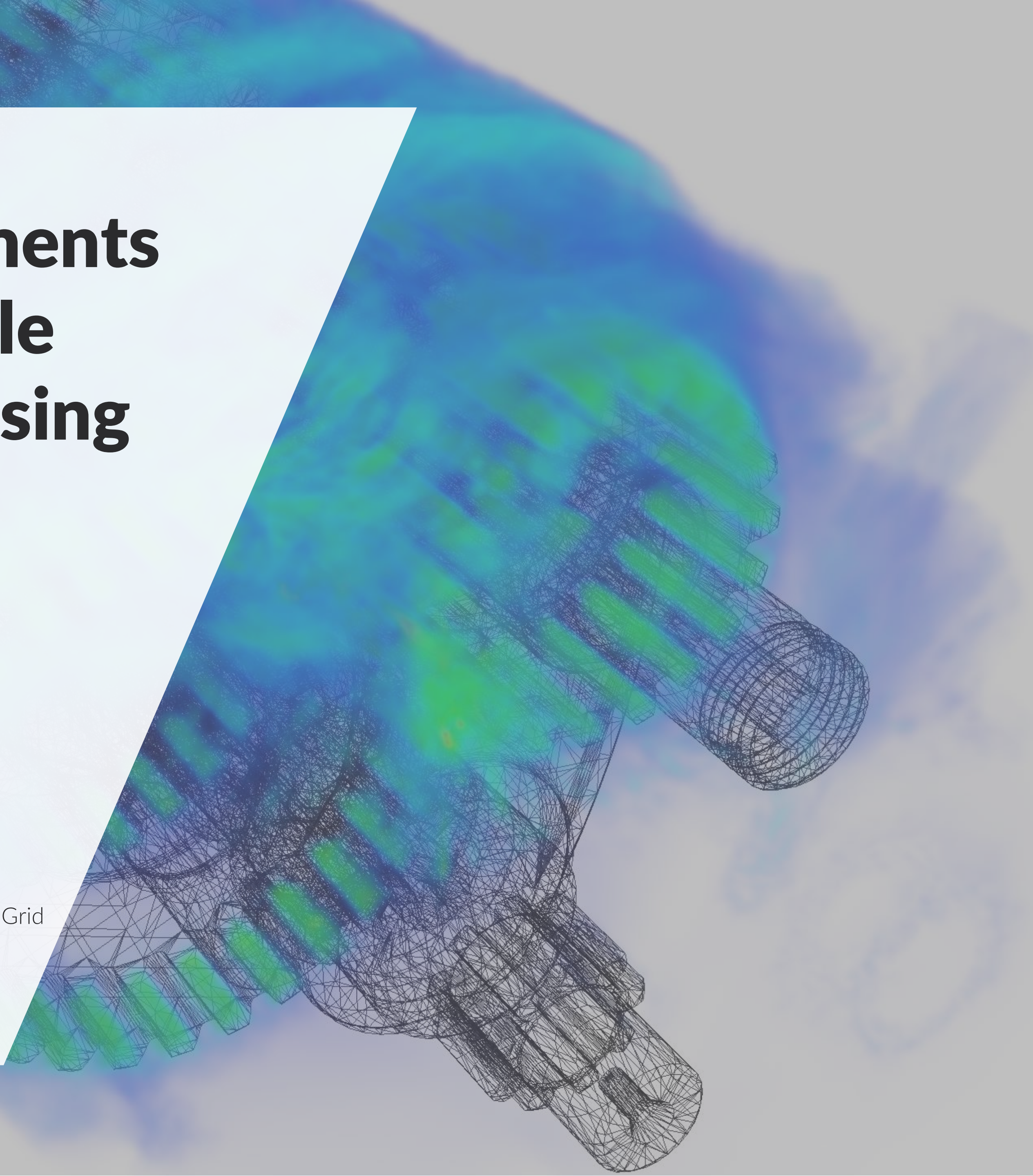
I/O Requirements for Exascale Post-Processing

Kitware

>25.09.2017

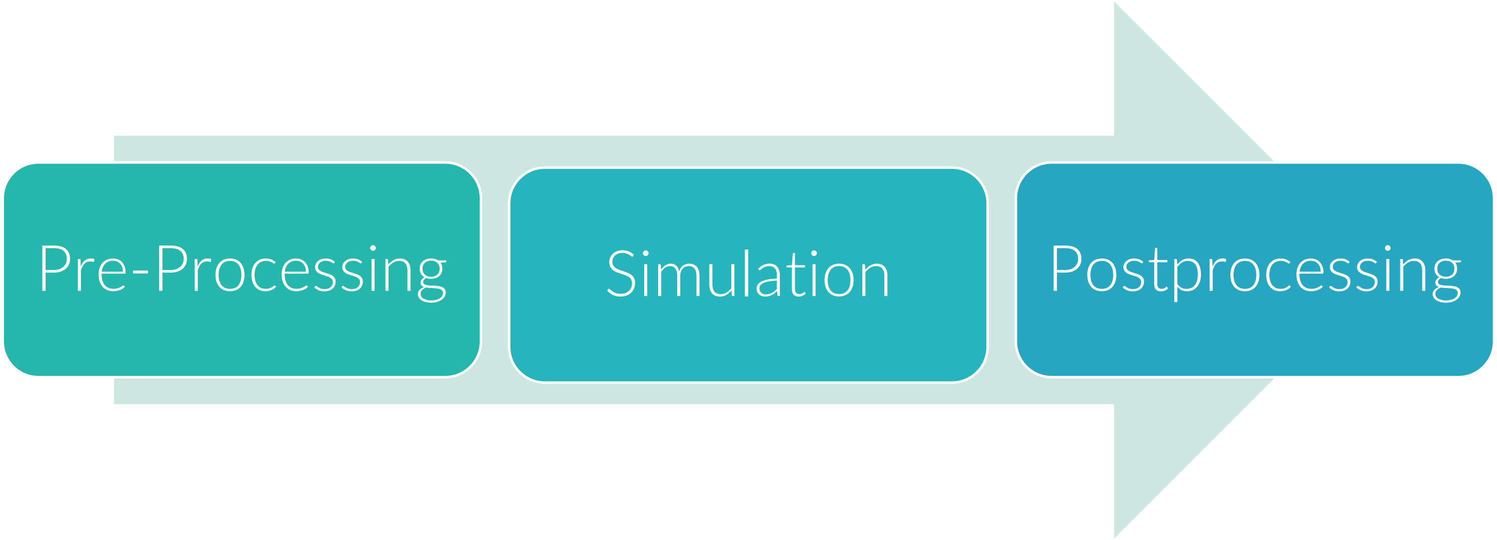
Workshop on Exascale I/O for Unstructured Grid
Hamburg, Germany

Julien Jomier – julien.jomier@kitware.com



>A WORLD WITHOUT (I/O) LIMITATIONS

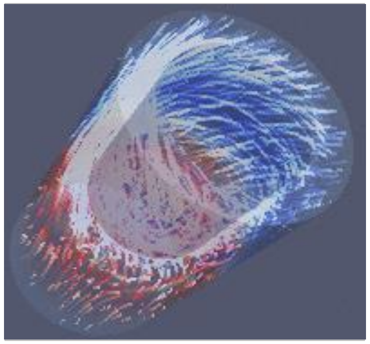
>>Let’s imagine



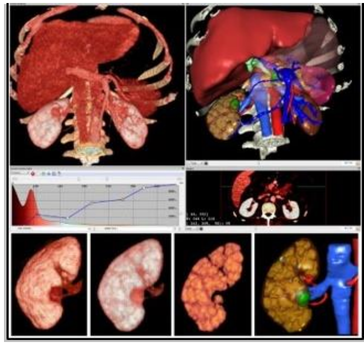
Processing **any data size, instantaneously** and **interactively**

> VTK

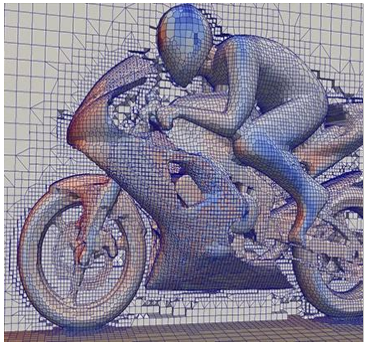
>>Open-Source Library for Scientific Visualization

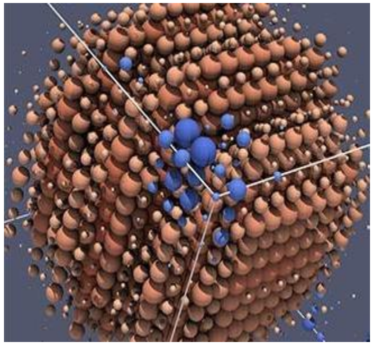


CROSS-PLATFORM VISUALIZATION LIBRARY



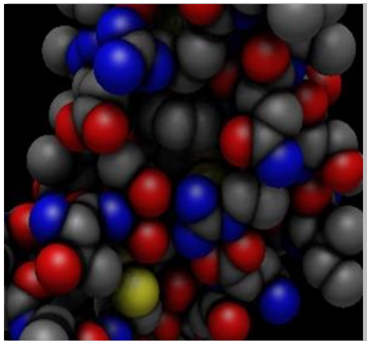
>> Visualization tools for scientific datasets
>> Advanced algorithms for data analysis
>> 3D Interactive Widgets





>> 25 years of development

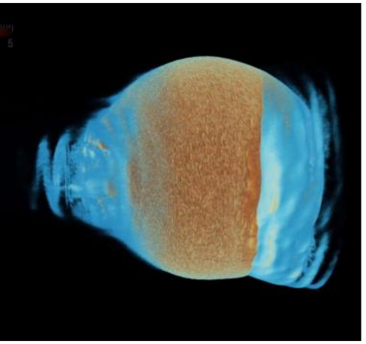
>> BSD License

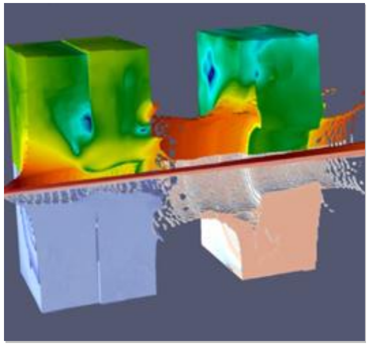


www.vtk.org

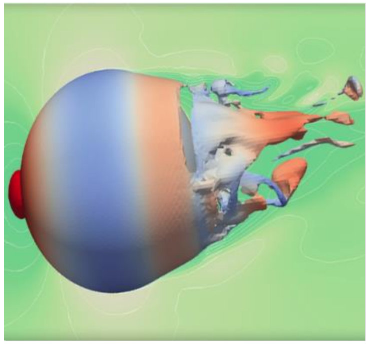
> ParaView

>> High performance Post-Processing





**CROSS-PLATFORM
OPEN-SOURCE
POST-PROCESSING
APPLICATION**

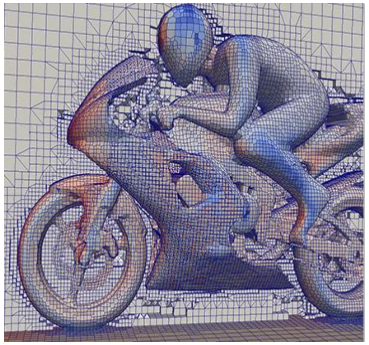


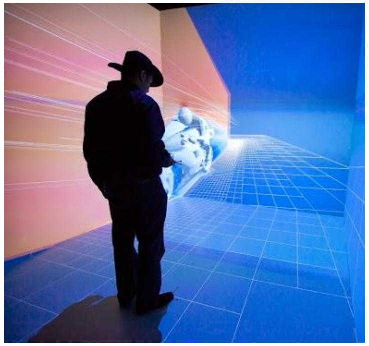
>> Parallel Visualization

>> Advanced post-processing algorithms

>> 3D Interactive widgets

>> In-situ processing

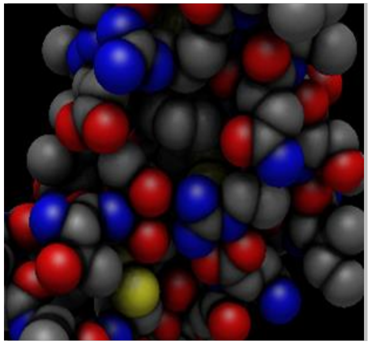




>> 20 years of development

>> Based on VTK

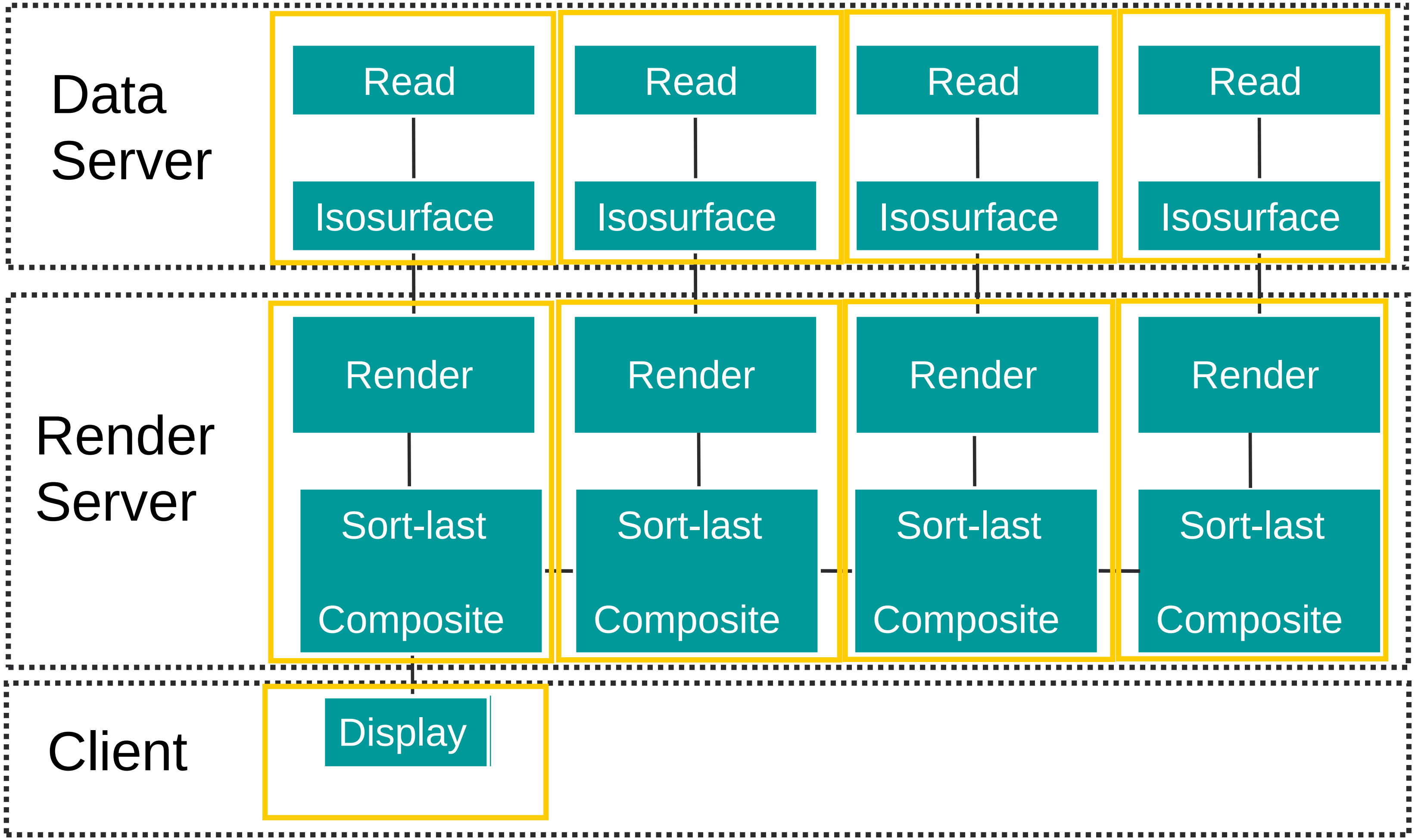
>> License BSD



www.paraview.org

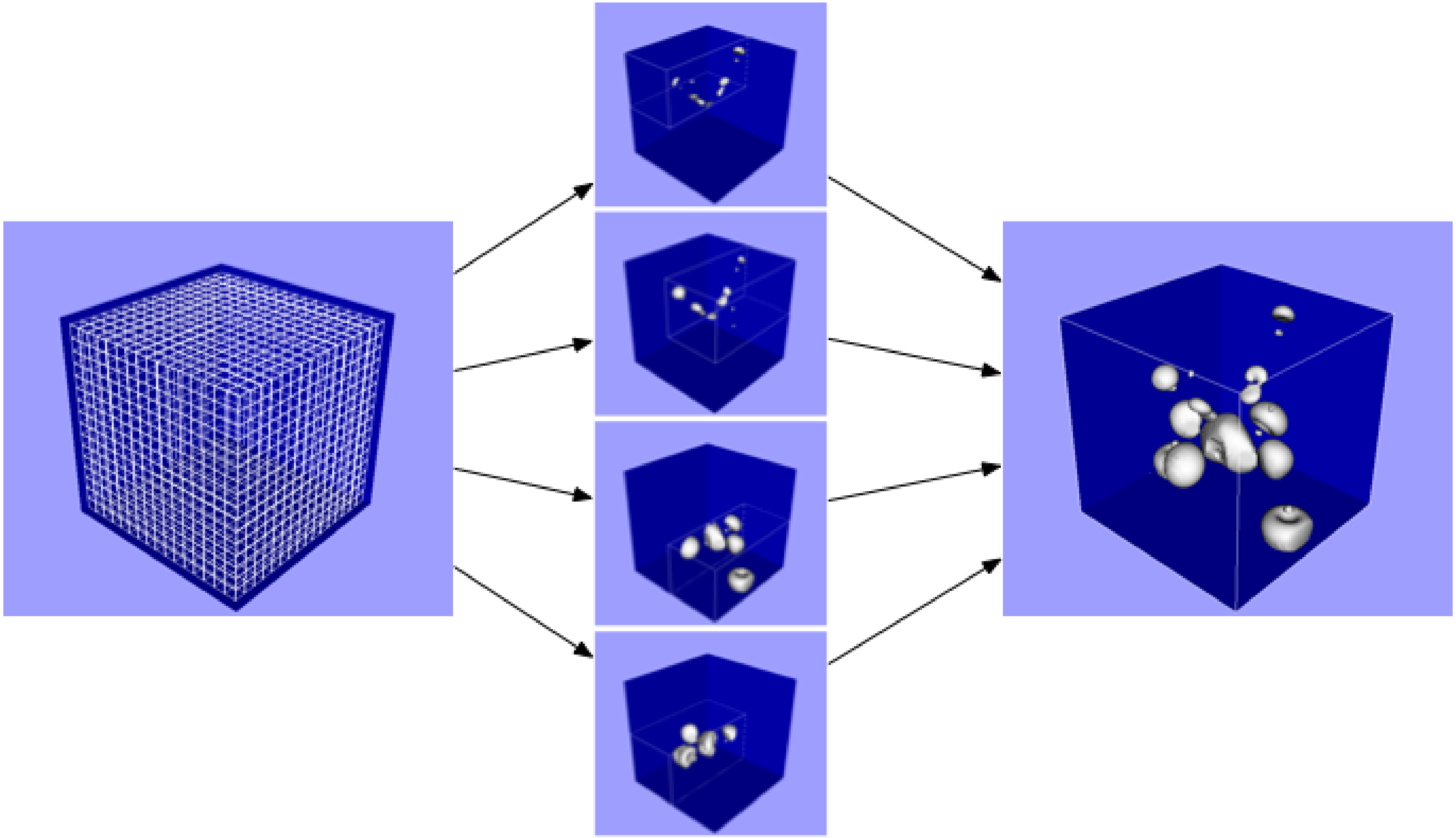
>DATA PARALLEL PIPELINE

>>In ParaView



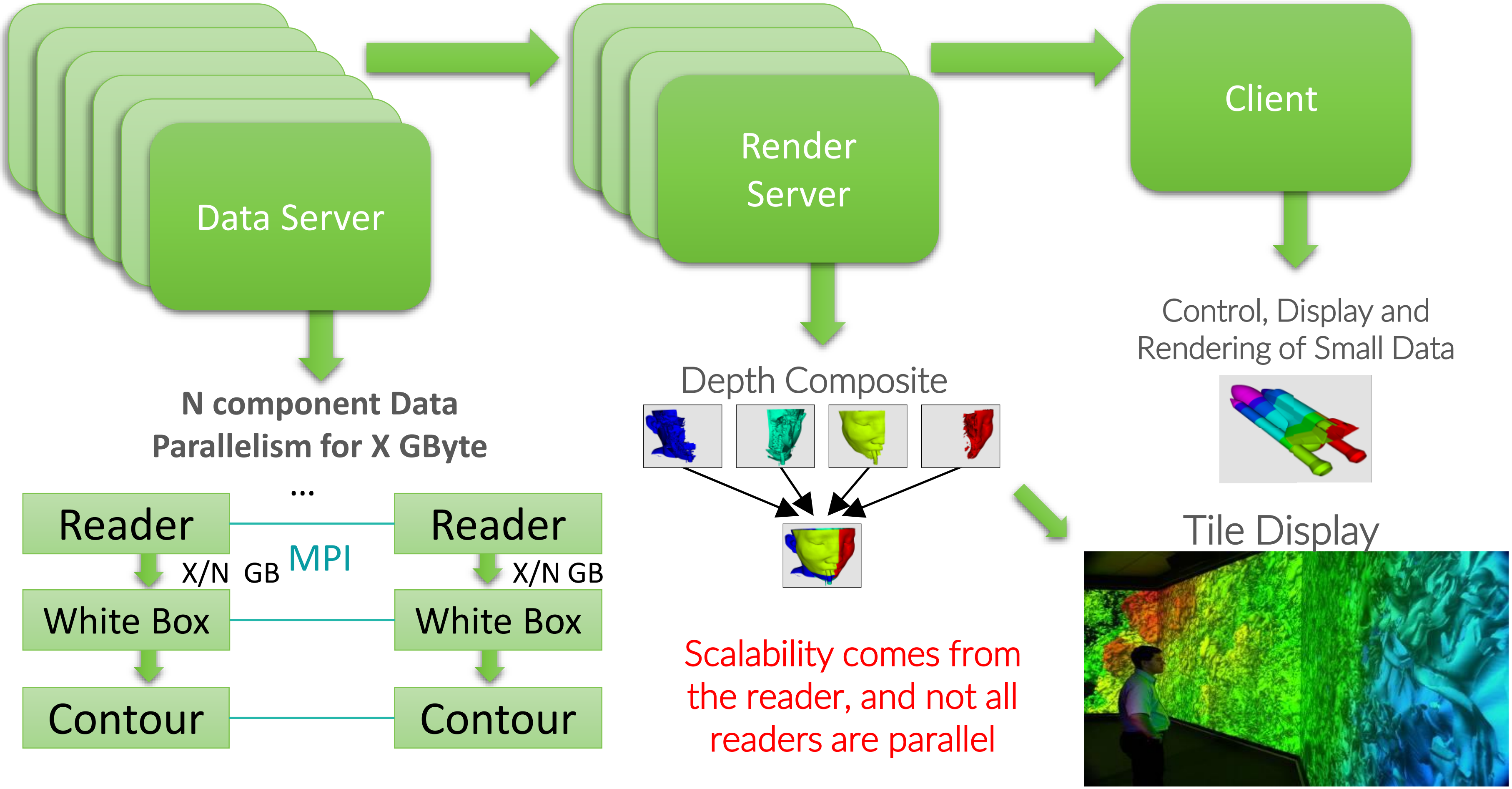
>PARALLEL PROCESSING

>>In ParaView



>PARAVIEW SCALABILITY

>>Large interactive post-processing



Data Server

Render Server

Client

N component Data
Parallelism for X GByte

…

Reader

X/N GB

MPI

Reader

X/N GB

White Box

White Box

Contour

Contour

Depth Composite

Control, Display and
Rendering of Small Data

Tile Display

Scalability comes from
the reader, and not all
readers are parallel

>PARALLEL I/O FORMATS

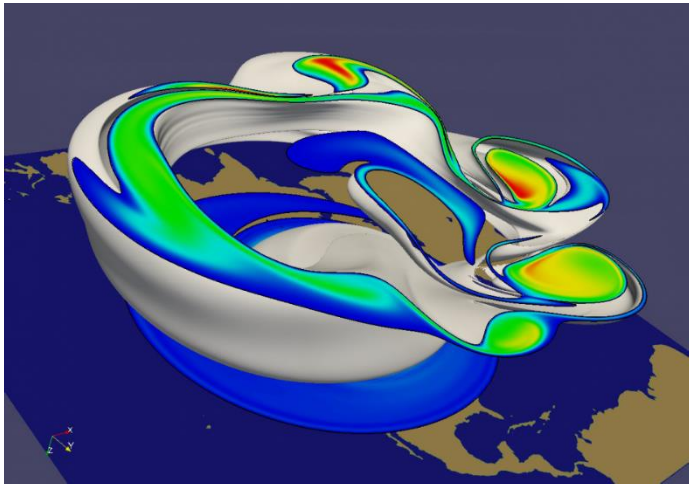
>> In VTK and ParaView

- ADIOS
- HDF5
- Exodus
- PVD
- Parallel VTK
- Ensight
- SPC TH
- LSDyna
- NetCDF

>POSTPROCESSING CHALLENGES

>>Interactive Visualization

- Exploratory visualization: no apriori knowledge on what to visualize
- Division of the data: simulation vs. rendering
- Spatial vs. Temporal data
- Bound by the slowest renderer
- Post-processing algorithm in parallel
- Exporting post-processing



>100K+ CORE CHALLENGE

>> DoD High Performance Modernization Program

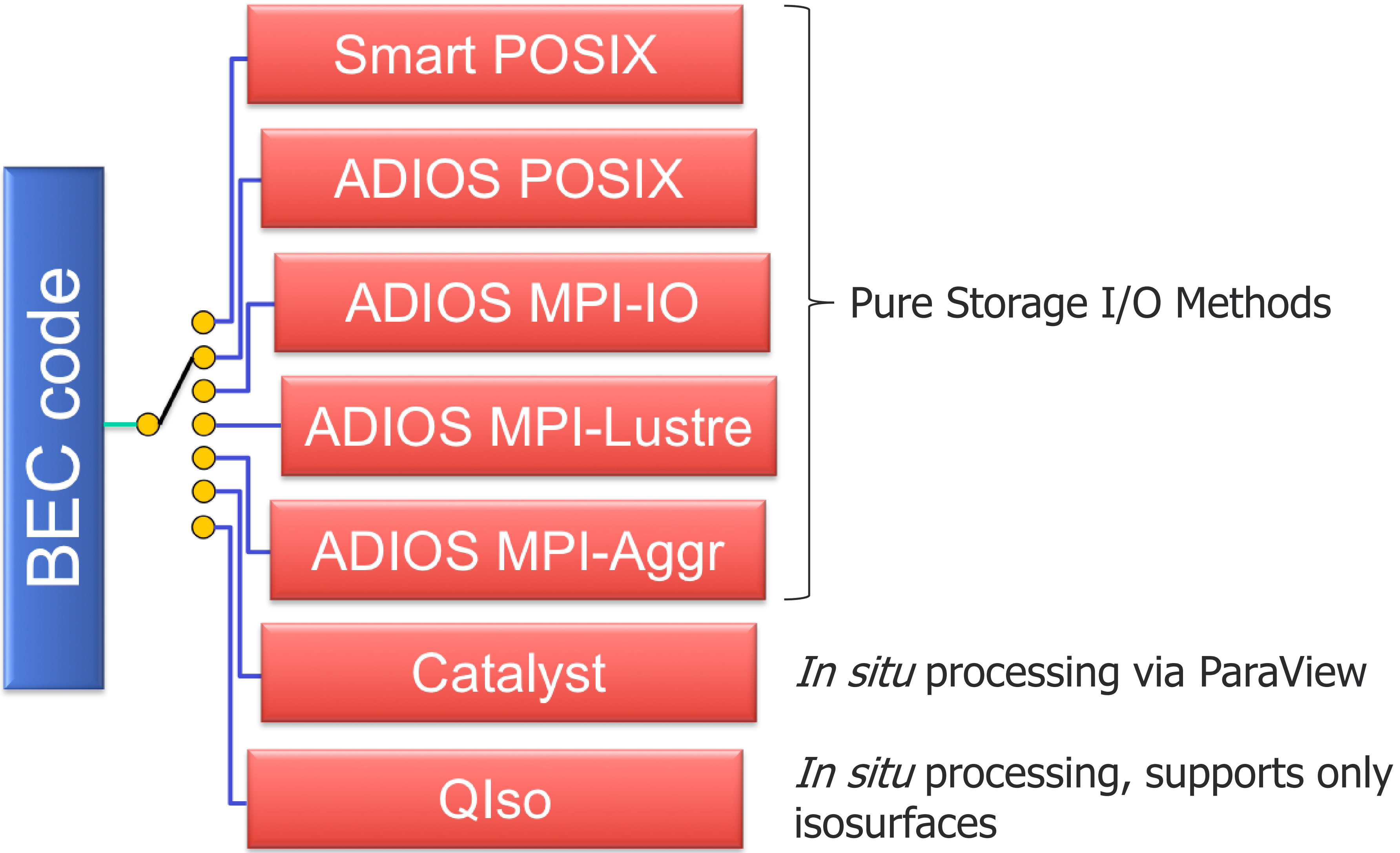
"Garnet"
DoD High Performance Modernization Program (HPCMP)
Engineer Research and Development Center (ERDC)
Cray XE6: 150,592 Compute Cores

Pioneer Project:

- Exercise the system at entire scale (all 150K Cores)
- Enable science that can only be accomplished at that scale
- All-encompassing scaling: computation, communication and **I/O**

>BENCHMARKING I/O OPTIONS

>> 100K+ Core challenge



BEC code

Smart POSIX

ADIOS POSIX

ADIOS MPI-IO

ADIOS MPI-Lustre

ADIOS MPI-Aggr

Catalyst

QIso

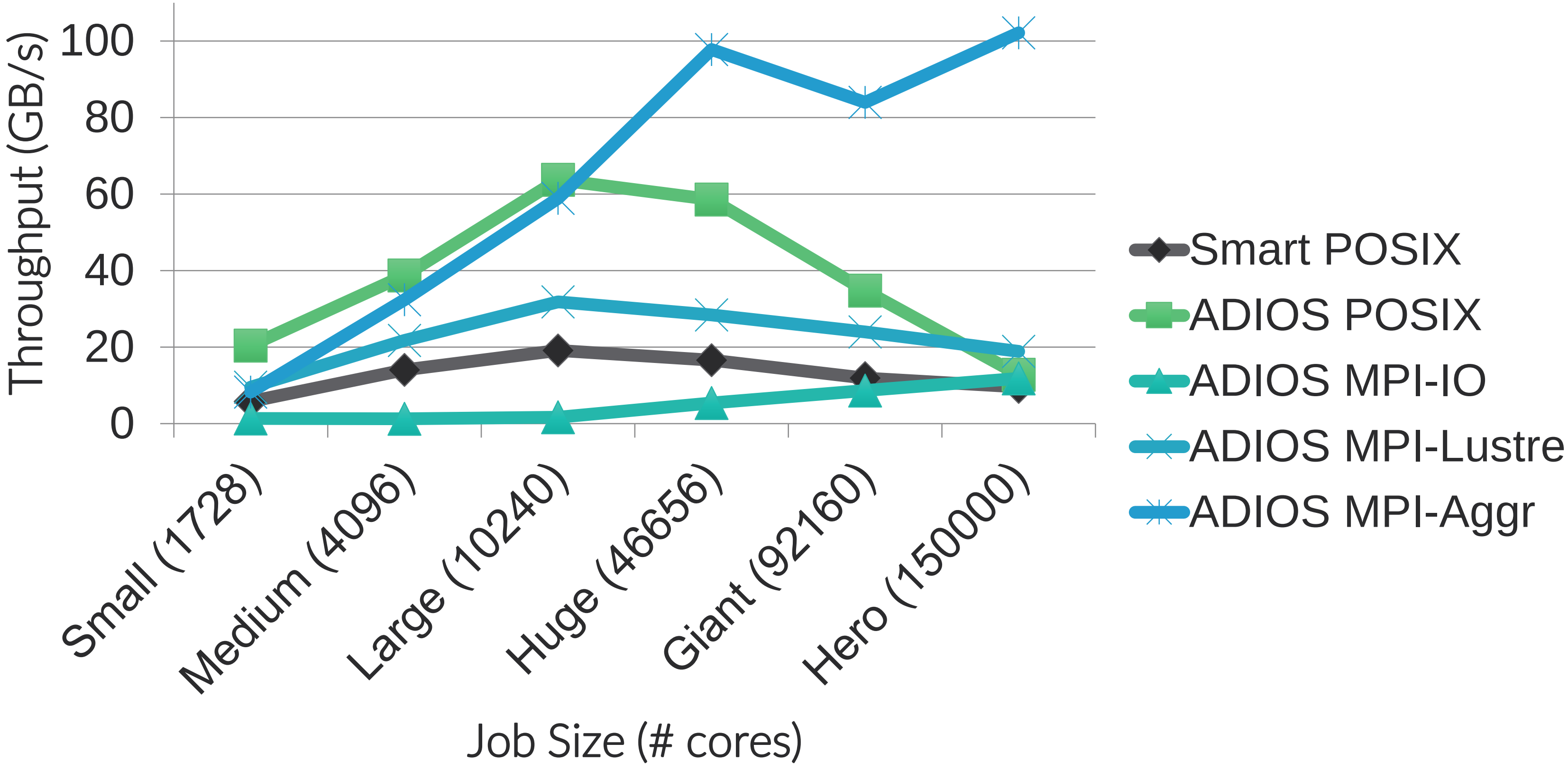
Pure Storage I/O Methods

In situ processing via ParaView

In situ processing, supports only isosurfaces

> BENCHMARK : I/O ONLY

>>100K+ Core challenge



- 2x-10x Improvement in I/O
  - depending upon scale and methods involved
- **I/O is still a major bottleneck**

>CHANGING FACE OF HPC

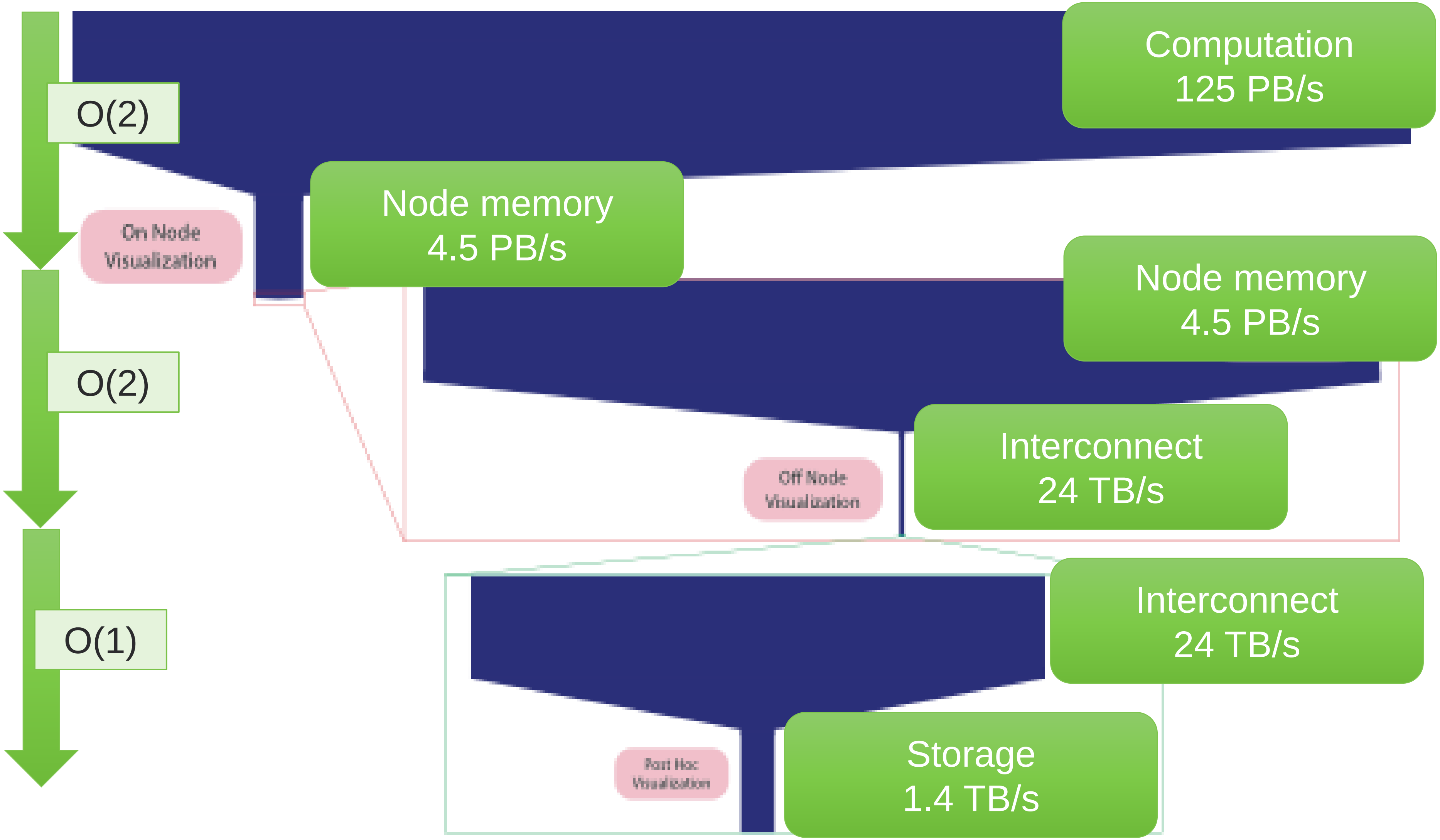
>> Oak Ridge National Laboratory

Attribute	Titan	Summit - 2018
Nodes	18,688	~4,600
Node Performance	1.4 TF	> 40 TF
Node Memory	38 GB	> 512 GB
Processor	(1) 16-core AMD Opteron	(2) IBM POWER 9s
Accelerator	(1) NVIDIA Kepler	(6) NVIDIA Volta
CPU-GPU Interconnect	PCIe2	NVLINK (5-12x PCIe3)
System Interconnect	Gemini (6.4 GB/s)	Dual Rail EDR-IB (23 GB/s)
File System	Lustre 32 PB 1 TB/s	GPFS 250 PB 2.5 TB/s
Peak Power	9 MW	15 MW

- Hierarchy of restricted/limited memory
- High latency and/or low bandwidth connections for non-traditional components
- Interconnect link saturation by the number of connected computational units
- Requiring explicit description of the fine- and coarse-grain parallelism

>COMPUTE VS I/O ON TITAN

>> Oak Ridge National Laboratory



O(2)

Computation
125 PB/s

Node memory
4.5 PB/s

On Node Visualization

O(2)

Node memory
4.5 PB/s

Interconnect
24 TB/s

Off Node Visualization

O(1)

Interconnect
24 TB/s

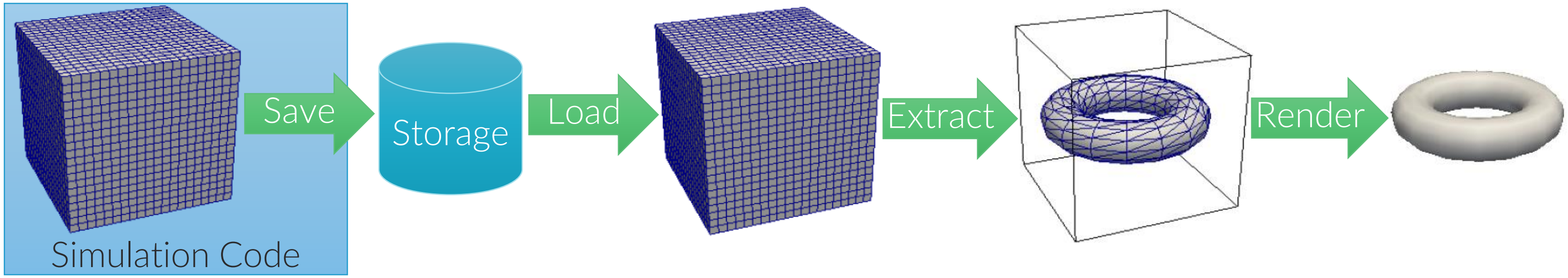
Storage
1.4 TB/s

Post Hoc Visualization

>DEALING WITH I/O BOTTLENECK

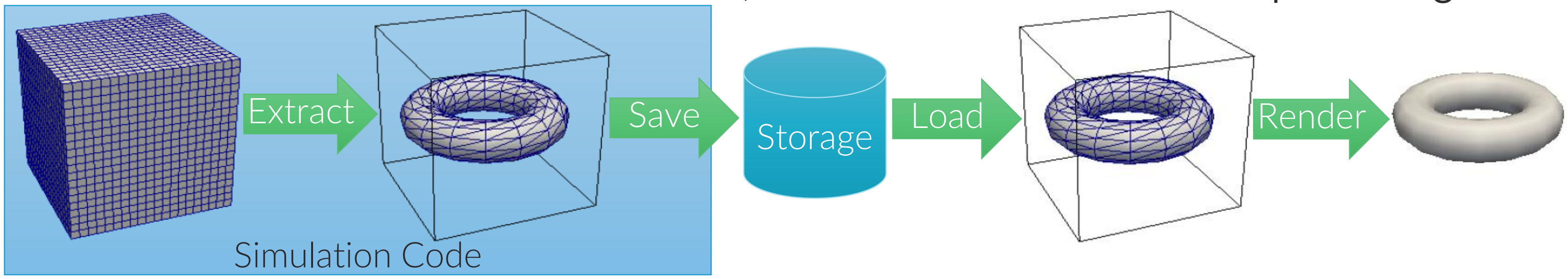
>>In-situ processing and rendering

Traditional Data I/O – No, *in situ*, save all data for future processing



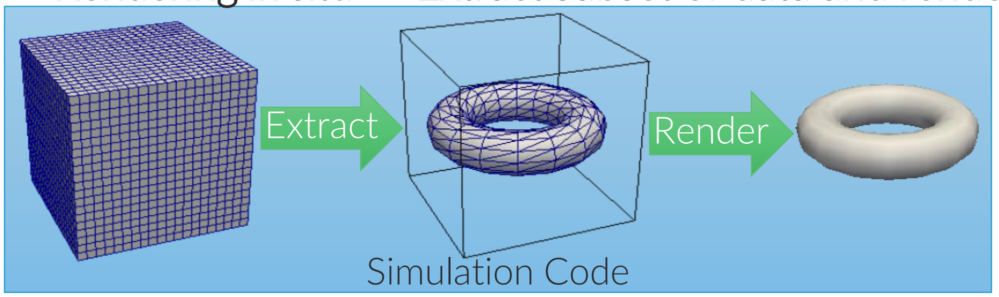
Save
Storage
Load
Extract
Render
Simulation Code

“Extraction *In situ*” – Extract subset of data, save the extraction for future processing



Extract
Save
Storage
Load
Render
Simulation Code

“Rendering *In situ*” – Extract subset of data and render immediately



Extract
Render
Simulation Code

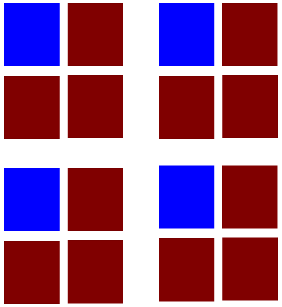
>WHAT IS IN-SITU VISUALIZATION

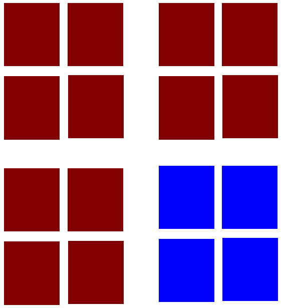
>> In-situ visualization

In Situ

- Many flavors/terms: tightly coupled, loosely coupled, in transit, co-processing, etc.
- Practical view: anything processed but not written to persistent storage is *in situ*

In situ – *no data movement*: Simulation and *in situ* methods share memory





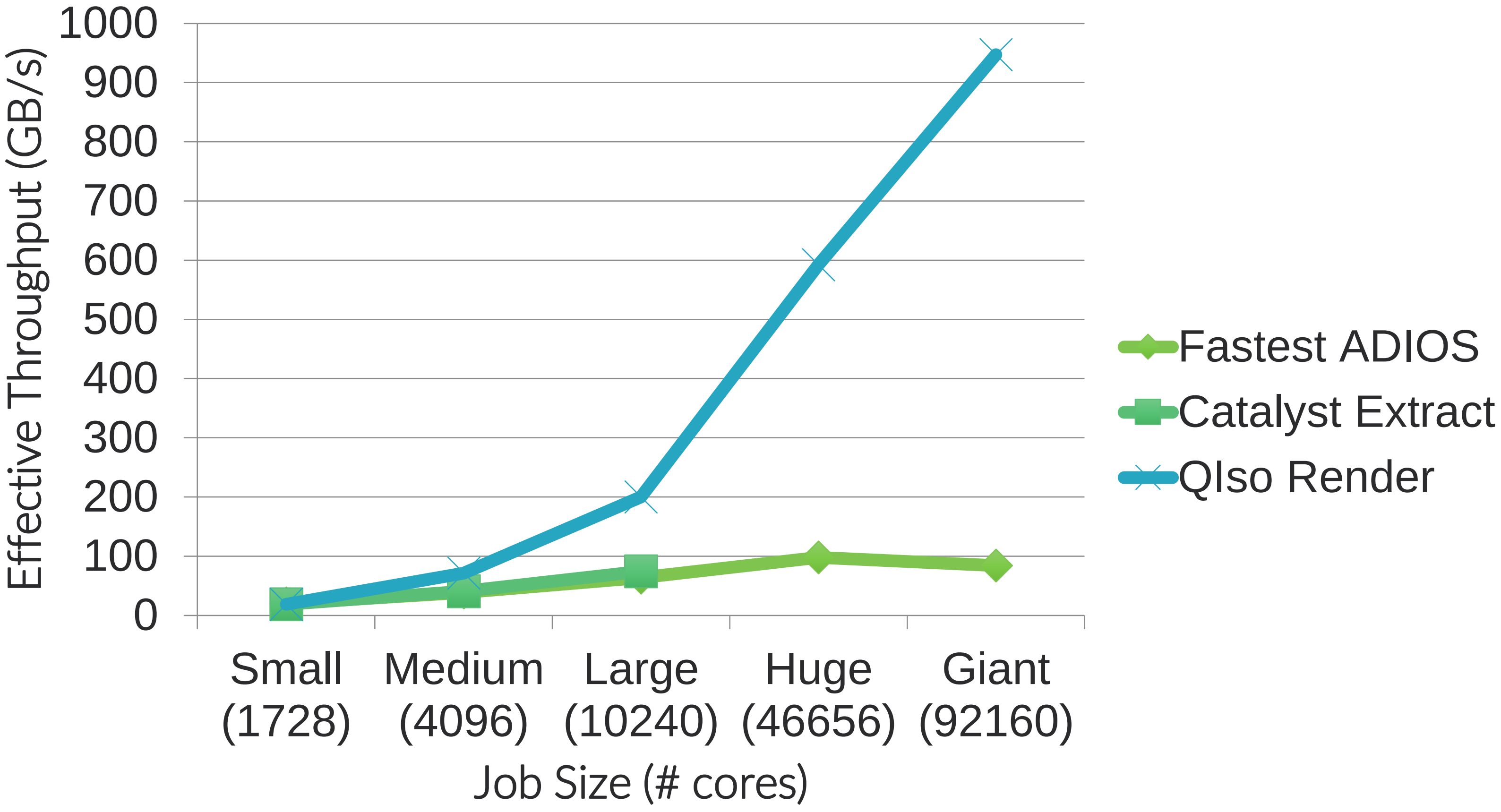
In transit – data is moved: Simulation and *in situ* methods do not share memory

🟥 Simulation Cores

🟦 *In Situ*/In-transit Cores

> BENCHMARK : IN SITU VS. I/O

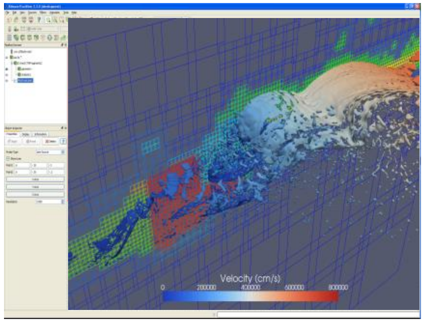
>>100K+ Core challenge



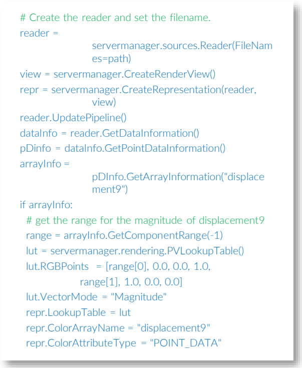
- 10x Improvement over Best I/O methods
- 100x Improvement over poor I/O methods

>CATALYST

>> In-situ visualization

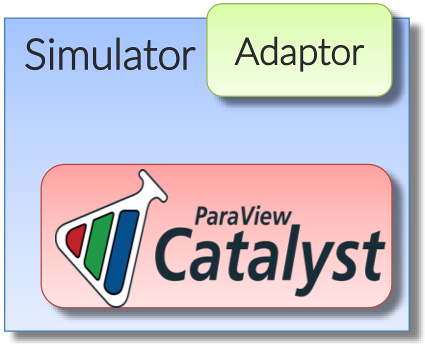


Script
Export

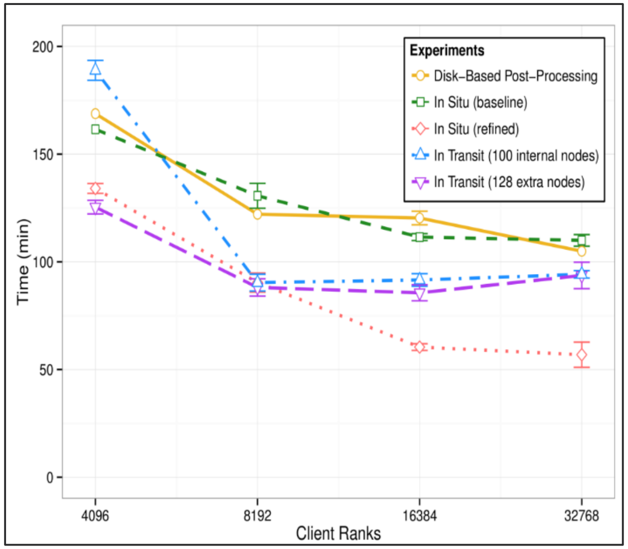


```
# Create the reader and set the filename.
reader =
            servermanager.sources.Reader(FileNam
            es=path)
view = servermanager.CreateRenderView()
repr = servermanager.CreateRepresentation(reader,
            view)
reader.UpdatePipeline()
dataInfo = reader.GetDataInformation()
pDinfo = dataInfo.GetPointDataInformation()
arrayInfo =
            pDInfo.GetArrayInformation("displace
            ment9")
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"
```

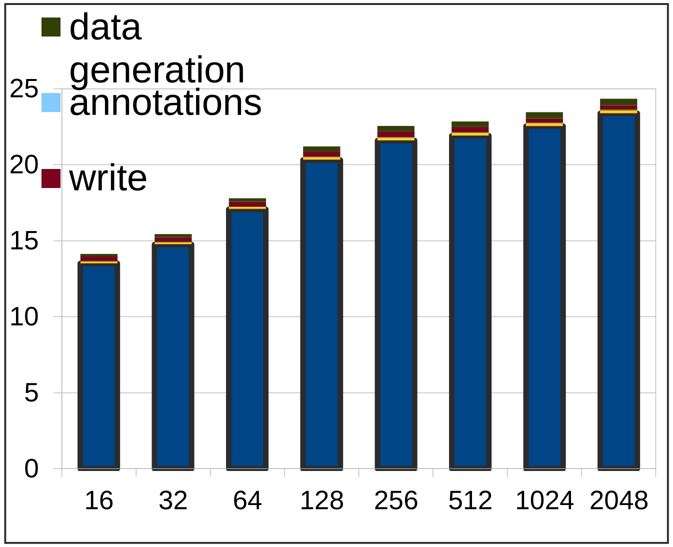
Script
Import



Faster Time to Solution Than Post Hoc



Small Run-Time Overhead

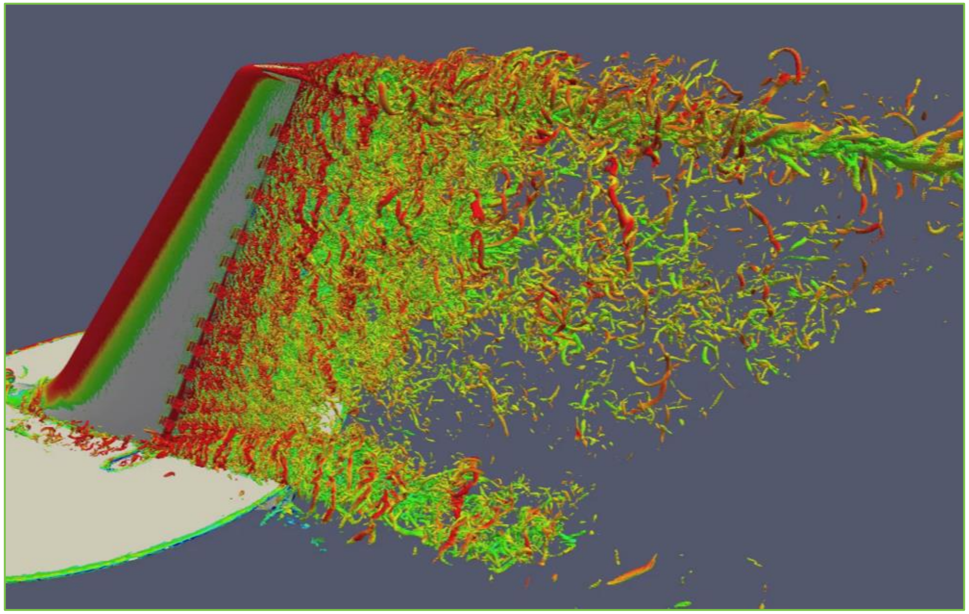


>CATALYST

>> Extreme scale use cases

- Highly scalable algorithms
- Improve Startup
  - Broadcasts scripts to other processes
- Static libraries

1M MPI ranks on Mira@ANL (BG/Q)

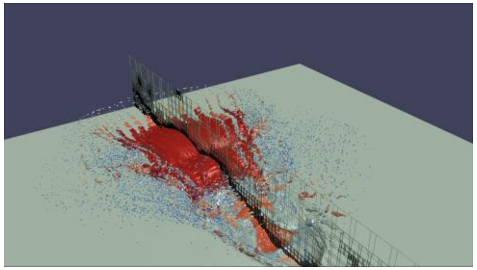


256K MPI ranks on Mira@ANL (IBM BG/Q)

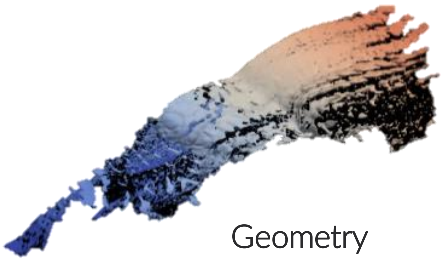
>CATALYST

>> Output Types

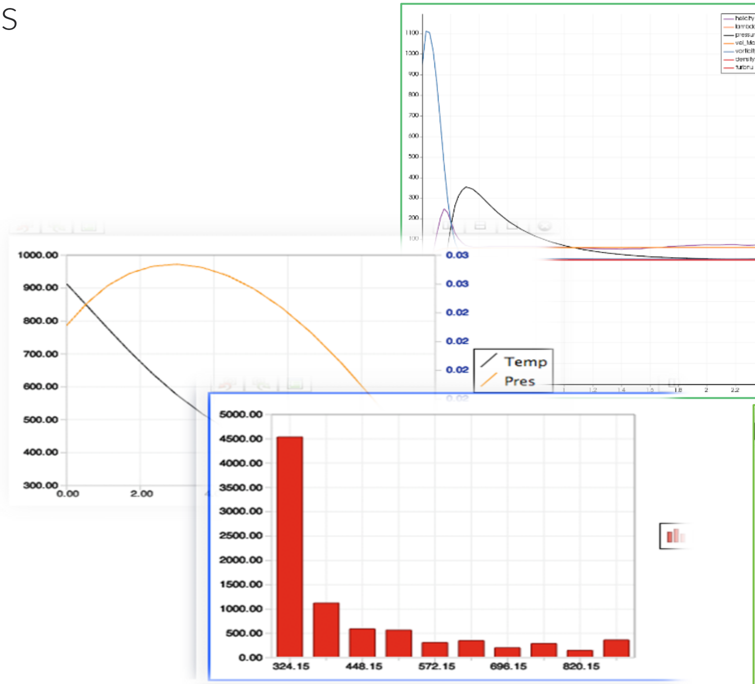
- Images
  - Static Images
  - Explorable Images
  - Charts
- Geometry extracts
  - VTK parallel formats
  - ADIOS format
  - Other formats
- ASCII output
  - Single quantities
  - Statistics



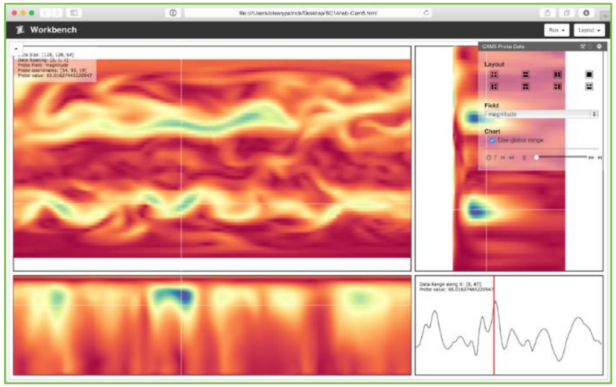
Rendered Images

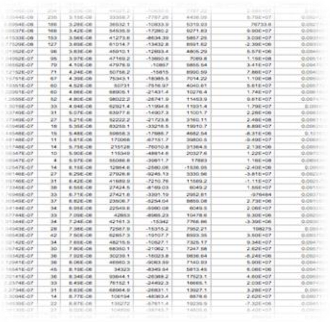


Geometry Extracts



Explorable Images





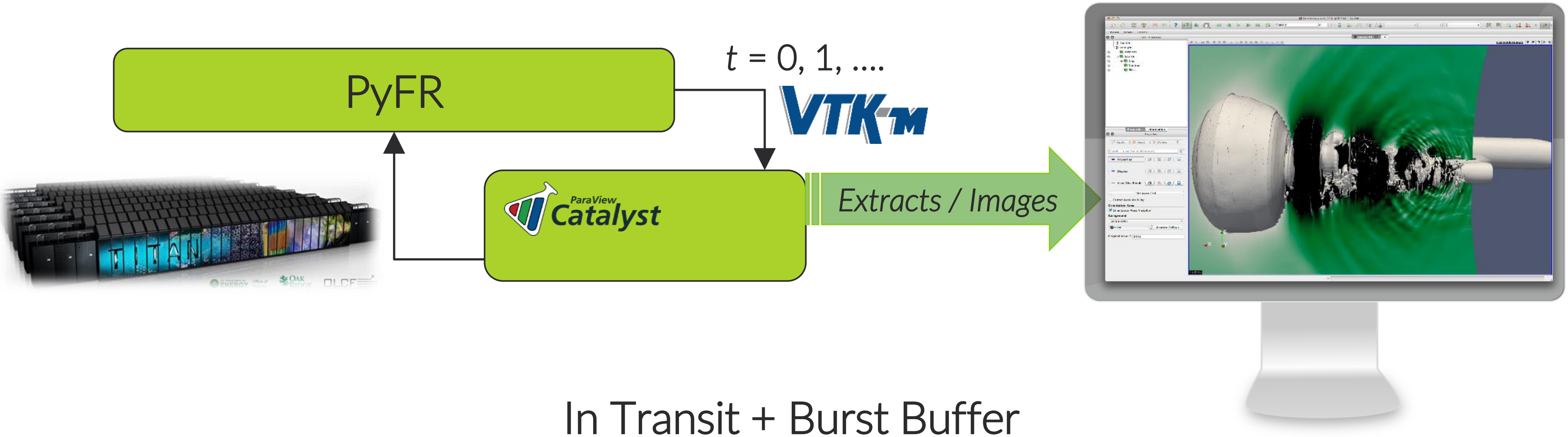
Statistics

Series Data

>CATALYST + VTK-M

>> Large scale in0sity

Live + VTK-m



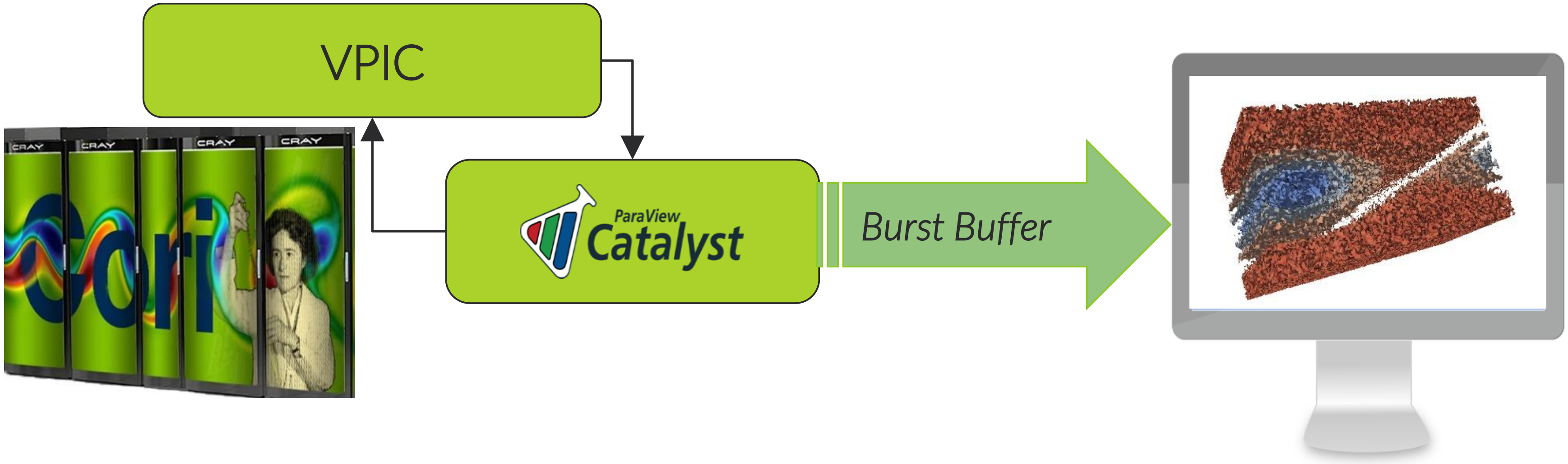
PyFR

t = 0, 1, ...

ParaView Catalyst

Extracts / Images

In Transit + Burst Buffer



VPIC

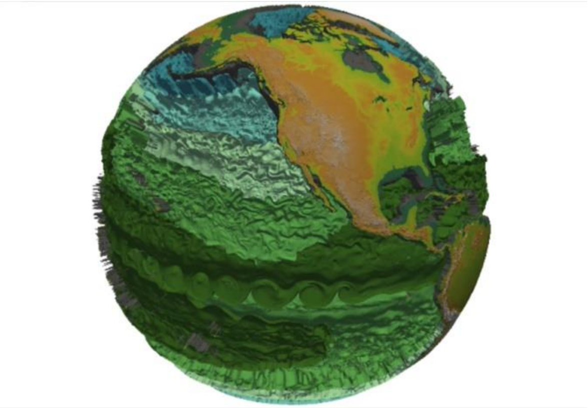
ParaView Catalyst

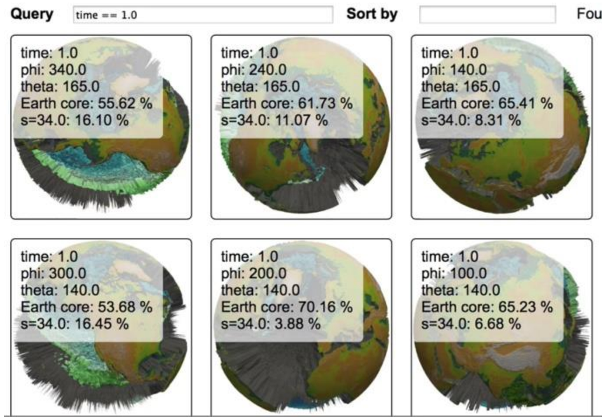
Burst Buffer

>CINEMA

>> Image-based rendering

- Image-based rendering
- Interactive exploration
- Query images based on
  - View
  - Time
  - Objects
- Composite images
- http://cinemascience.org





Query time == 1.0 Sort by Fou

time: 1.0
phi: 340.0
theta: 165.0
Earth core: 55.62 %
s=34.0: 16.10 %

time: 1.0
phi: 240.0
theta: 165.0
Earth core: 61.73 %
s=34.0: 11.07 %

time: 1.0
phi: 140.0
theta: 165.0
Earth core: 65.41 %
s=34.0: 8.31 %

time: 1.0
phi: 300.0
theta: 140.0
Earth core: 53.68 %
s=34.0: 16.45 %

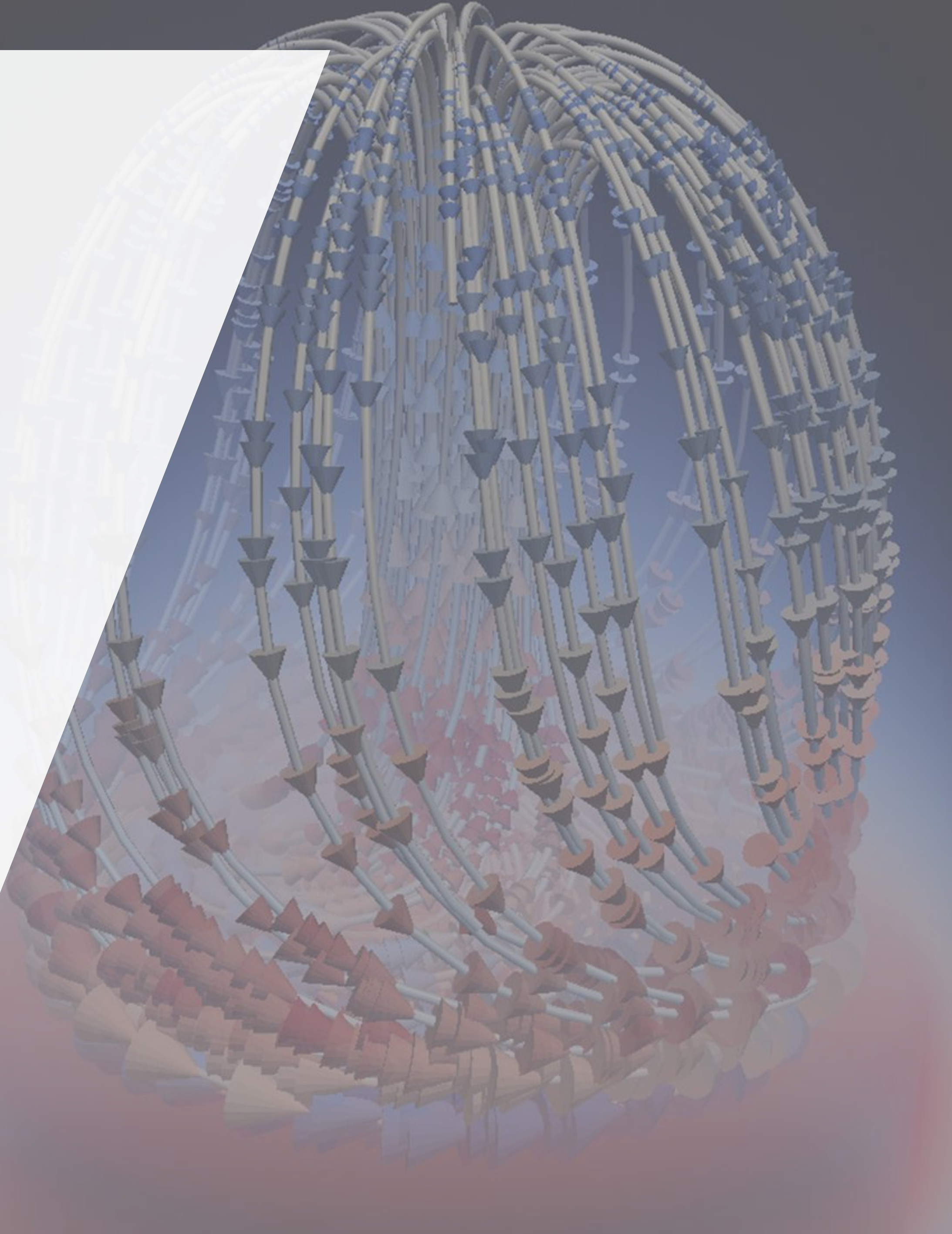
time: 1.0
phi: 200.0
theta: 140.0
Earth core: 70.16 %
s=34.0: 3.88 %

time: 1.0
phi: 100.0
theta: 140.0
Earth core: 65.23 %
s=34.0: 6.68 %

Kitware

Thanks

Questions ?



www.kitware.eu
julien.jomier@kitware.com