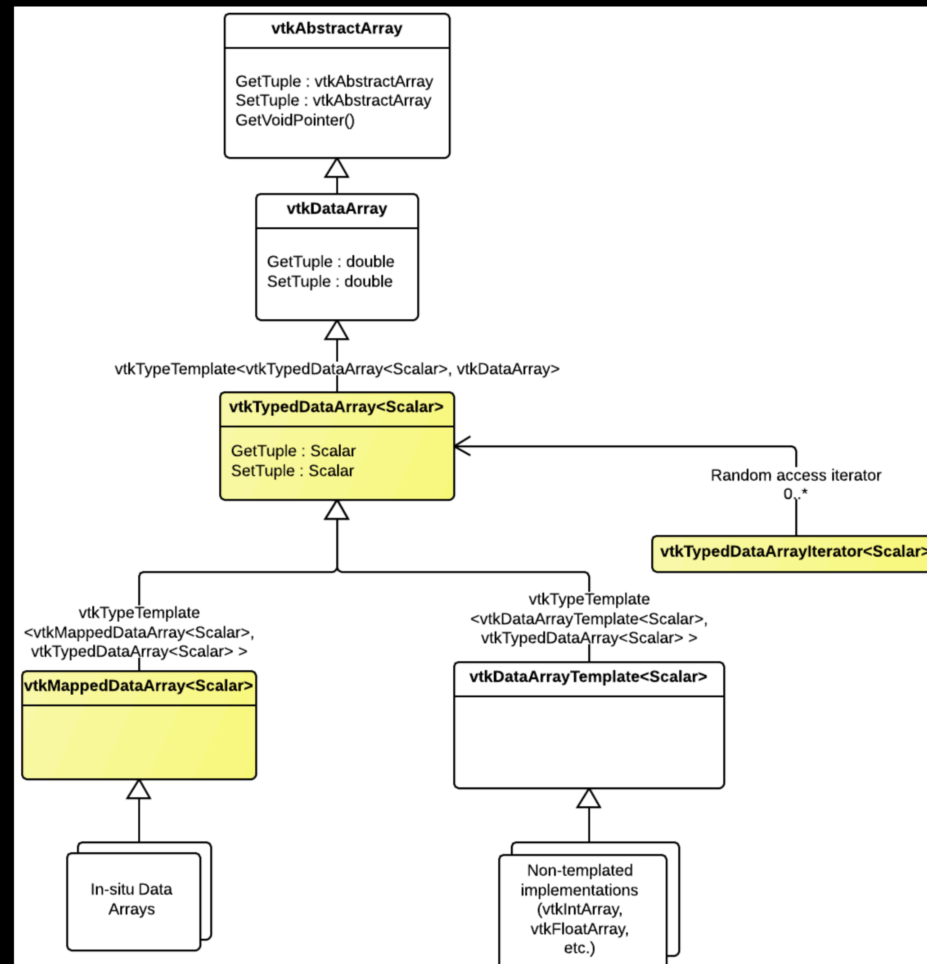
In Situ Library Challenges: Footprint

- Catalyst sits on libraries comprising over 4 million lines of code (most in VTK) developed for more than 2 decades.
- To support script interpreters, all code that *might* be accessed must be loaded.
  - Leads to big library files and program text sections.
- Solution: load subsets of code.
  - VTK repository modularized. Smaller libraries with clearer dependencies.
  - Provide Catalyst “Editions.”
    - Collections of functionality common to certain usage.
    - Makes it easier to get useful subset.

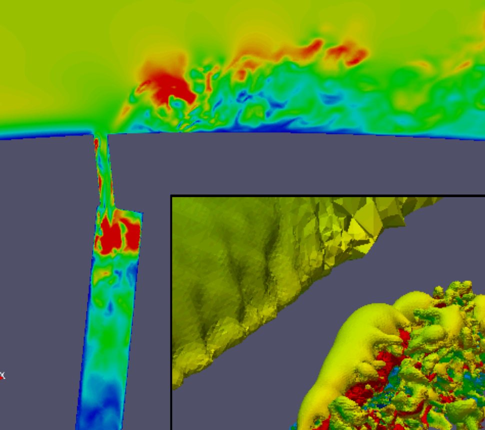
In Situ Library Challenges: Footprint

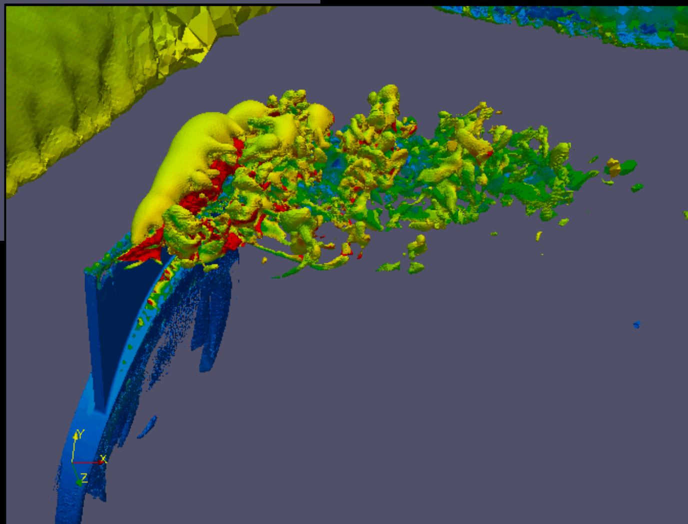
Sandia National Laboratories

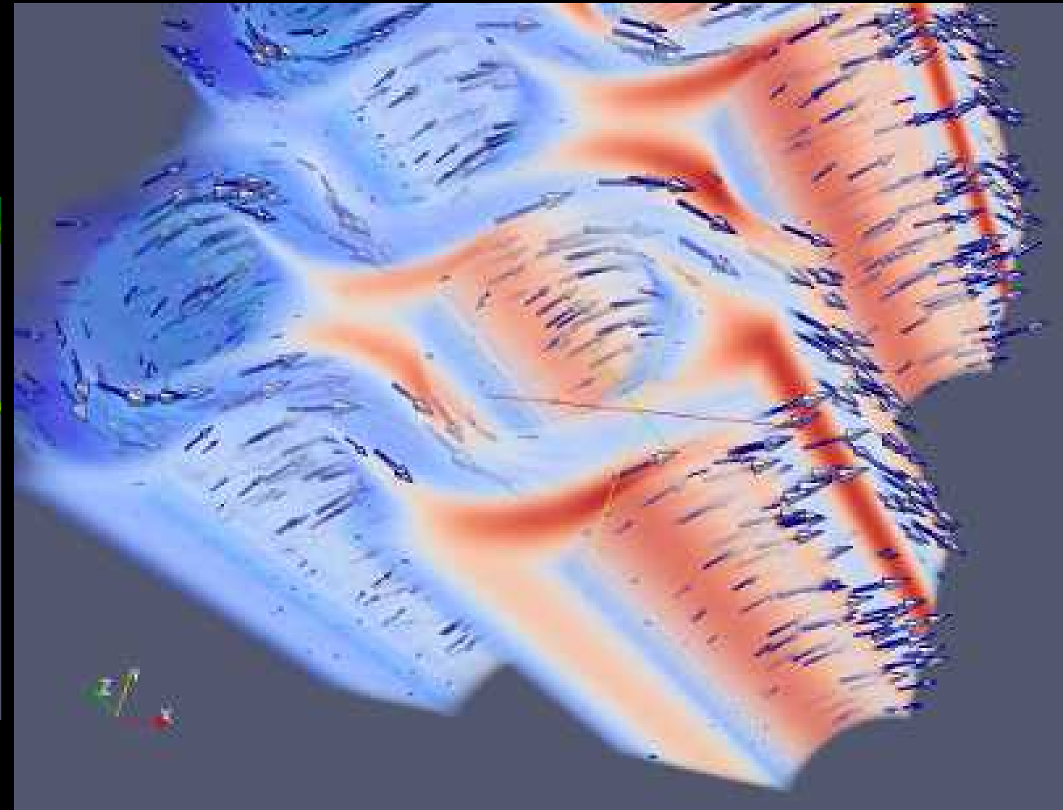
- Catalyst uses VTK for algorithm implementation.
  - VTK expects a particular data layout in memory.
  - If simulation uses a different layout, deep copy memory: double storage.
- Recent update of VTK data structure.
  - Support generic data layout through virtual method interface.

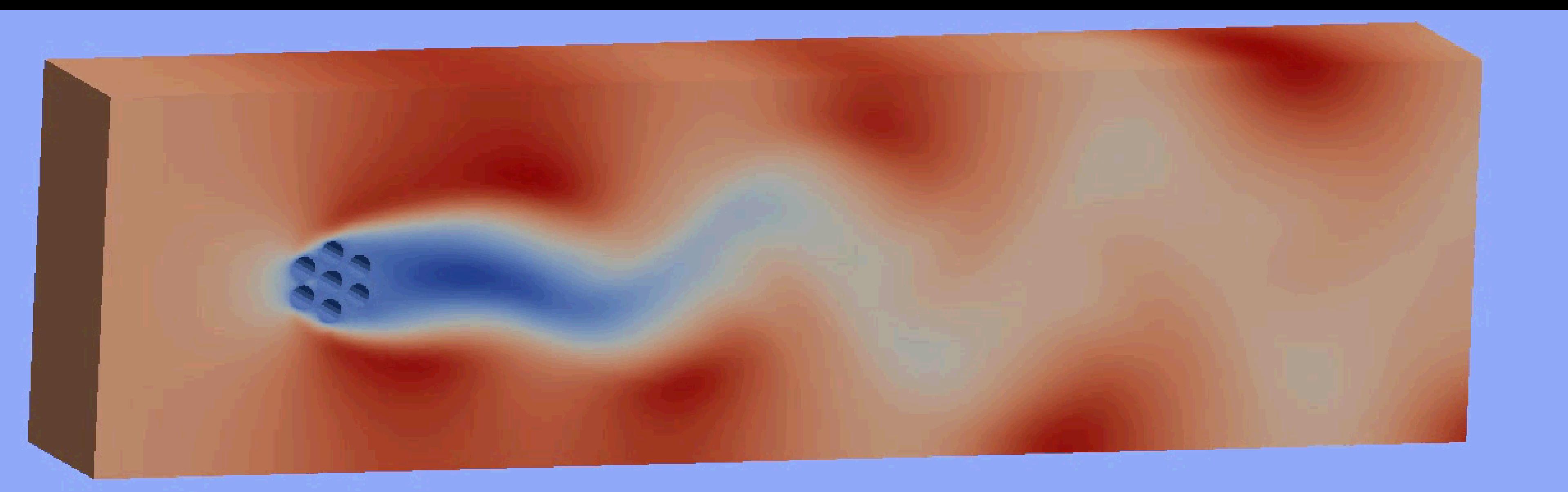


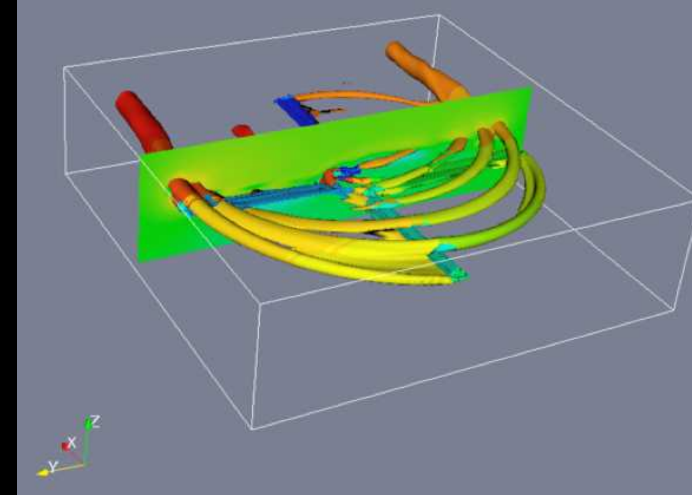
Lorendeau, Fournier, and Ribes. “In-Situ visualization in fluid mechanics using Catalyst: a case study for Code_Saturne.”

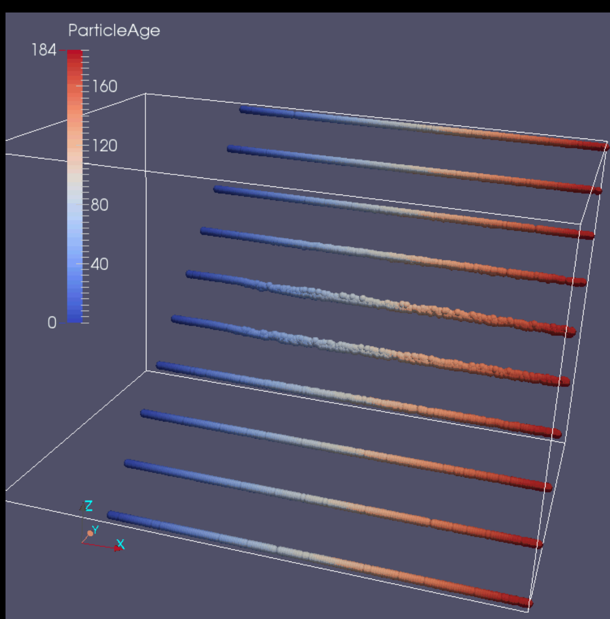


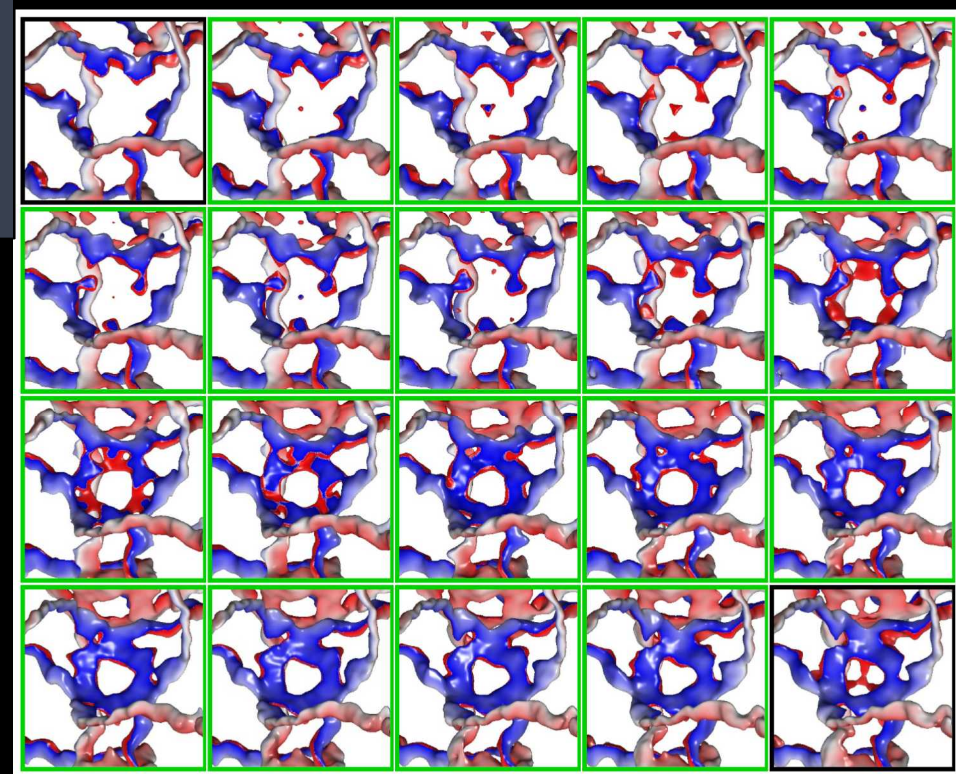












Ziegeler, Pettey, and Koop. “100K+ Core Chellenge for Comprehensive Computing at Scale.”

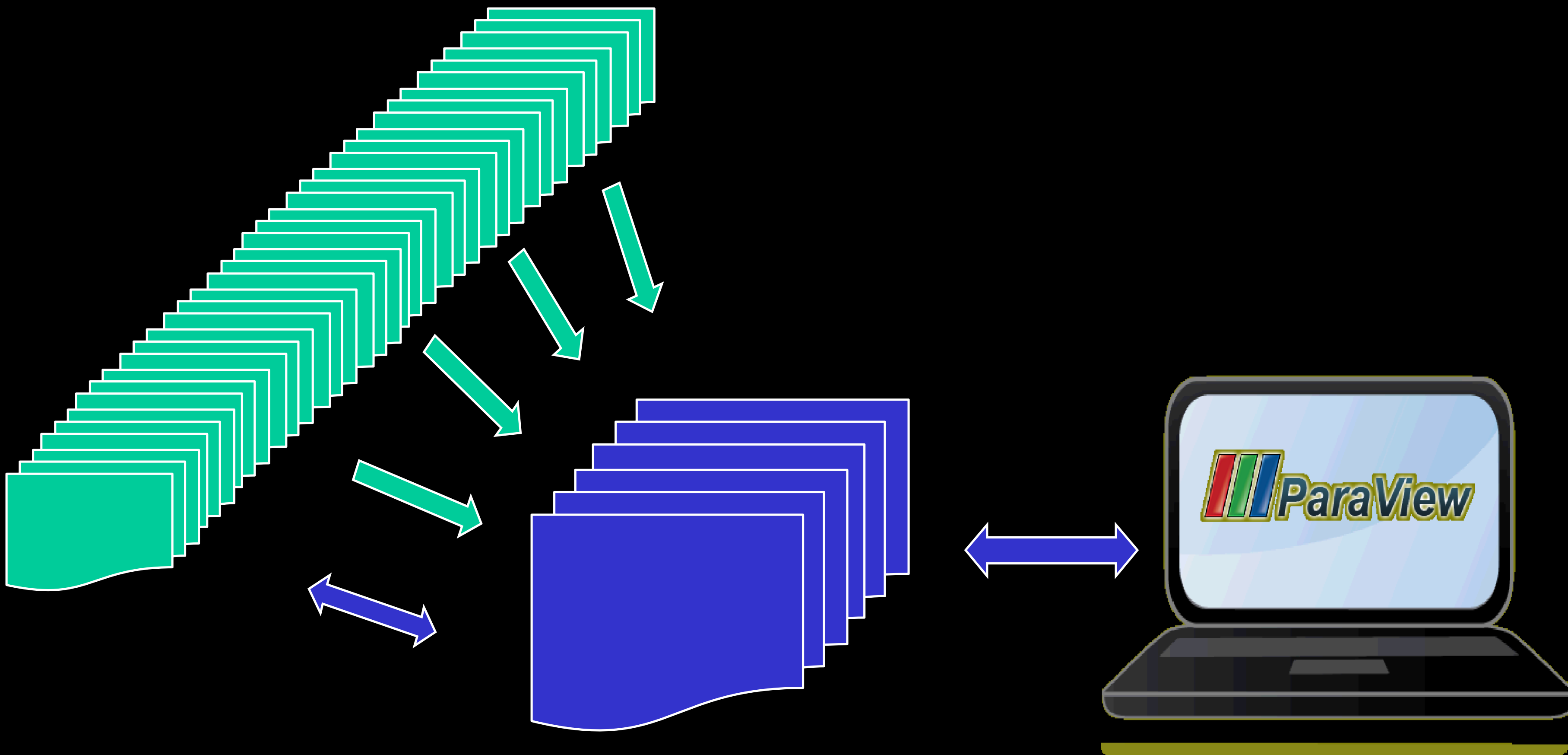
Source: Archive of Ridiculous Metaphors

Space of Solutions

Sandia National Laboratories

	Capability	Coupling	Footprint	Transfer	Interactive
Custom	Low	Tight	**Low**	**None**	No
ParaView In Situ, In Core	**High**	Tight	High	Possible memcpy	No
ParaView In Situ, Multi-Job	**High**	Tight	Medium	Subset Hi Speed Transfer	**Yes**
ParaView In Transit	**High**	**Loose**	~5% Extra Nodes	Hi Speed Transfer	**Yes**
File-based Post Process	**High**	**Loose**	**None**	Slow Disk, Archive Cost	**Yes**

Interactive Vis Service

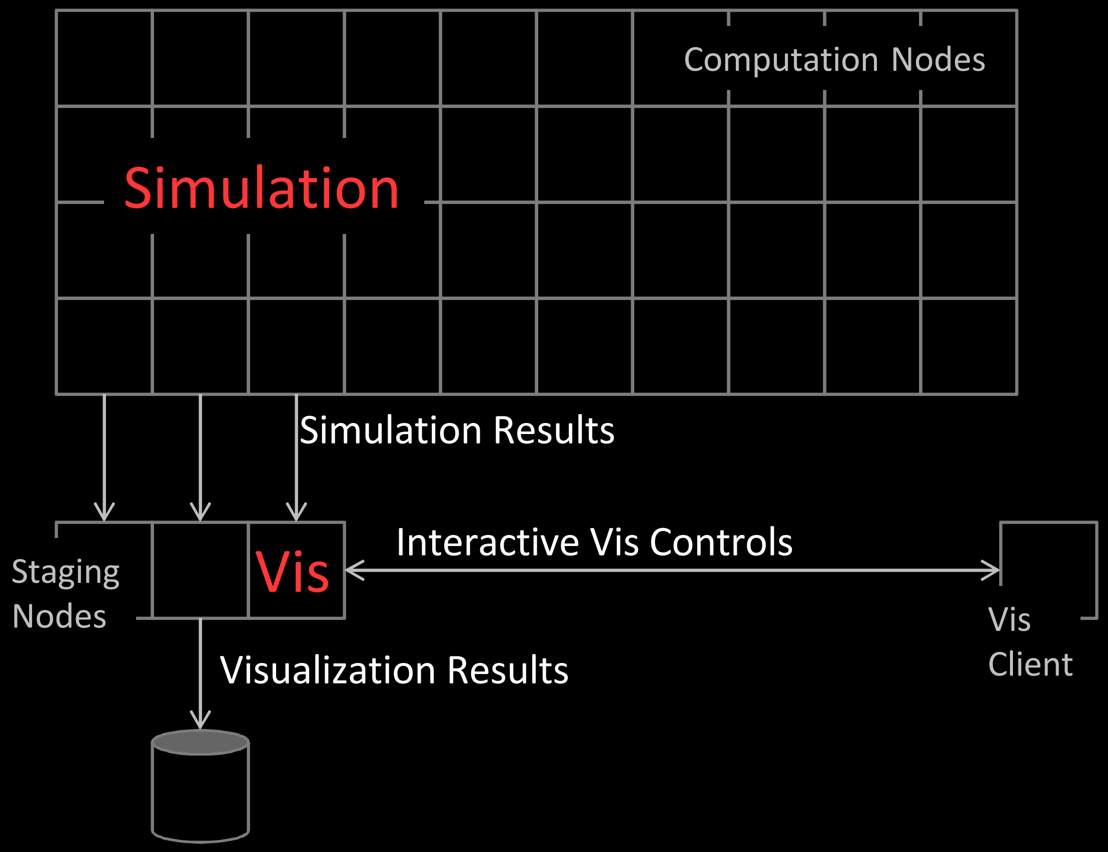


Full scale Simulation w/ In situ

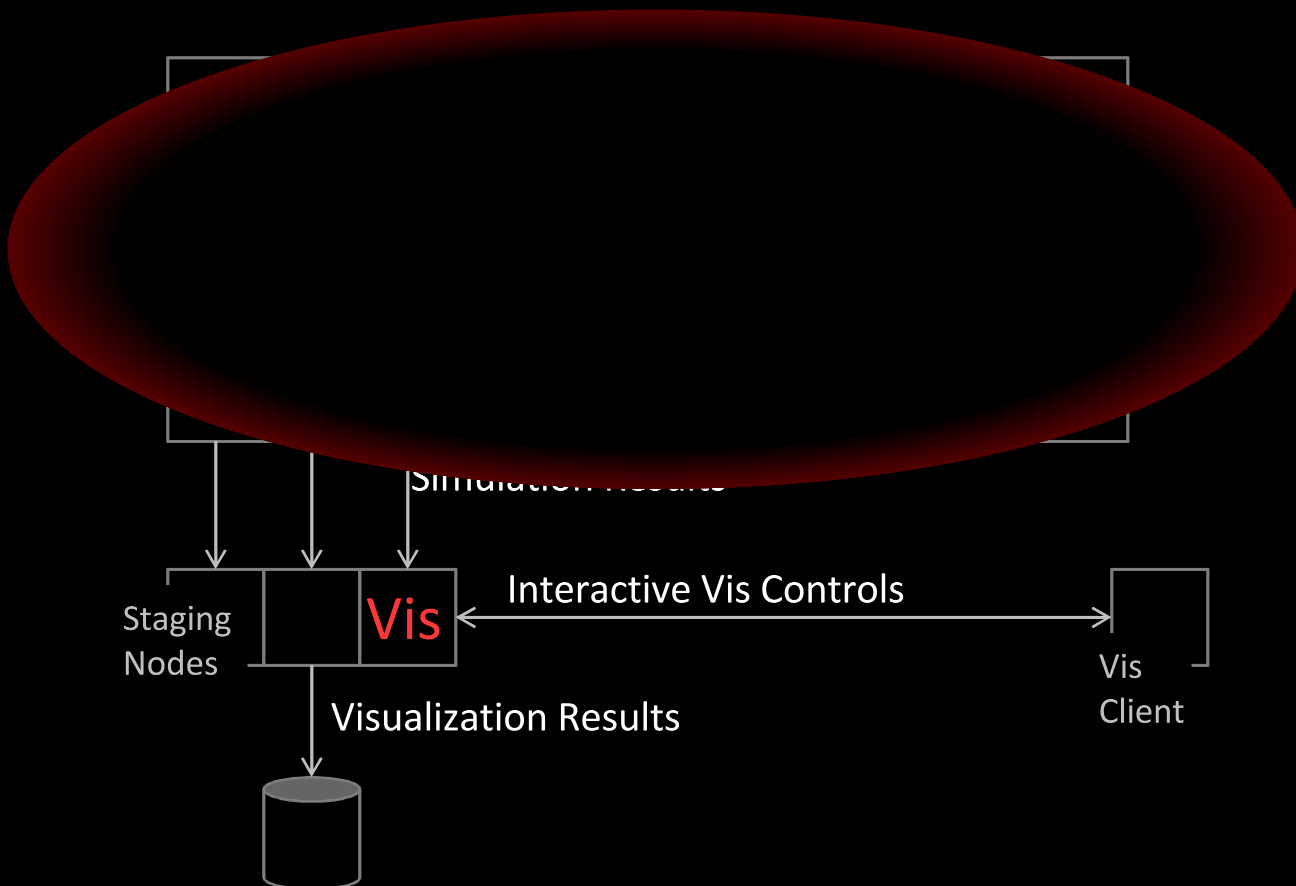
Smaller scale Covisualization

Interactive Client

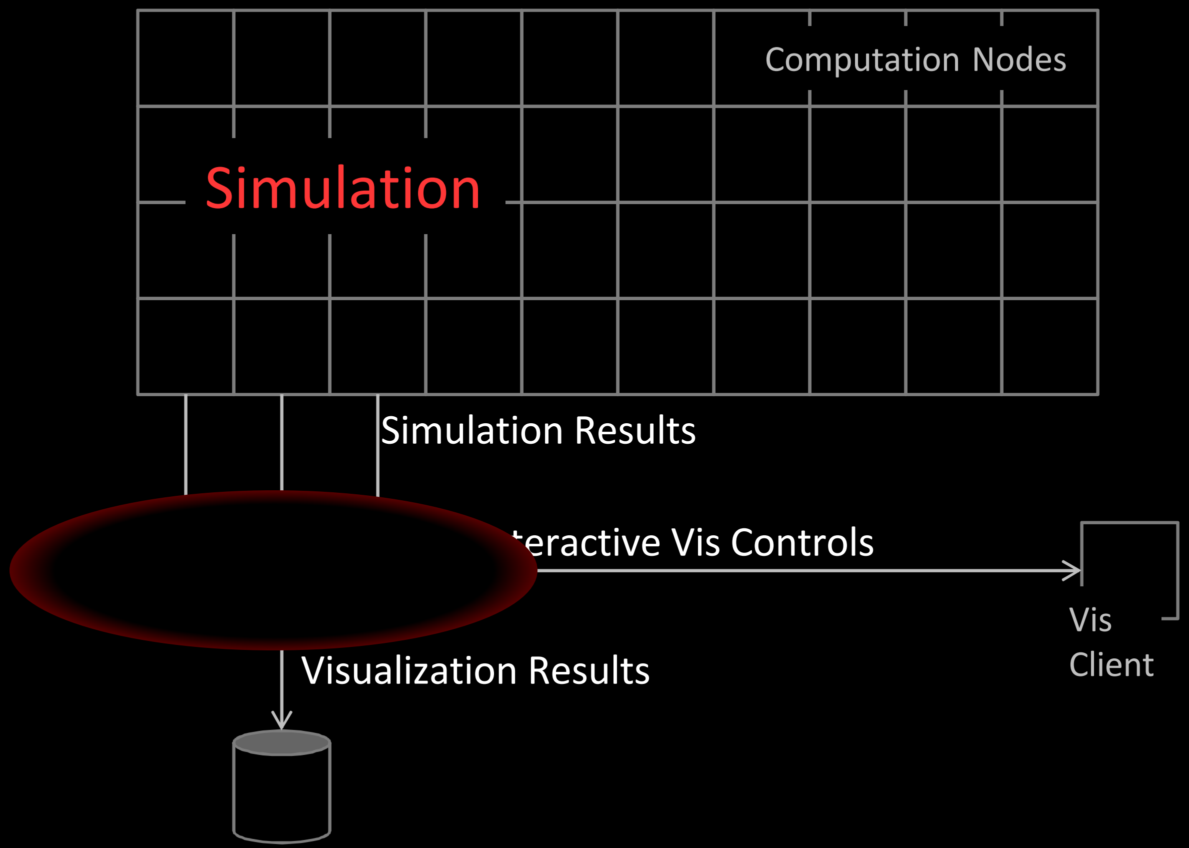
In Transit Visualization



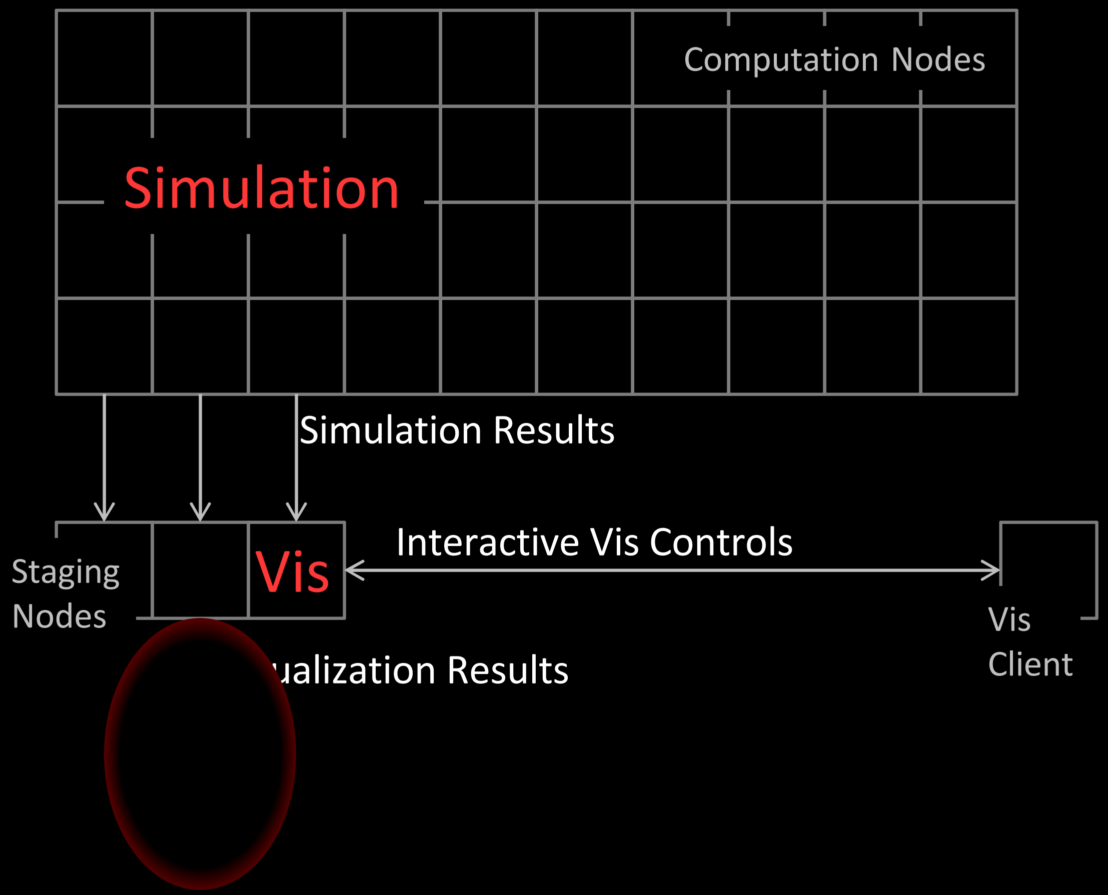
In Transit Visualization



In Transit Visualization

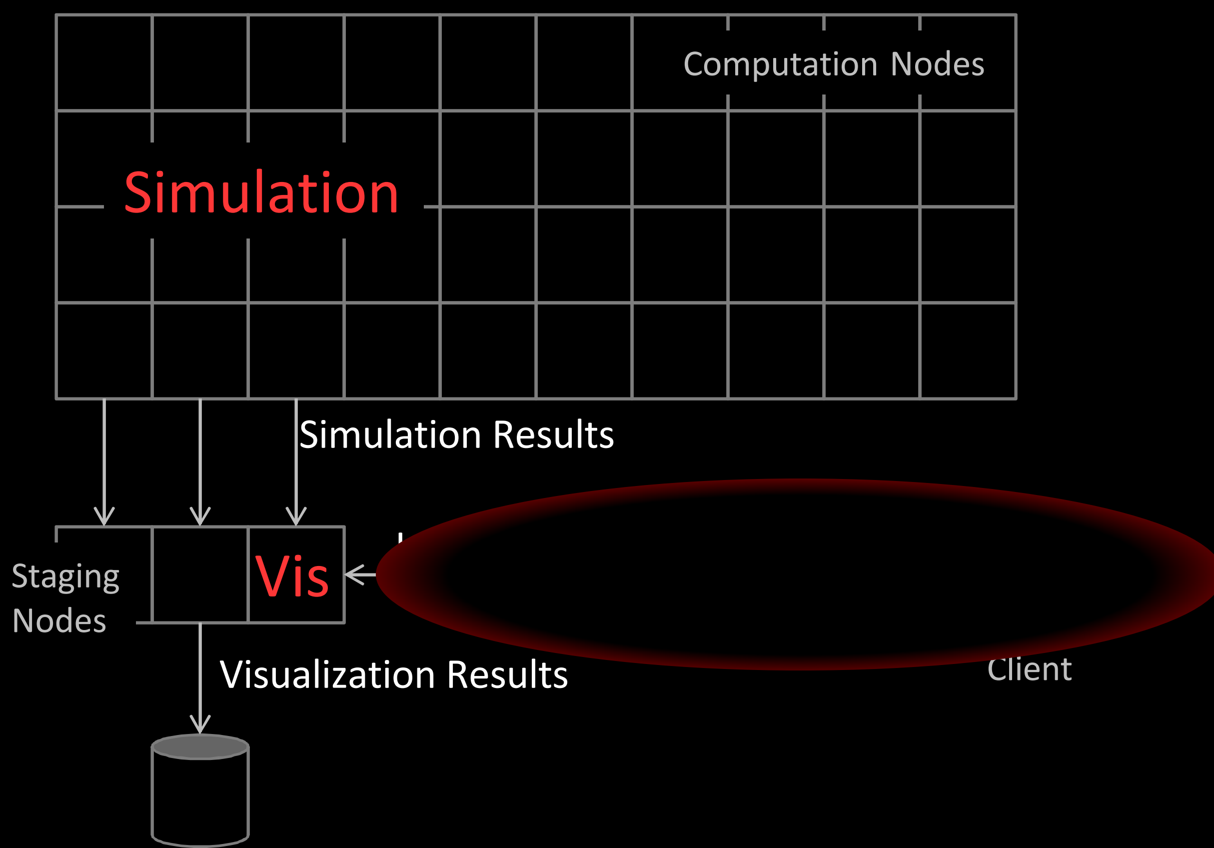


In Transit Visualization



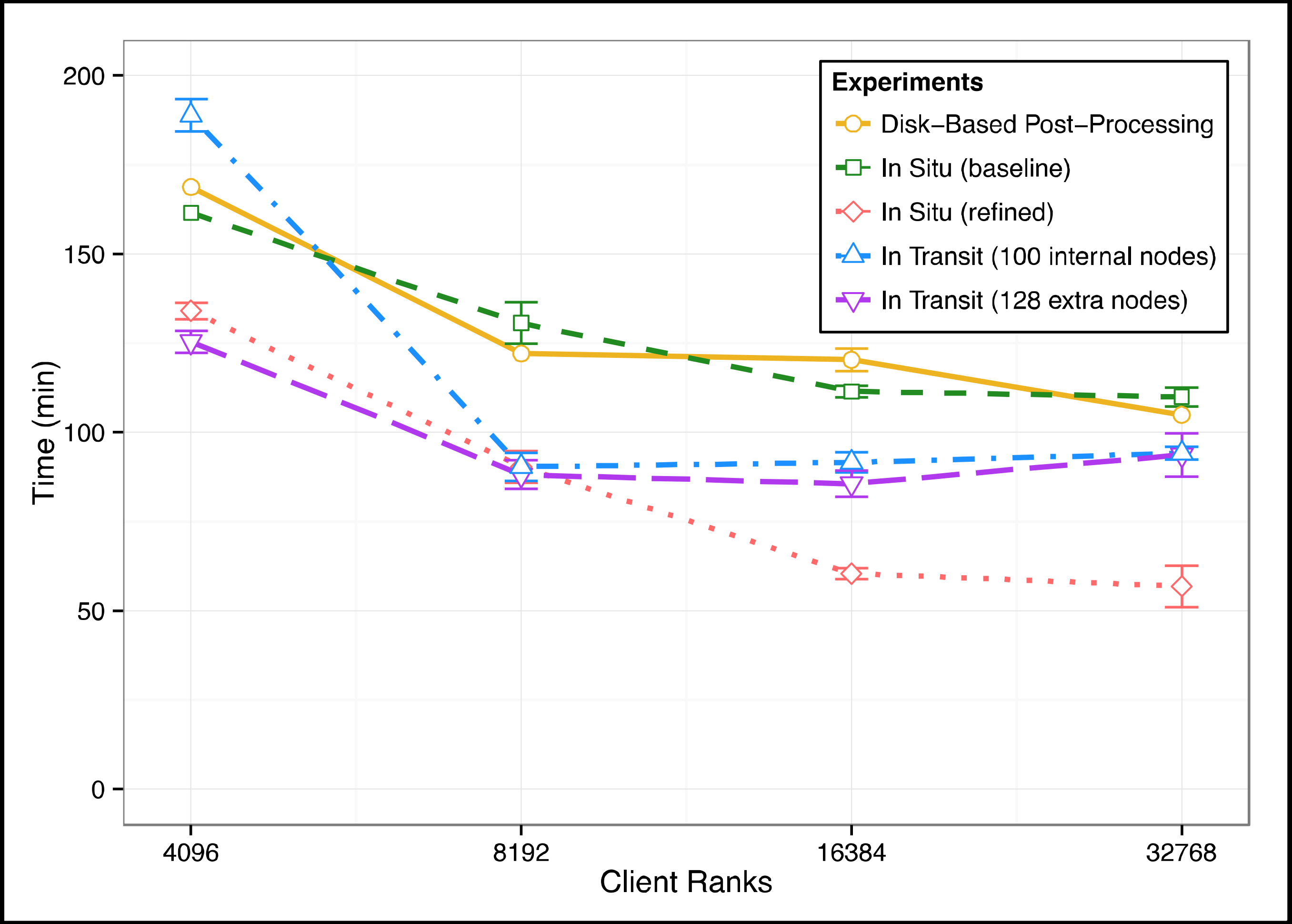
In Transit Visualization

Sandia National Laboratories

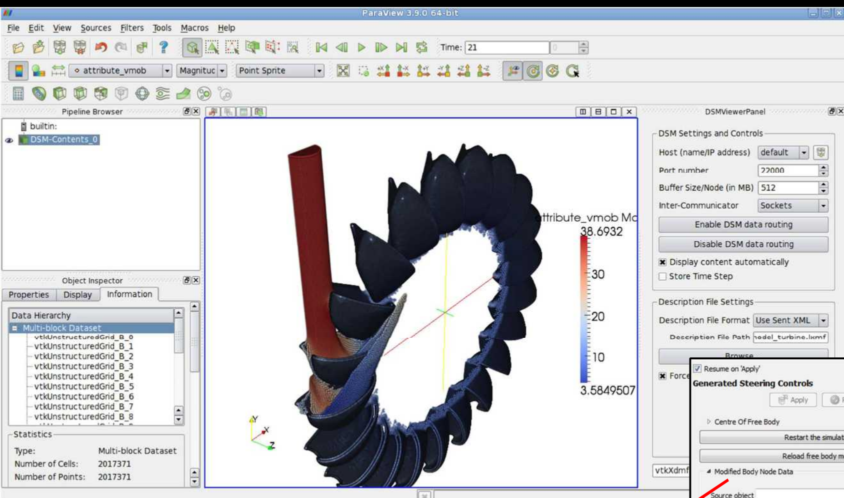


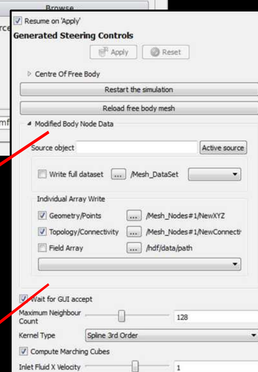
Efficiency of In Transit

- Although it requires large data transfers, in transit visualization can be more efficient than in situ visualization in some circumstances.

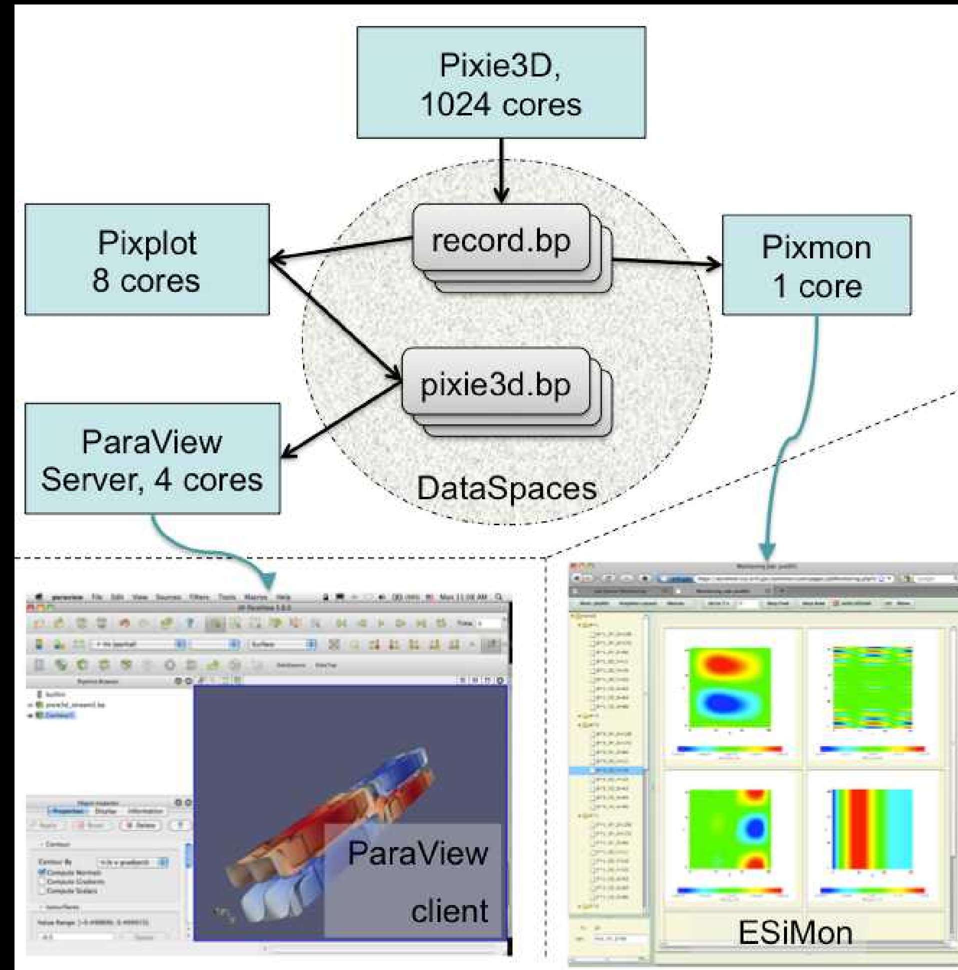


Generalizing In Transit






```
<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/NewC..."
   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.”

Conclusion

- In Situ visualization is a critical technology
  - As relative storage performance drops, eventually impossible to store enough
  - It’s not a problem until it’s a really big problem
- In Situ integration is messy
  - Integration of code can cause contention
  - One size doesn’t fit all
- Reusable production tools are important
  - It’s not reasonable or feasible to reinvent the wheel every time

Acknowledgements

- 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.
- Additional support by the Director, Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. 12-015215, through the Scientific Discovery through Advanced Computing (SciDAC) Institute of Scalable Data Management, Analysis and Visualization.