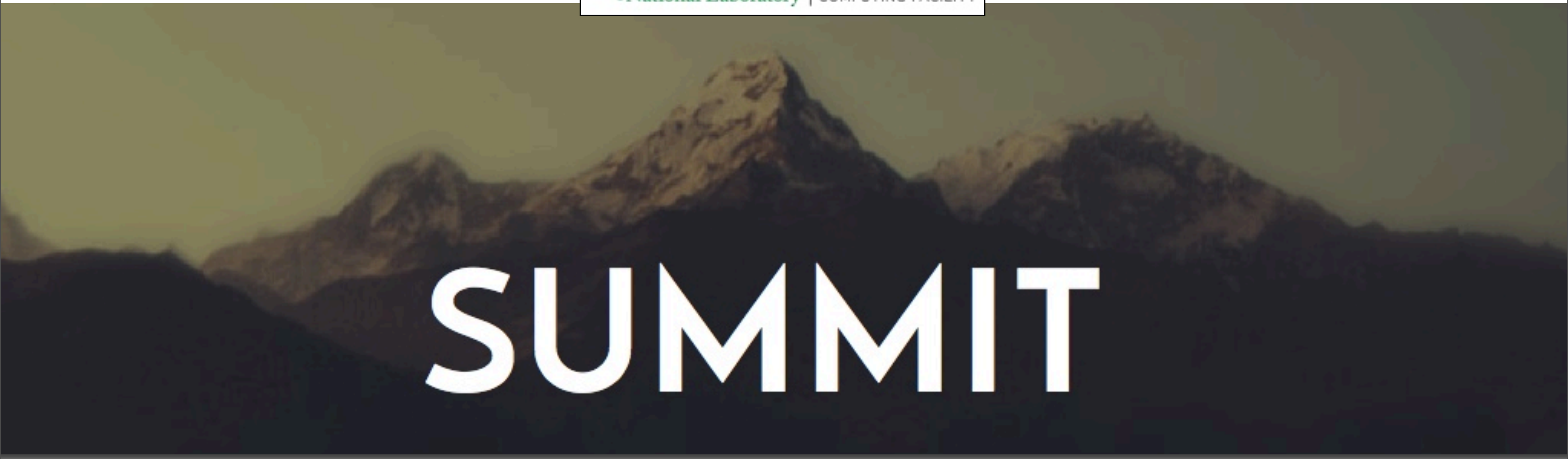
# Visualization Frameworks for Data Staging and In-Situ Environments

**David Pugmire**
Scientific Computing Group,
Oak Ridge National Laboratory

Thanks to: H. Abbasi, S. Ahern, C. Chang, J. Choi, S. Ku, S. Klasky, J. Kress, J. Logan, Q. Liu, J. Meredith, K. Mu, G. Ostrouchov, N. Podhorszki, R. Sisneros, Y. Tian + many more

16 November 2014


SUMMIT


SUMMIT


**or**

# Data Driven Science and Scientific Visualization

| | |
|---|---|
| **Volume** | Increasing mesh resolutions<br>Increasing temporal resolution |
| **Velocity** | Increasing temporal resolution<br>Frequency of data |
| **Variety** | Multi-variate<br>Ensembles<br>Increasing complexity |
| **Veracity** | Uncertainty<br>Errors<br>Approximations |
| **Value** | Visualization and Analysis<br>Feature detection<br>Scientific insight |

# Today's Tools in Data Driven Science World

| | |
|---|---|
| **Volume** | Increasing mesh resolutions<br>Increasing temporal resolution |
| **Velocity** | Increasing temporal resolution<br>Frequency of data |
| **Variety** | Multi-variate<br>Ensembles<br>Increasing complexity |
| **Veracity** | Uncertainty<br>Errors<br>Approximations |
| **Value** | Visualization and Analysis<br>Feature detection<br>Scientific insight |

# Today's Tools in Data Driven Science World

| | |
|---|---|
| **Volume** | Increasing mesh resolutions<br>Increasing temporal resolution |
| **Velocity** | Increasing temporal resolution<br>Frequency of data |
| **Variety** | Multi-variate<br>Ensembles<br>Increasing complexity |
| **Veracity** | Uncertainty<br>Errors<br>Approximations |
| **Value** | Visualization and Analysis<br>Feature detection<br>Scientific insight |

Focused on **Volume**

Others V's are harder, and often a function of Volume

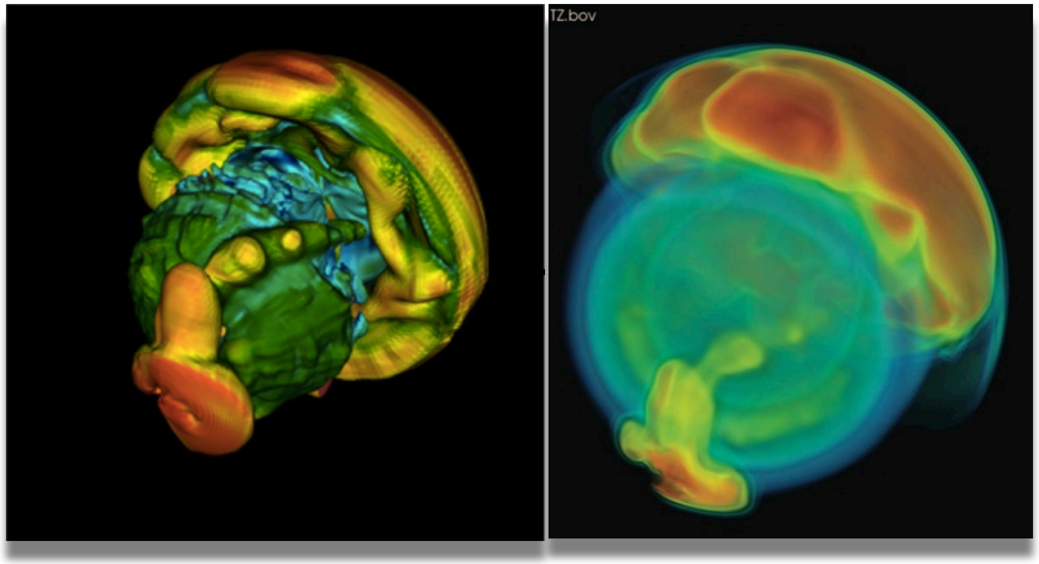
# Scalability of Visualization Tools

## Research Questions:

- Can current visualization tools survive at the exascale?
- What are the bottlenecks at the largest scales?
- What differences do architectures make?

## Methodology:

- "Create" exascale data (trillions of zones)
- Execute a simple workflow :
  - Read data
  - Volume render / contour data
  - Render and composite




Core-collapse supernova simulation. Data courtesy of T. Mezzacappa (GenASiS)

see: **Extreme Scaling of Production Visualization Software on Diverse Architectures**, IEEE CG&A, 2010

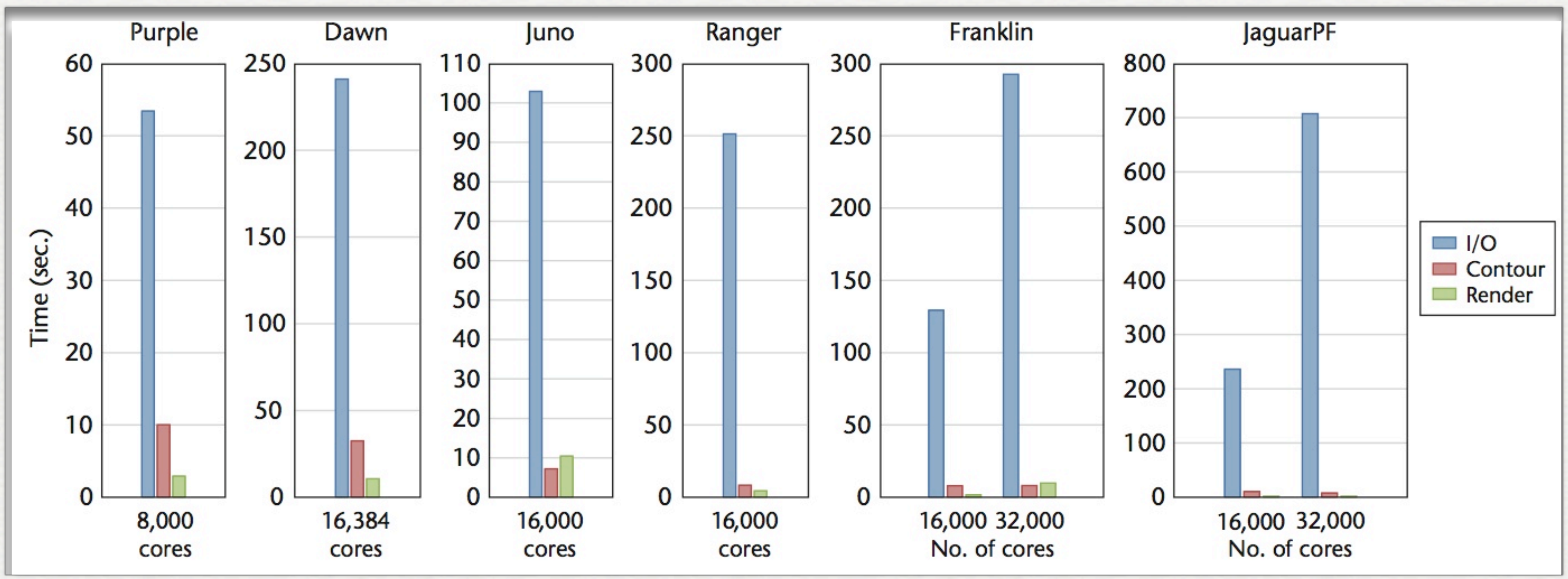
OAK RIDGE National Laboratory XVis SDG SDAV OLCF OAK RIDGE LEADERSHIP COMPUTING FACILITY

# Scalability of Visualization Tools

| Machine name | Machine type or OS | Total no. of cores | Memory per core (Gbytes) | System type | Clock speed | Peak flops | Top 500 rank (as of Nov. 2009) |
|---|---|---|---|---|---|---|---|
| JaguarPF | Cray | 224,162 | 2.0 | XT5 | 2.6 GHz | 2.33 Pflops | 1 |
| Ranger | Sun Linux | 62,976 | 2.0 | Opteron Quad | 2.0 GHz | 503.8 Tflops | 9 |
| Dawn | Blue Gene/P | 147,456 | 1.0 | PowerPC | 850.0 MHz | 415.7 Tflops | 11 |
| Franklin | Cray | 38,128 | 1.0 | XT4 | 2.6 GHz | 352 Tflops | 15 |
| Juno | Commodity (Linux) | 18,402 | 2.0 | Opteron Quad | 2.2 GHz | 131.6 Tflops | 27 |
| Purple | AIX (*Advanced Interactive Executive*) | 12,208 | 3.5 | Power5 | 1.9 GHz | 92.8 Tflops | 66 |

# Scalability of Visualization Tools

| Machine name | Machine type or OS | Total no. of cores | Memory per core (Gbytes) | System type | Clock speed | Peak flops | Top 500 rank (as of Nov. 2009) |
|---|---|---|---|---|---|---|---|
| JaguarPF | Cray | 224,162 | 2.0 | XT5 | 2.6 GHz | 2.33 Pflops | 1 |
| Ranger | Sun Linux | 62,976 | 2.0 | Opteron Quad | 2.0 GHz | 503.8 Tflops | 9 |
| Dawn | Blue Gene/P | 147,456 | 1.0 | PowerPC | 850.0 MHz | 415.7 Tflops | 11 |
| Franklin | Cray | 38,128 | 1.0 | XT4 | 2.6 GHz | 352 Tflops | 15 |
| Juno | Commodity (Linux) | 18,402 | 2.0 | Opteron Quad | 2.2 GHz | 131.6 Tflops | 27 |
| Purple | AIX (Advanced Interactive Executive) | 12,208 | 3.5 | Power5 | 1.9 GHz | 92.8 Tflops | 66 |




OAK RIDGE National Laboratory XVis SDG SDAV OLCF Oak Ridge Leadership Computing Facility

# Challenges at Exascale:

| System Parameter | 2011 | "2018" Swim Lane 1 | "2018" Swim Lane 2 | Factor Change |
|---|---|---|---|---|
| System Peak | 2 Pf/s | 1 Ef/s | | 500 |
| Power | 6 MW | $\leq$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 GB/s | 100 GB/s | 1000 GB/s | 66–660 |
| System Size (nodes) | 18700 | 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 | | 100-200 |

From: Exascale Workshop on Data Analysis, Management and Visualization. DOE ASCR 2011

# Challenges at Exascale:

| System Parameter | 2011 | "2018" Swim Lane 1 | "2018" Swim Lane 2 | Factor Change |
|---|---|---|---|---|
| System Peak | 2 Pf/s | 1 Ef/s | | 500 |
| Power | 6 MW | $\leq$20 MW | | 3 |
| System Memory | 0.3 PB | 32–64 PB | | 100–200 |
| Total Concurrency | 225K | 1B$\times$10 | 1B$\times$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 GB/s | 100 GB/s | 1000 GB/s | 66–660 |
| System Size (nodes) | 18700 | 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 | | 100-200 |

From: Exascale Workshop on Data Analysis, Management and Visualization. DOE ASCR 2011

# Challenges at Exascale:

| System Parameter | 2011 | "2018" Swim Lane 1 | "2018" Swim Lane 2 | Factor Change |
|---|---|---|---|---|
| System Peak | 2 Pf/s | 1 Ef/s | | 500 |
| Power | 6 MW | $\leq$20 MW | | 3 |
| System Memory | 0.3 PB | 32–64 PB | | $\boxed{100\text{–}200}$ |
| Total Concurrency | 225K | 1B×10 | 1B×100 | $\boxed{40{,}000\text{–}400{,}000}$ |
| Node Performance | 125 GF | 1 TF | 10 TF | 8–80 |
| Node Concurrency | 12 | 1,000 | 10,000 | 83–830 |
| Network BW | 1.5 GB/s | 100 GB/s | 1000 GB/s | 66–660 |
| System Size (nodes) | 18700 | 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 | | 100-200 |

From: Exascale Workshop on Data Analysis, Management and Visualization. DOE ASCR 2011

# Challenges at Exascale:

| System Parameter | 2011 | "2018" Swim Lane 1 | "2018" Swim Lane 2 | Factor Change |
|---|---|---|---|---|
| System Peak | 2 Pf/s | 1 Ef/s | | 500 |
| Power | 6 MW | $\leq$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 GB/s | 100 GB/s | 1000 GB/s | 66–660 |
| System Size (nodes) | 18700 | 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 | | 100-200 |

From: Exascale Workshop on Data Analysis, Management and Visualization. DOE ASCR 2011

# Challenges at Exascale:

| System Parameter | 2011 | "2018" Swim Lane 1 | "2018" Swim Lane 2 | Factor Change |
|---|---|---|---|---|
| System Peak | 2 Pf/s | 1 Ef/s | | 500 |
| Power | 6 MW | $\leq$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 GB/s | 100 GB/s | 1000 GB/s | 66–660 |
| System Size (nodes) | 18700 | 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 | | 100-200 |

From: Exascale Workshop on Data Analysis, Management and Visualization. DOE ASCR 2011

I/O Caveats:

| System | System Peak | I/O Peak | I/O Reality | I/O Hero |
|---|---|---|---|---|
| JaguarPF | 2PF | 200 GB/s | 1 GB/s | 60 GB/s |
| Titan | 20PF | 1.2 TB/s | 1 GB/s | 120 GB/s |
| Future | 1000PF | 10 TB/s (?) | ?? | ?? |

# Challenges at Exascale:

| System Parameter | 2011 | "2018" Swim Lane 1 | "2018" Swim Lane 2 | Factor Change |
|---|---|---|---|---|
| System Peak | 2 Pf/s | 1 Ef/s | | 500 |
| Power | 6 MW | $\leq$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 GB/s | 100 GB/s | 1000 GB/s | 66–660 |
| System Size (nodes) | 18700 | 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 | | 100-200 |

From: Exascale Workshop on Data Analysis, Management and Visualization. DOE ASCR 2011

I/O Caveats:

| System | System Peak | I/O Peak | I/O Reality | I/O Hero |
|---|---|---|---|---|
| JaguarPF | 2PF | 200 GB/s | 1 GB/s | 60 GB/s |
| Titan | 20PF | 1.2 TB/s | 1 GB/s | 120 GB/s |
| Future | 1000PF | 10 TB/s (?) | ?? | ?? |

We will get **less** of what we **want**

We will get **more** of what we **don't know how to use**

OAK RIDGE National Laboratory  XVis  SDG  SDAV  OLCF Oak Ridge Leadership Computing Facility

# Impacts on Visualization

## Massive Concurrency

- Production tools of today cannot fully utilize
- Challenges of new programming models

## Complex Memory

- Visualization APIs not available
- New algorithms and programming models

## Decreased I/O Performance

- Cannot rely on storage system in workflows
- In situ methods become critical

## Memory Constraints

- Expressive and flexible data models
- Efficient data models become imperative, especially for zero-copy in situ applications

# Path Forward

### Massive Concurrency

- Library that supports/abstracts:
  - Heterogeneous computing
  - Fine grained parallelism

### Complex Memory

- Advanced data model
- API that manages/abstracts the complexities

### Decreased I/O Performance

- Data management and movement library
- Flexible in situ interface

### Memory Constraints

- Advanced, expressive data model
- Efficient data model
  - Representation and execution

# Path Forward

## Extreme-Scale Analysis and Visualization Library (EAVL)

- Advanced visualization and analysis for next generation computer architectures
- Part of DOE funded VTK-m efforts

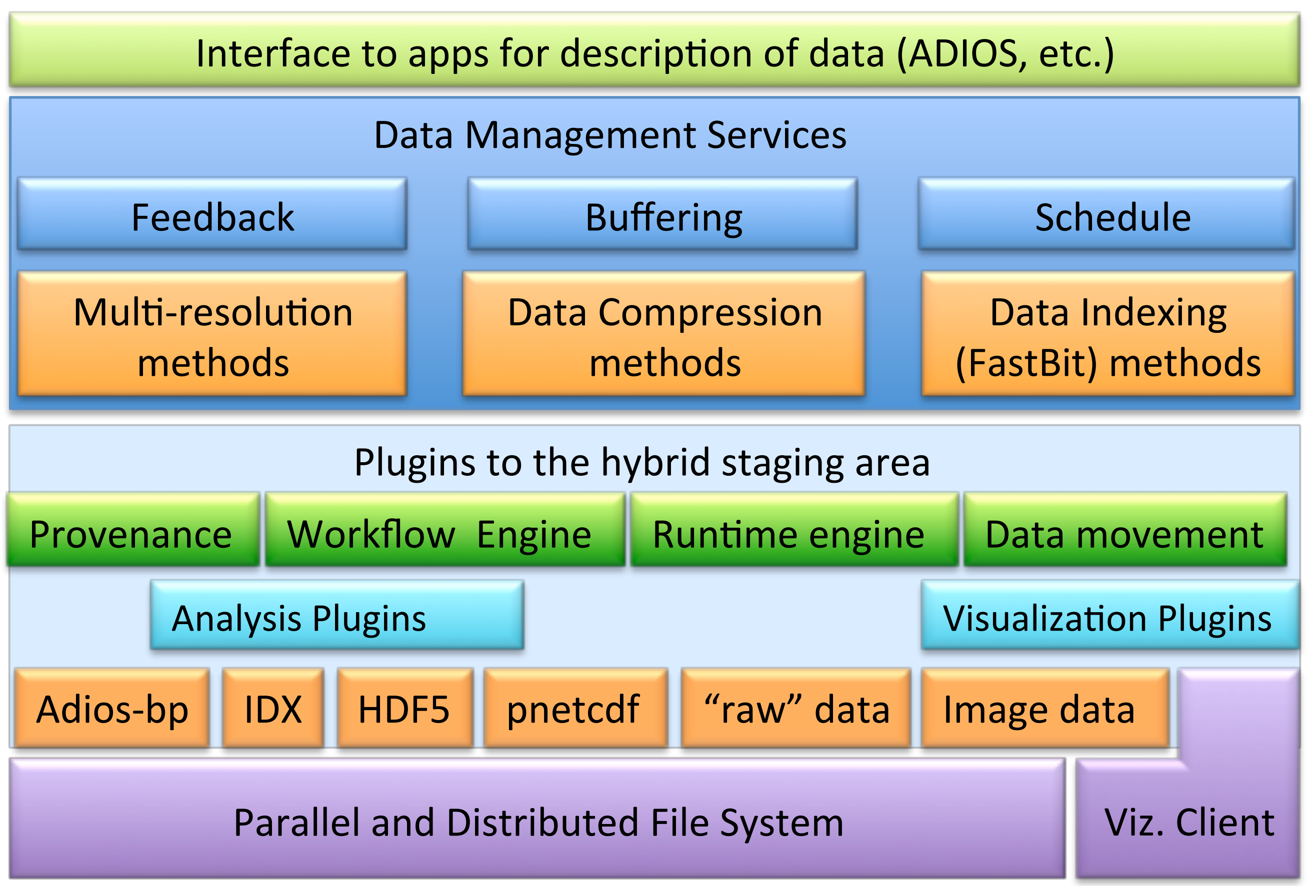
## Adaptable I/O System (ADIOS)

- Middleware abstraction of I/O for HPC systems
- Provides increased performance for disk based I/O, and in situ processing

# Data Management Framework: ADIOS

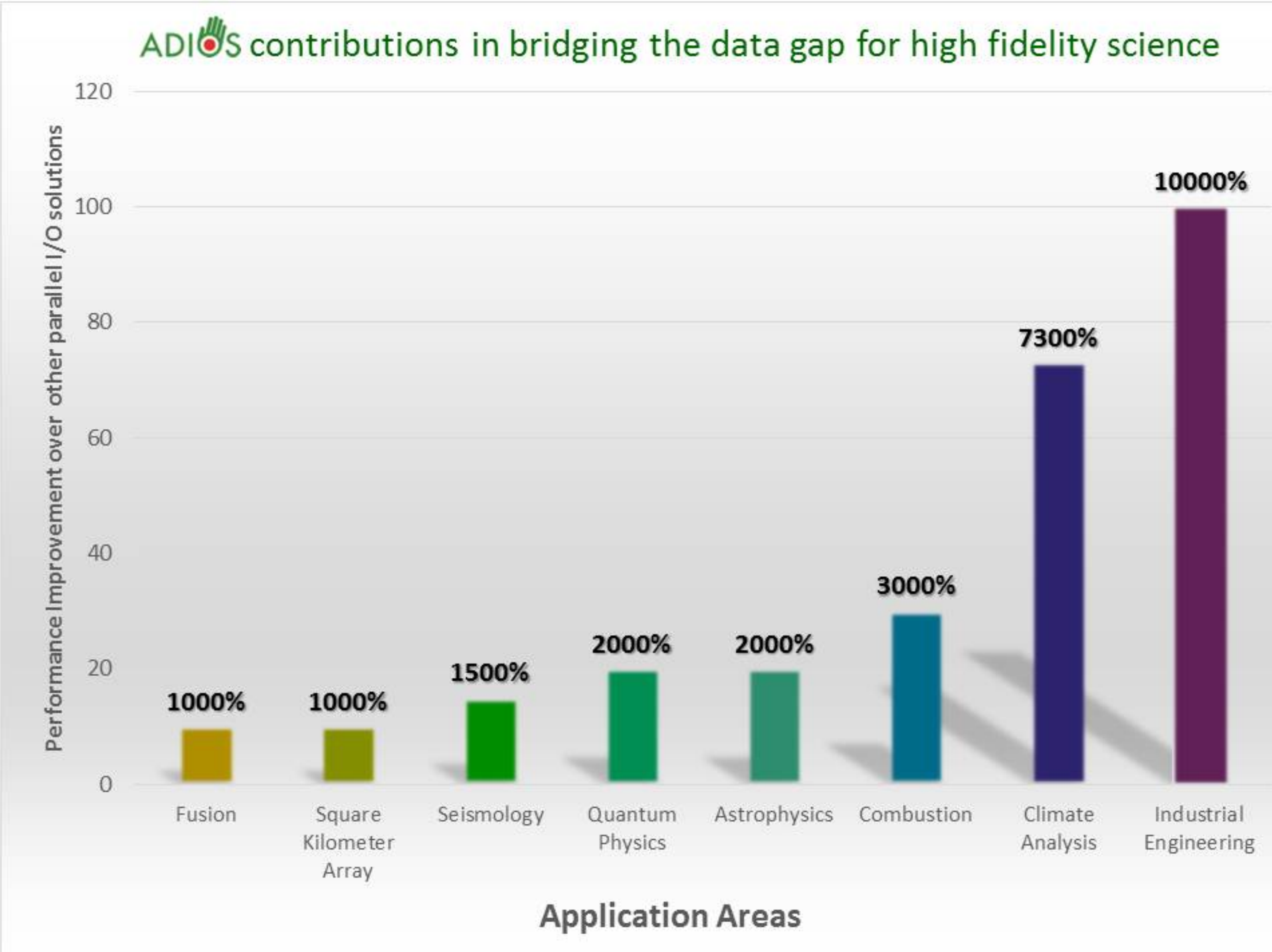
- An I/O abstraction framework
- Provides portable, fast, easy-to-use metadata rich output
- Change I/O method on-the-fly
- Abstract the API from the method
- Looks to provide support for "90% of applications"

http://www.nccs.gov/user-support/center-projects/adios/



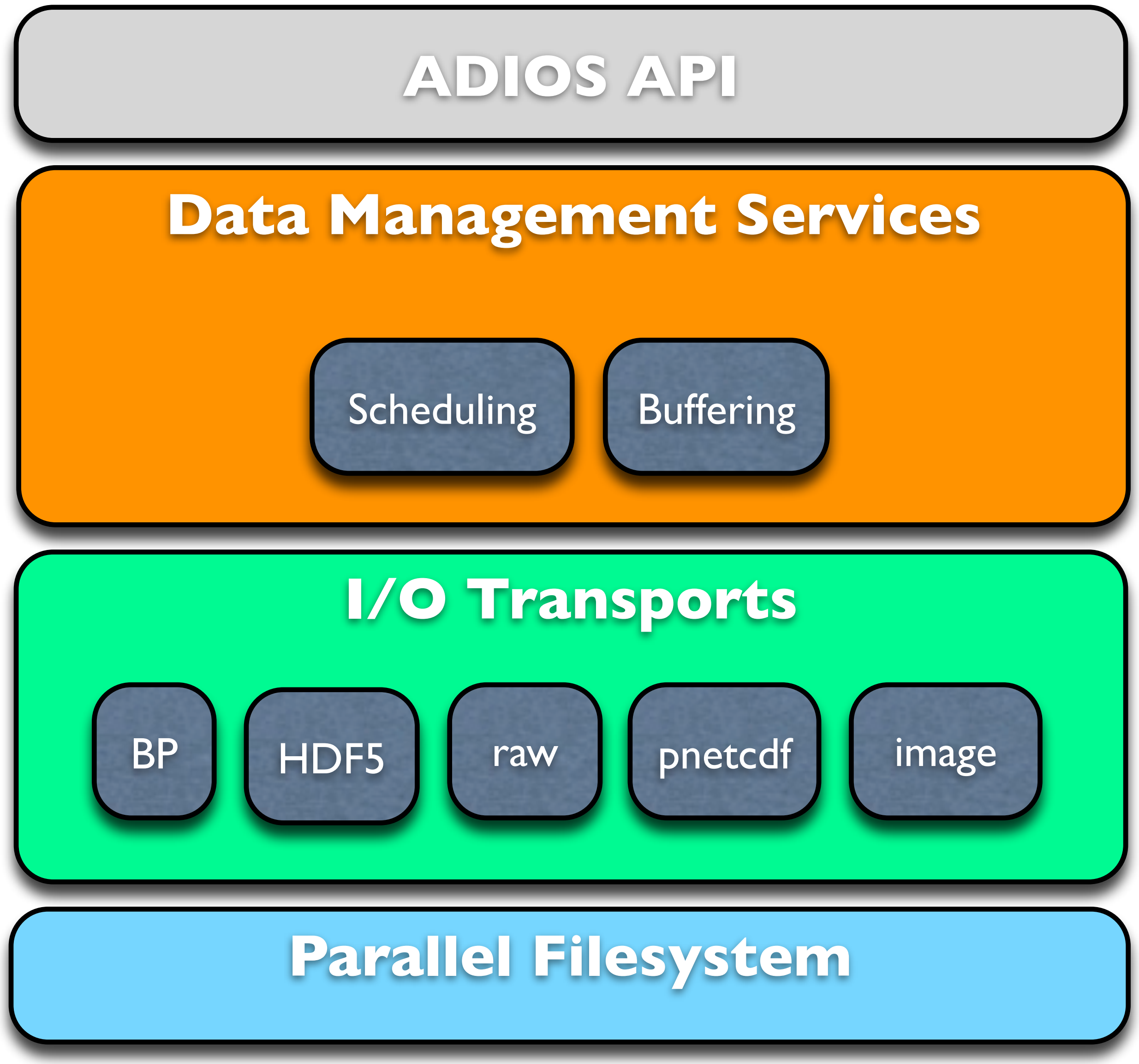

- Astrophysics
- Climate
- Combustion
- CFD
- Environmental Science
- Fusion
- Earthquake
- Material Science
- Medical: Pathology
- Neutron Science
- Nuclear Science
- Quantum Turbulence
- Relativity
- Seismology
- Sub-surface Modeling
- Weather
- Satellite Processing

Oak Ridge National Laboratory · XVis · SDG · SDAV · OLCF Oak Ridge Leadership Computing Facility

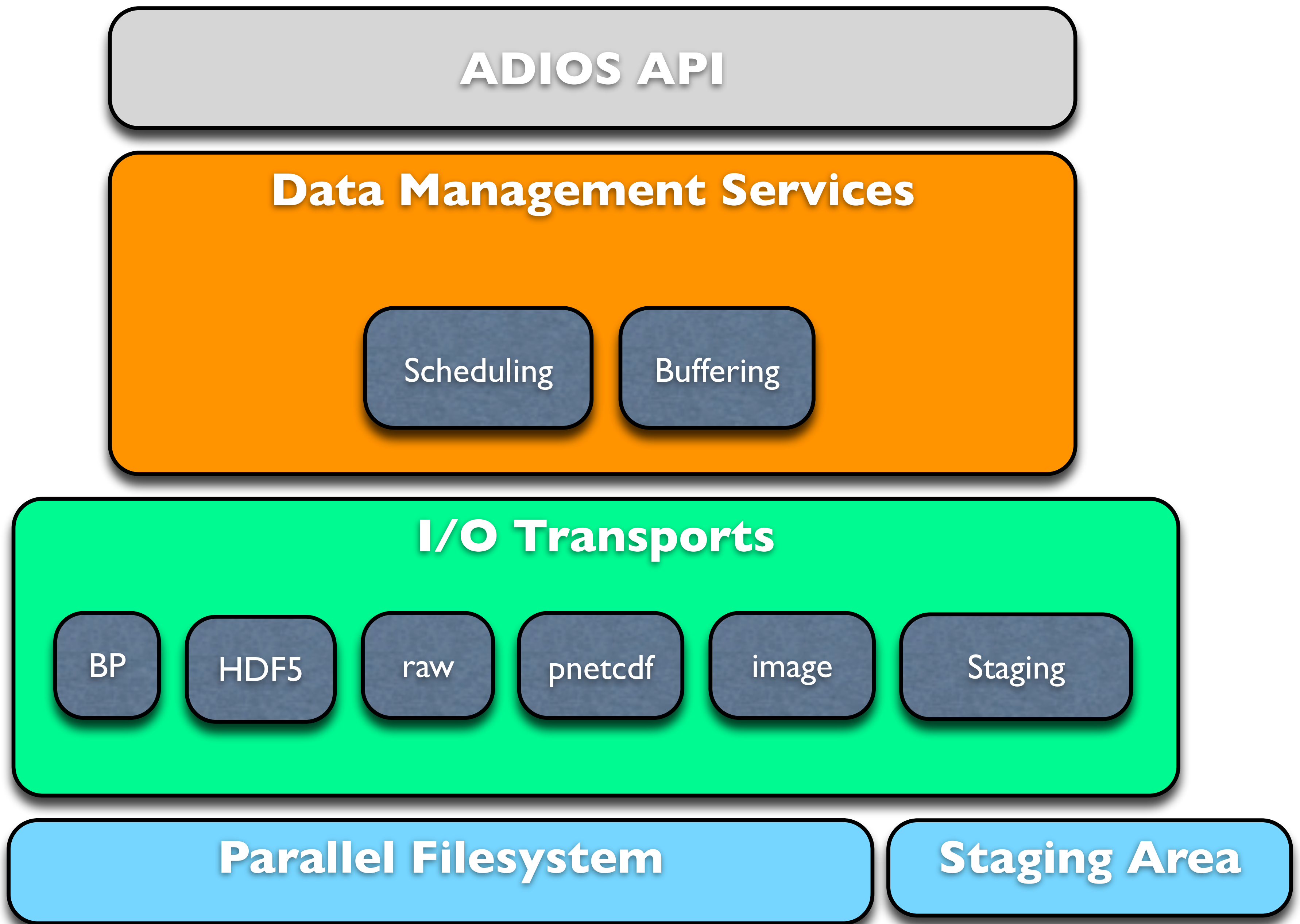

ADIOS contributions in bridging the data gap for high fidelity science
Performance Improvement over other parallel I/O solutions
120
100
80
60
40
20
0
1000%
1000%
1500%
2000%
2000%
3000%
7300%
10000%
Fusion
Square Kilometer Array
Seismology
Quantum Physics
Astrophysics
Combustion
Climate Analysis
Industrial Engineering
Application Areas

# I/O in ADIOS

- Carefully manage movement of data in network and I/O system
- Data format agnostic
- Allows simulations to spend more time in compute, or allows more frequent output of data
- Visualization is especially sensitive to I/O performance

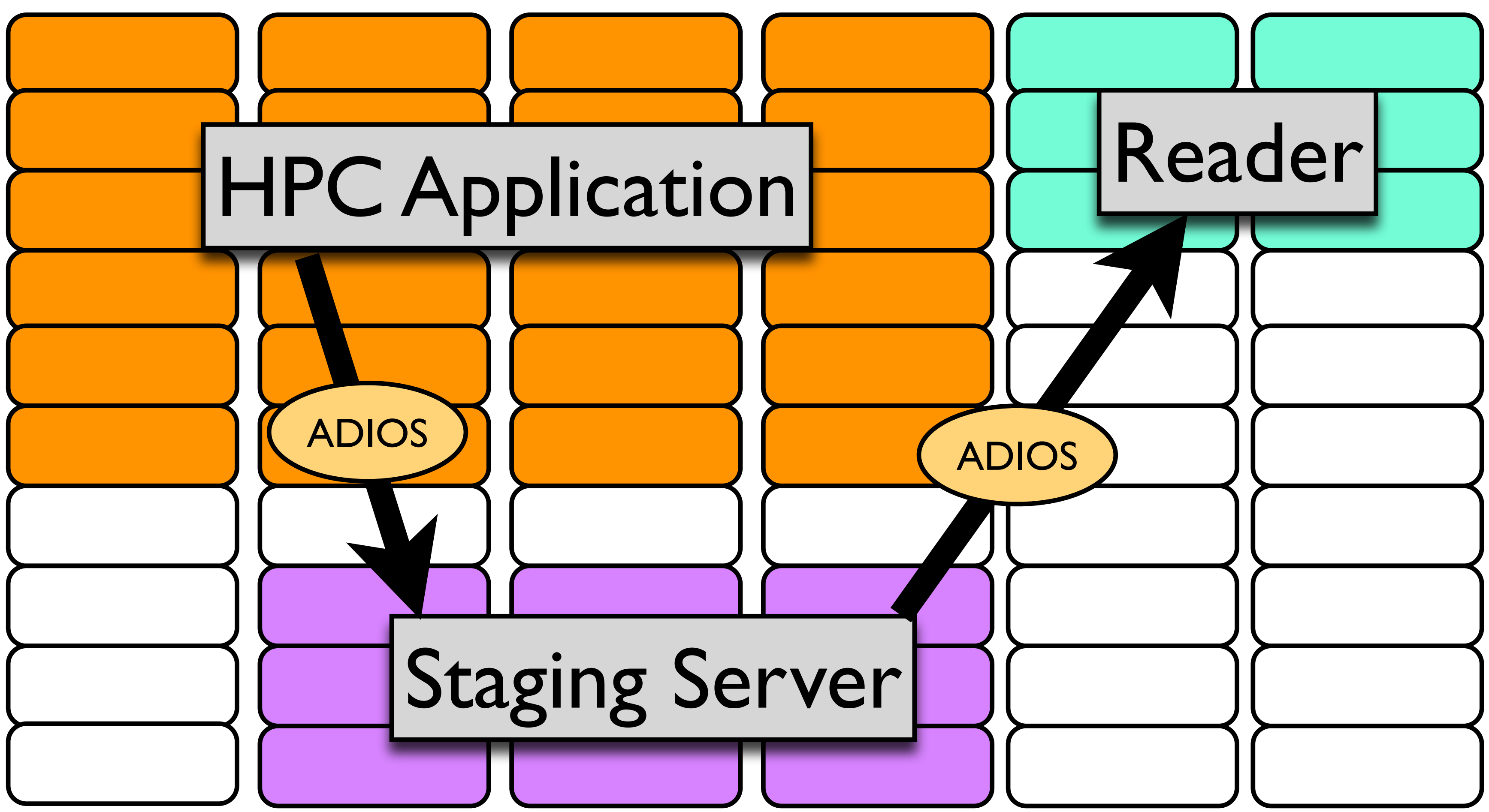


# Data Staging in ADIOS



# Data Staging in ADIOS

- Same application API can be used to do more advanced data movement
- Plugins will operate on data streams in user-defined ways



# Extreme-scale Analysis and Visualization Library **(EAVL)**

EAVL enables advanced visualization and analysis for the next generation scientific simulations, supercomputing systems, and end-user analysis tools.

**New Mesh Layouts**

- More accurately represent simulation data in analysis results
- Support novel simulation applications

**Greater Memory Efficiency**

- Support future low-memory systems
- Minimize data movement and transformation costs

**Parallel Algorithm Framework**

- Accelerator-based system support
- Pervasive parallelism for multi-core and many-core processors

**In Situ Support**

- Direct zero-copy mapping of data from simulation to analysis codes
- Heterogeneous processing models allow broad platform support

J.S. Meredith, S. Ahern, D. Pugmire, R. Sisneros, "EAVL: The Extreme-scale Analysis and Visualization Library", Eurographics Symposium on Parallel Graphics and Visualization (EGPGV), 2012.

**http://ft.ornl.gov/eavl**

Oak Ridge National Laboratory XVis SDG SDAV OLCF Oak Ridge Leadership Computing Facility

# Gaps in Current Data Models

- Traditional data set models target only common combinations of cell and point arrangements
- This limits their expressiveness and flexibility

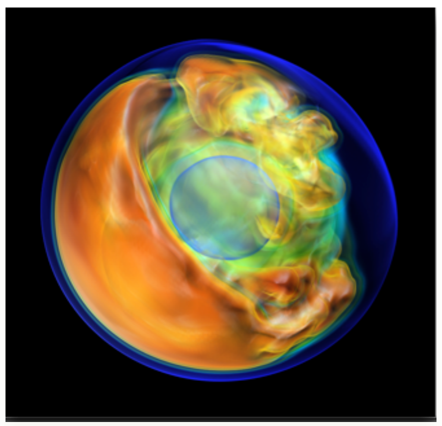
| | | **Point Arrangement** | | |
|---|---|---|---|---|
| **Cells** | **Coordinates** | **Explicit** | **Logical** | **Implicit** |
| Structured | Strided | Structured Grid | | |
| | Separated | | Rectilinear Grid | Image Data |
| Unstructured | Strided | Unstructured Grid | | |
| | Separated | | | |

# Arbitrary Compositions for Flexibility

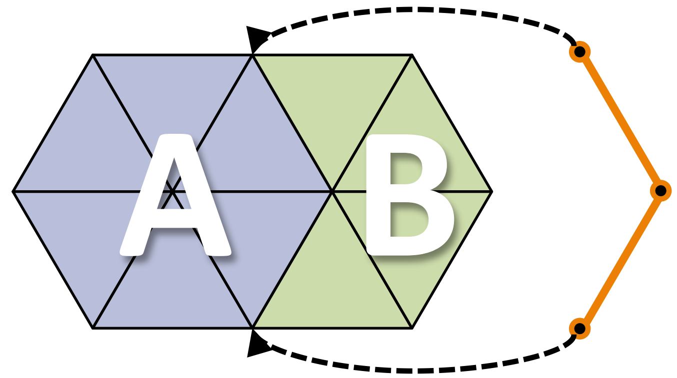
- EAVL allows clients to construct data sets from cell and point arrangements that exactly match their original data
  - In effect, this allows for hybrid and novel mesh types
- Native data results in great accuracy and efficiency

| | | Point Arrangement | | |
|---|---|---|---|---|
| **Cells** | **Coordinates** | **Explicit** | **Logical** | **Implicit** |
| Structured | Strided | ✓ | ✓ | ✓ |
| | Separated | ✓ | ✓ | ✓ |
| Unstructured | Strided | ✓ | ✓ | ✓ |
| | Separated | ✓ | ✓ | ✓ |

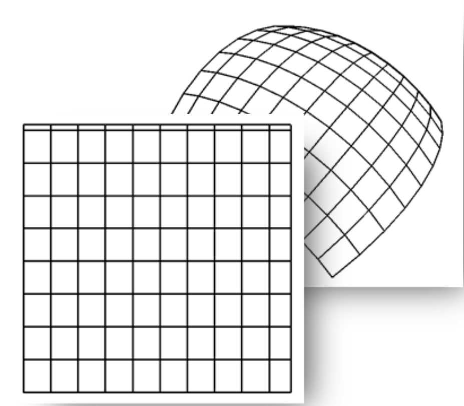
# Other Data Model Gaps Addressed in EAVL



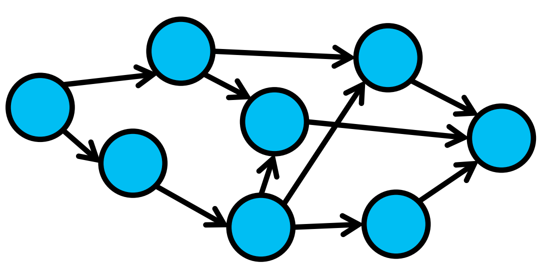
Low/high dimensional data (7D GenASiS)



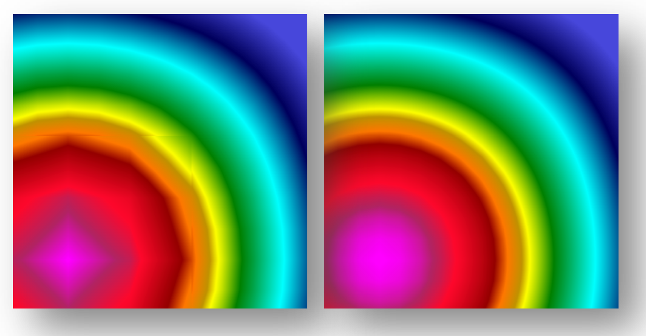

Multiple cell groups in one mesh



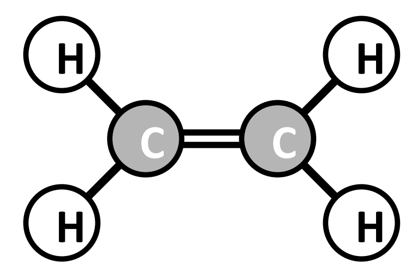
Multiple coordinate systems (lat/lon + XY)



Non-physical data (graphs, sensor, etc)



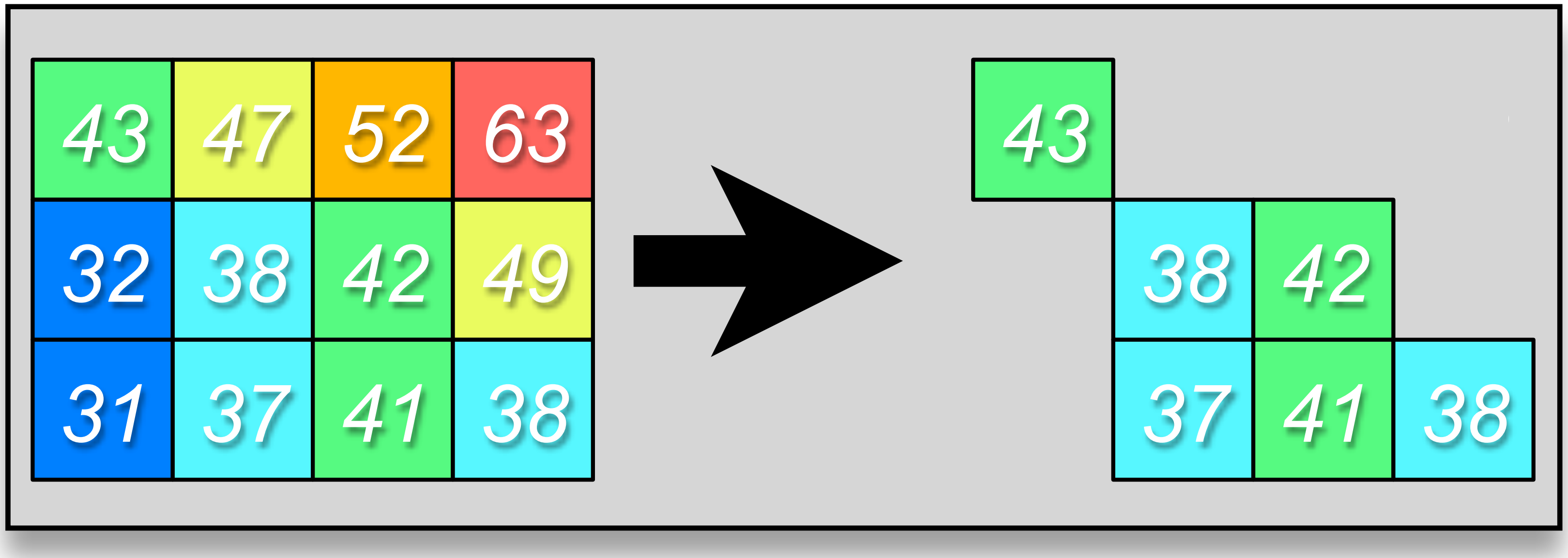
Novel and hybrid mesh types (quadtree grid from MADNESS)




Mixed topology (atoms+bonds)

# Example: Memory and Algorithmic Efficiency

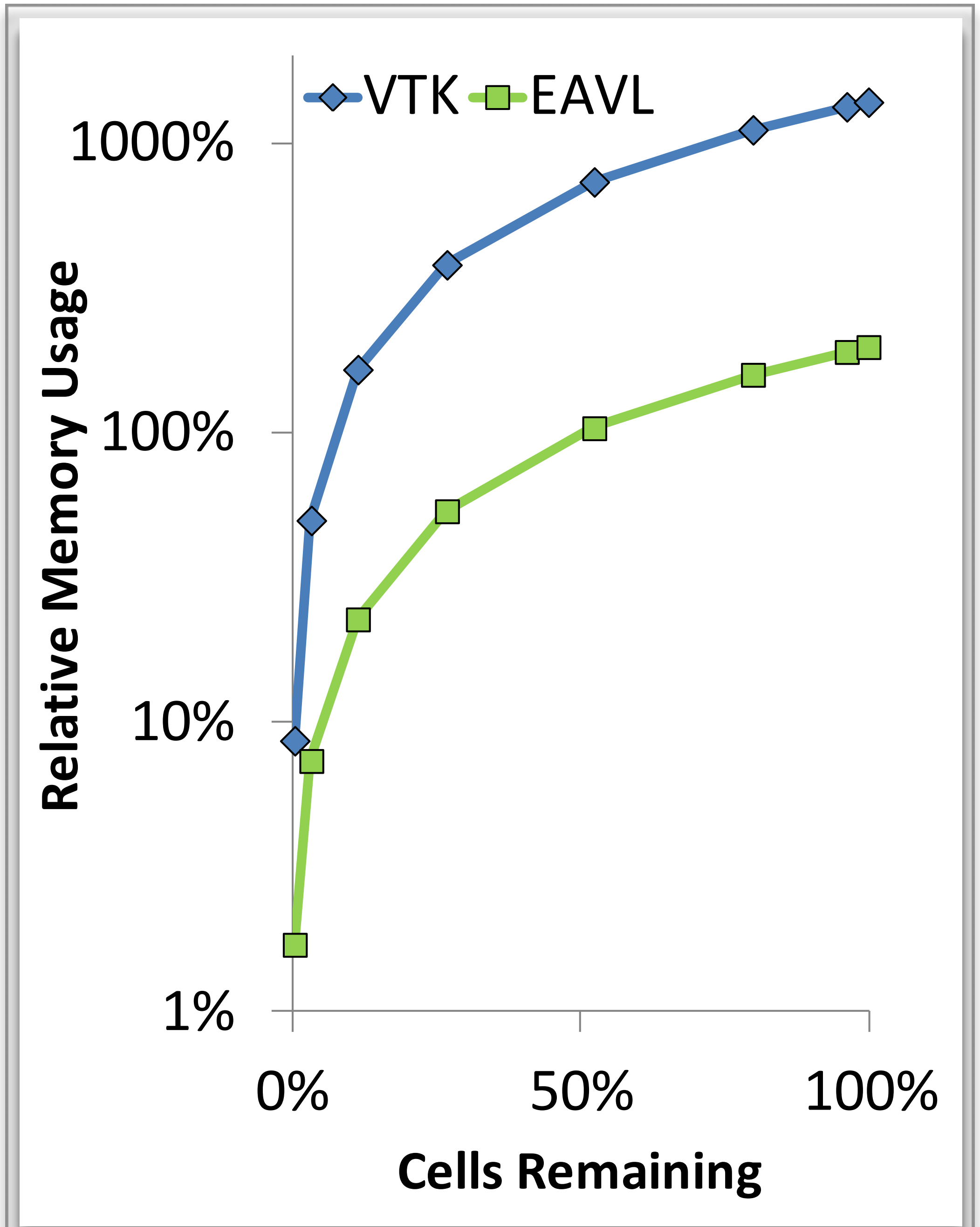
Threshold regular grid: 35 < pressure < 45



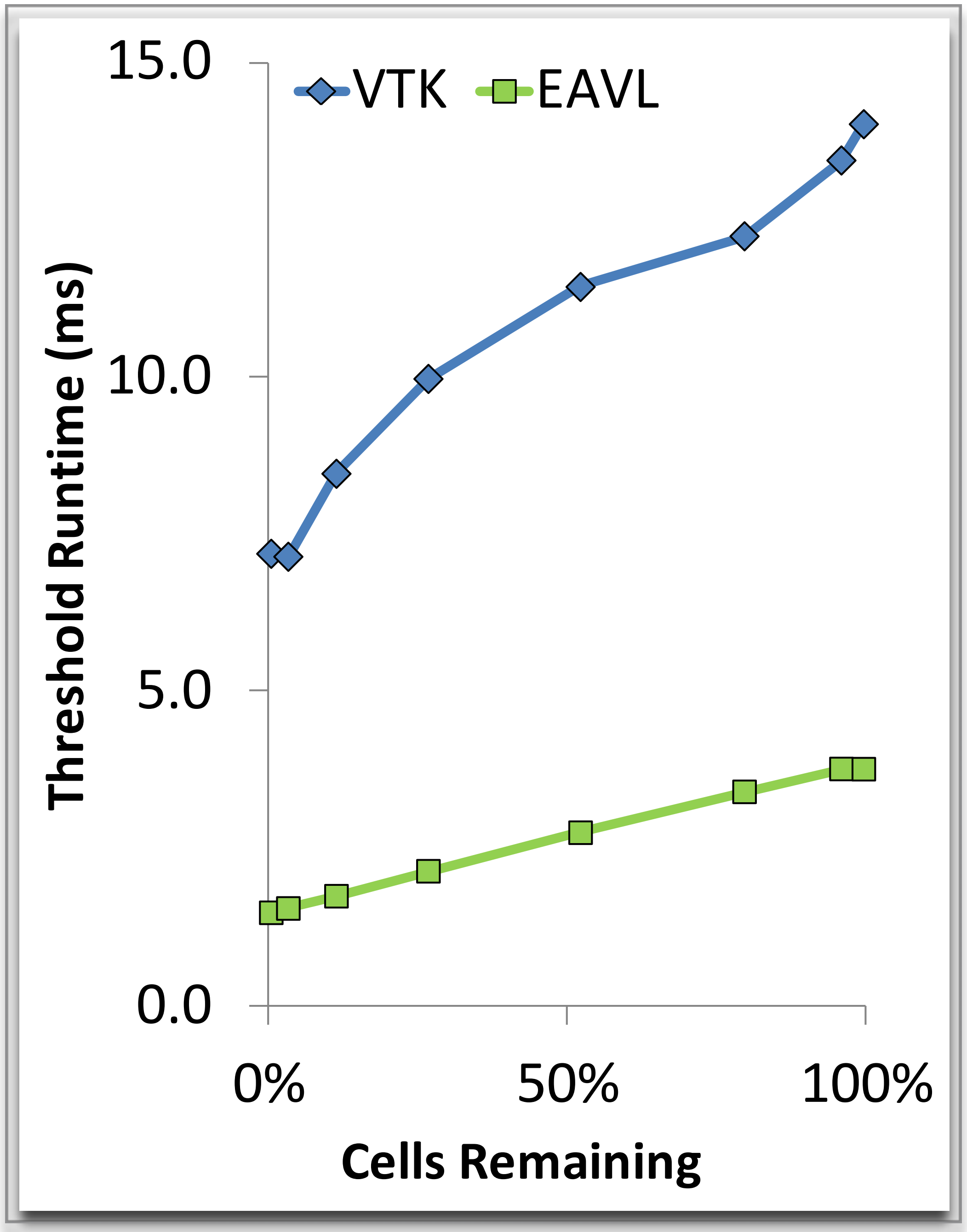

| **Traditional Data Model** | **EAVL Data Model** |
|---|---|
| Fully unstructured grid | Hybrid implicit/explicit grid |
| • Explicit points | • Implicit points |
| • Explicit cells | • Explicit cells |

# Memory and Algorithmic Efficiency

## EAVL: 7X reduction in memory usage

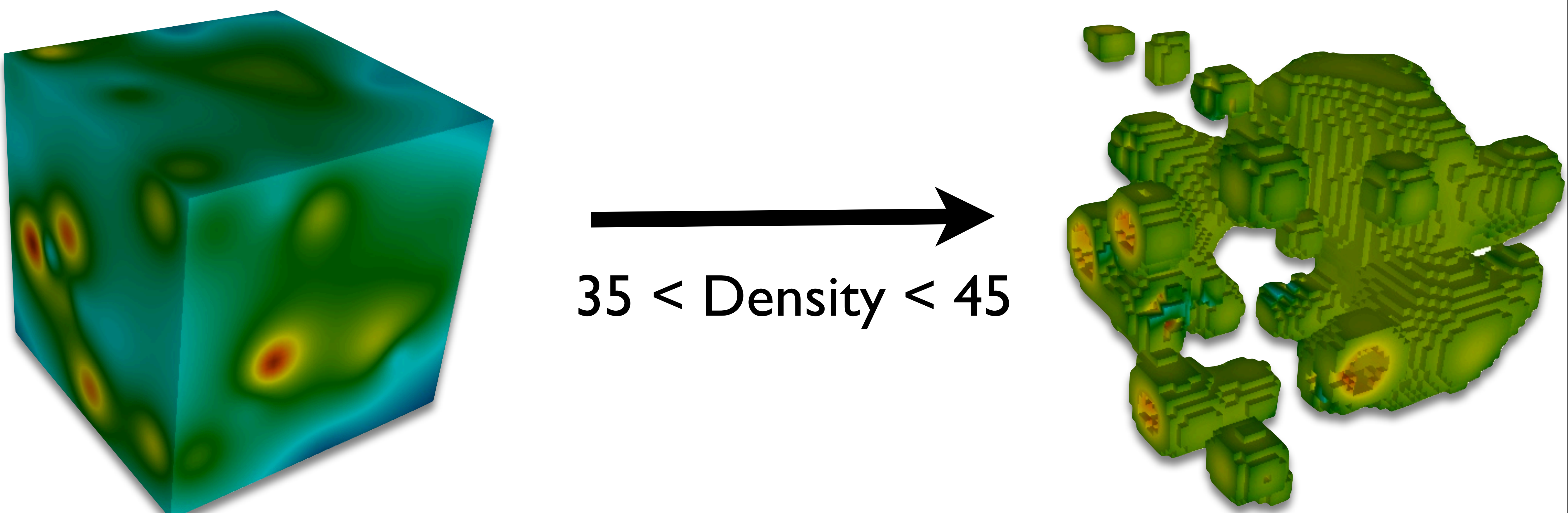



## EAVL: 4-5x performance improvement

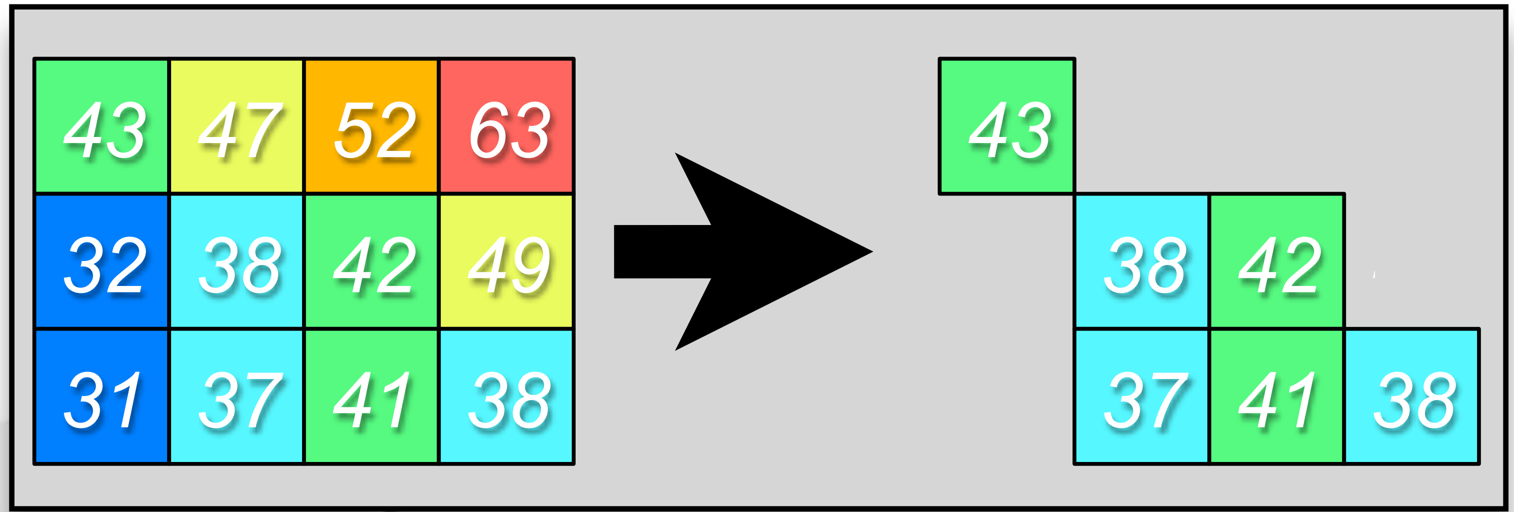


# Data Parallel Programming

- This can be very difficult to do
- Simple example: Threshold operator



# Threshold on a CPU

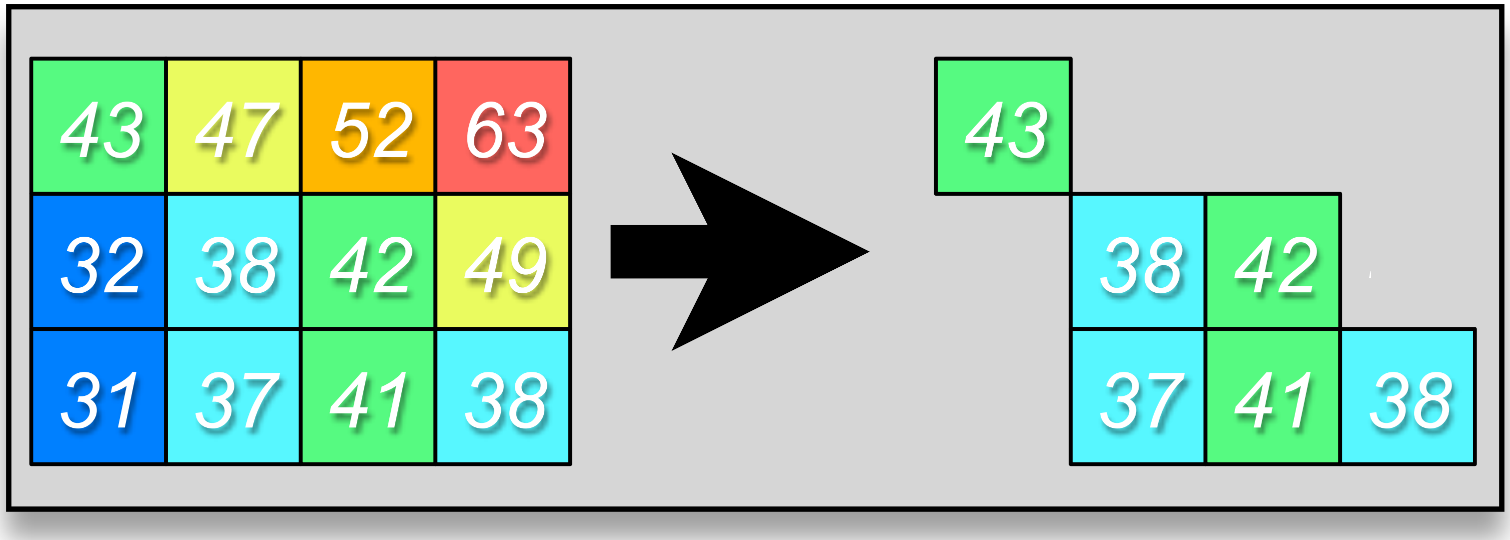



```
mesh = new unstructured mesh
for each cell in Orig_Mesh
{
   if density(cell) in [35,45]
      mesh.AddCell(cell)
}
```

- Data-parallel method is a **VERY** different story
- Following slides courtesy of Jeremy Meredith

OAK RIDGE National Laboratory XVis SDG SDAV OLCF Oak Ridge Leadership Computing Facility

# Threshold on a GPU




- Data-parallel method is a **VERY** different story
- Following slides courtesy of Jeremy Meredith

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **density** | 43 | 47 | 52 | 63 | 32 | 38 | 42 | 49 | 31 | 37 | 41 | 38 |

# Which Cells to Include?

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 |

Evaluate a Map operation with this functor:

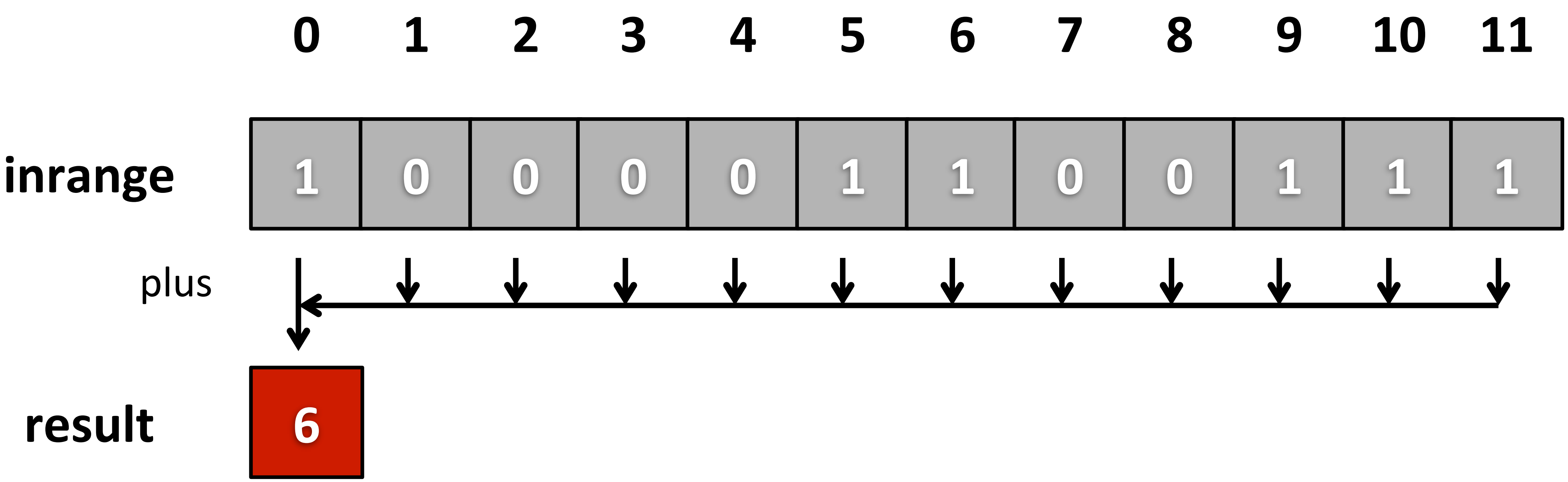
```
struct InRange {
 float lo, hi;
 InRange(float l, float h) : lo(l), hi(h) { }
 int operator()(float x) { return x>lo && x<hi; }
}
```

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **density** | 43 | 47 | 52 | 63 | 32 | 38 | 42 | 49 | 31 | 37 | 41 | 38 |
| InRange() | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| **inrange** | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

# How Many Cells in Output?

Evaluate a Reduce operation using the Add<> functor.

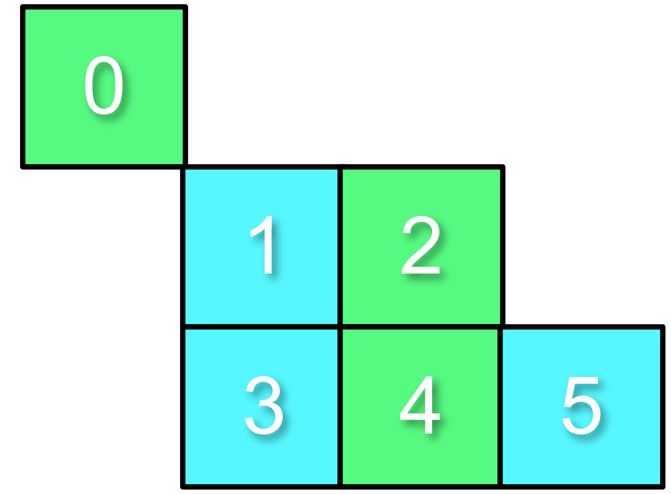
We can use this to create output cell length arrays.

# Where Do the Output Cells Go?


Input indices

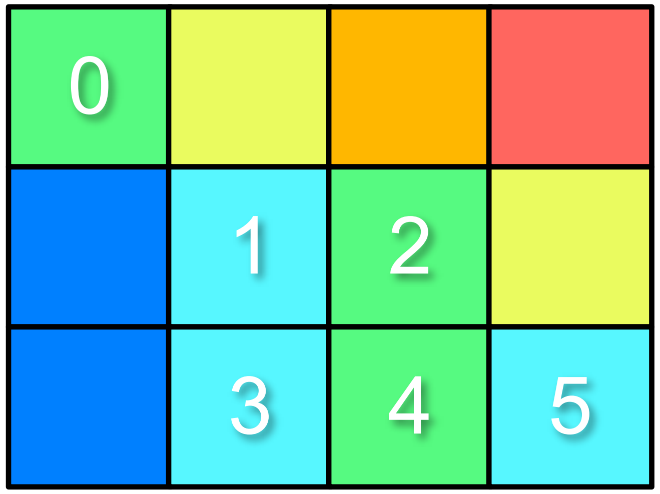
Output indices



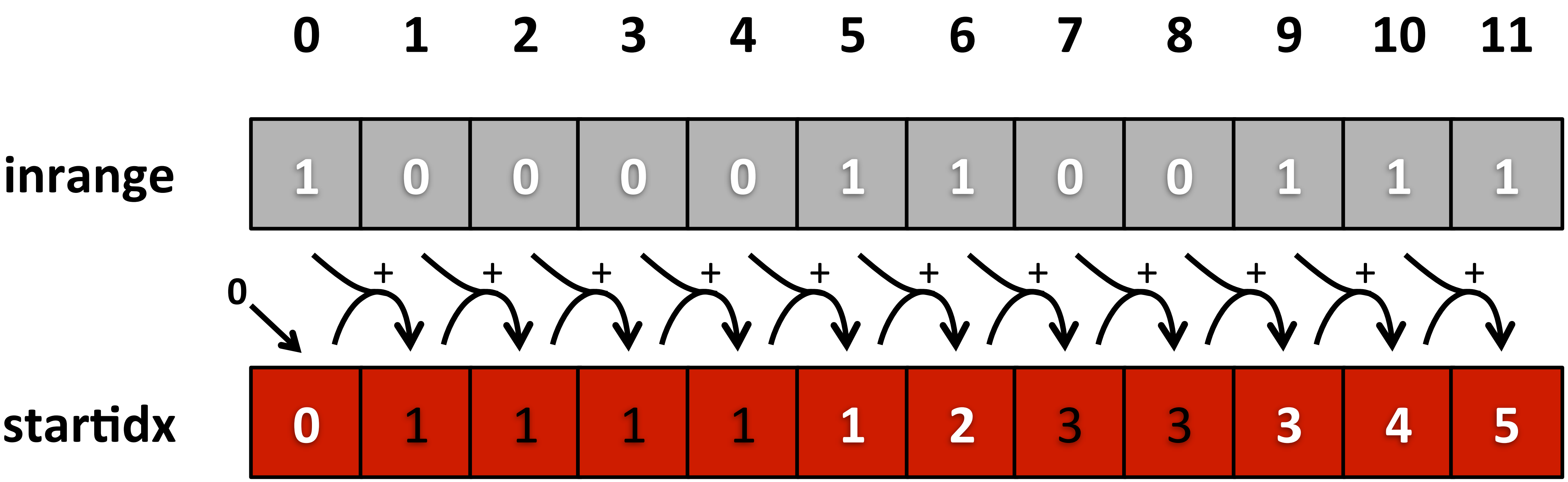

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **input cell** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** |
| **output cell** | **0** | | | | | **1** | **2** | | | **3** | **4** | **5** |

How do we create this mapping?

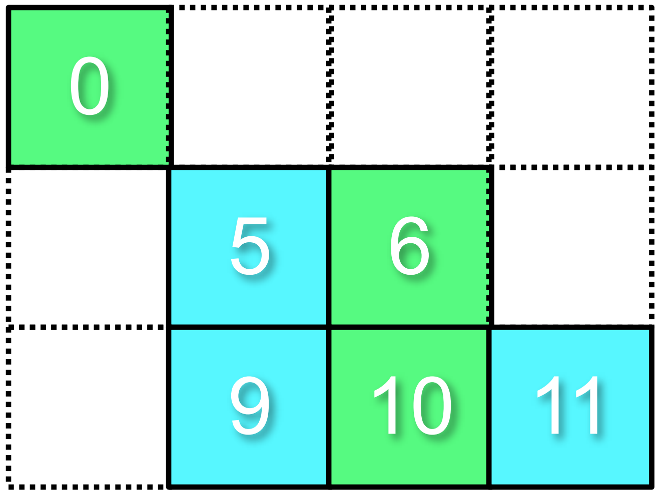
# Create Input-to-Output Indexing?



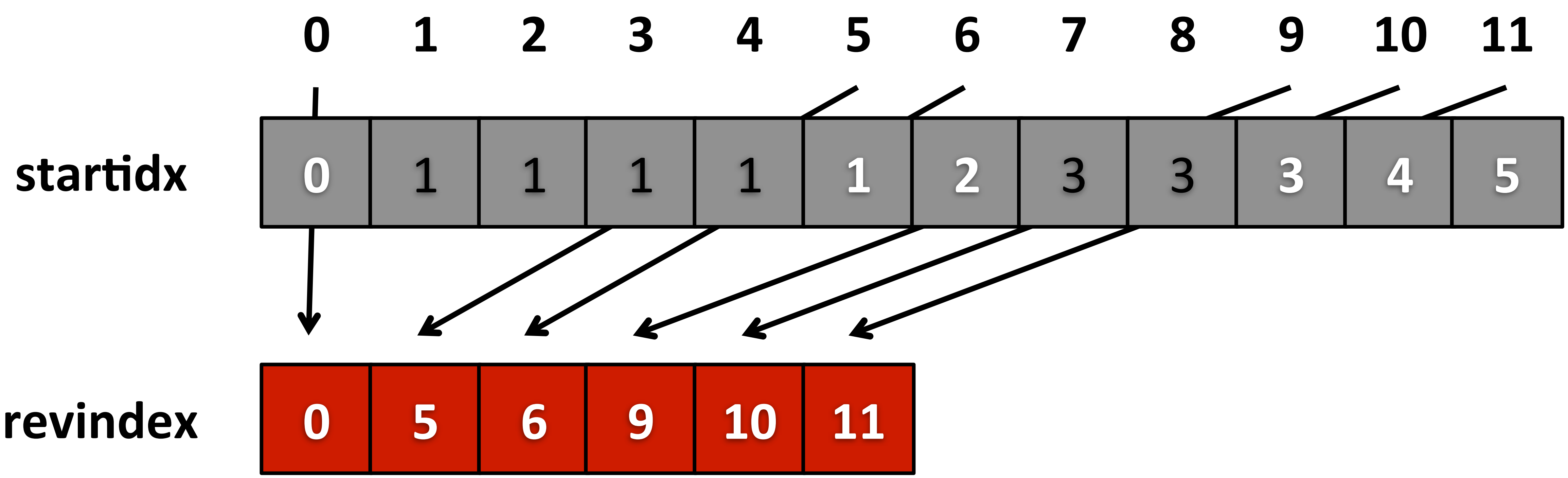

Exclusive Scan (exclusive prefix sum) gives us the output index positions.



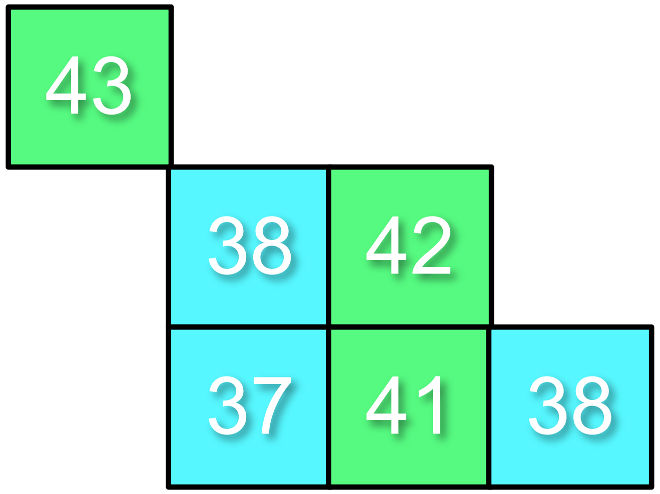
# Create Output-to-Input Indexing?



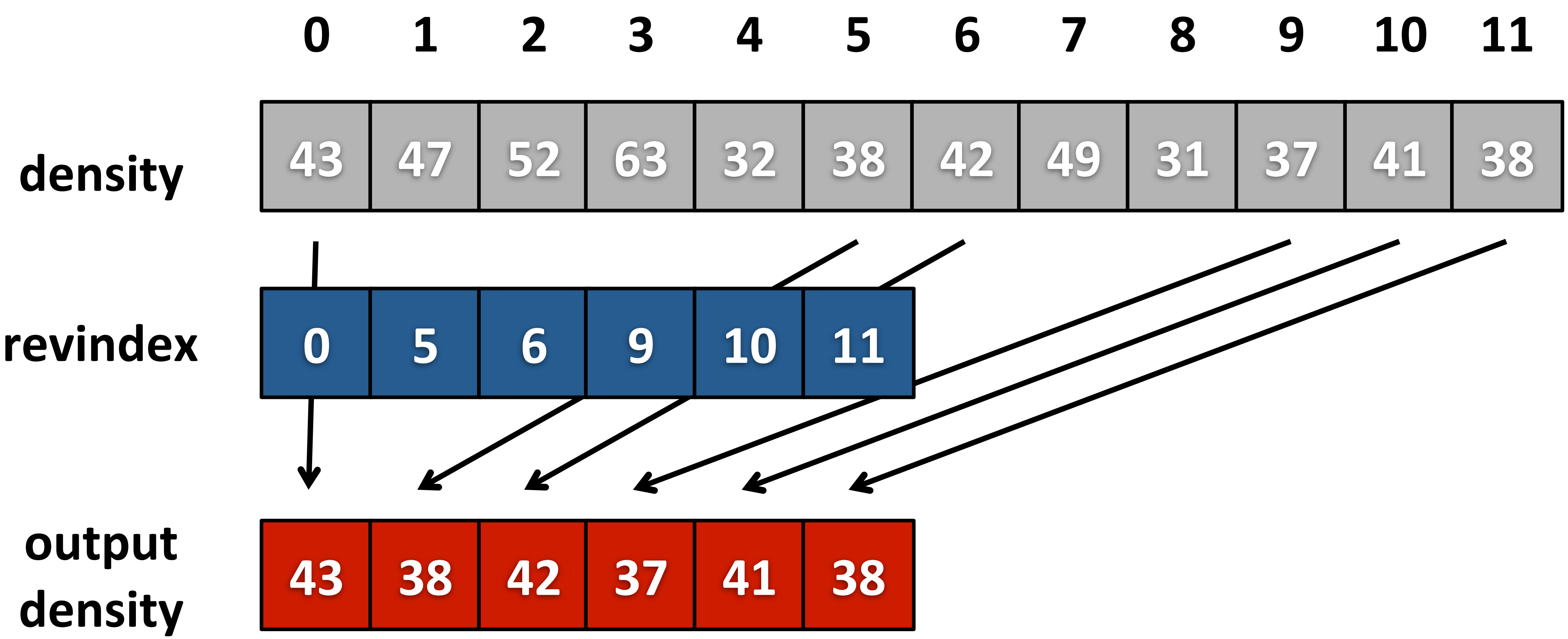

We want to work in the shorter output-length arrays and use gathers. A specialized scatter in EAVL creates this reverse index.



# Gather Input Mesh Arrays to Output?

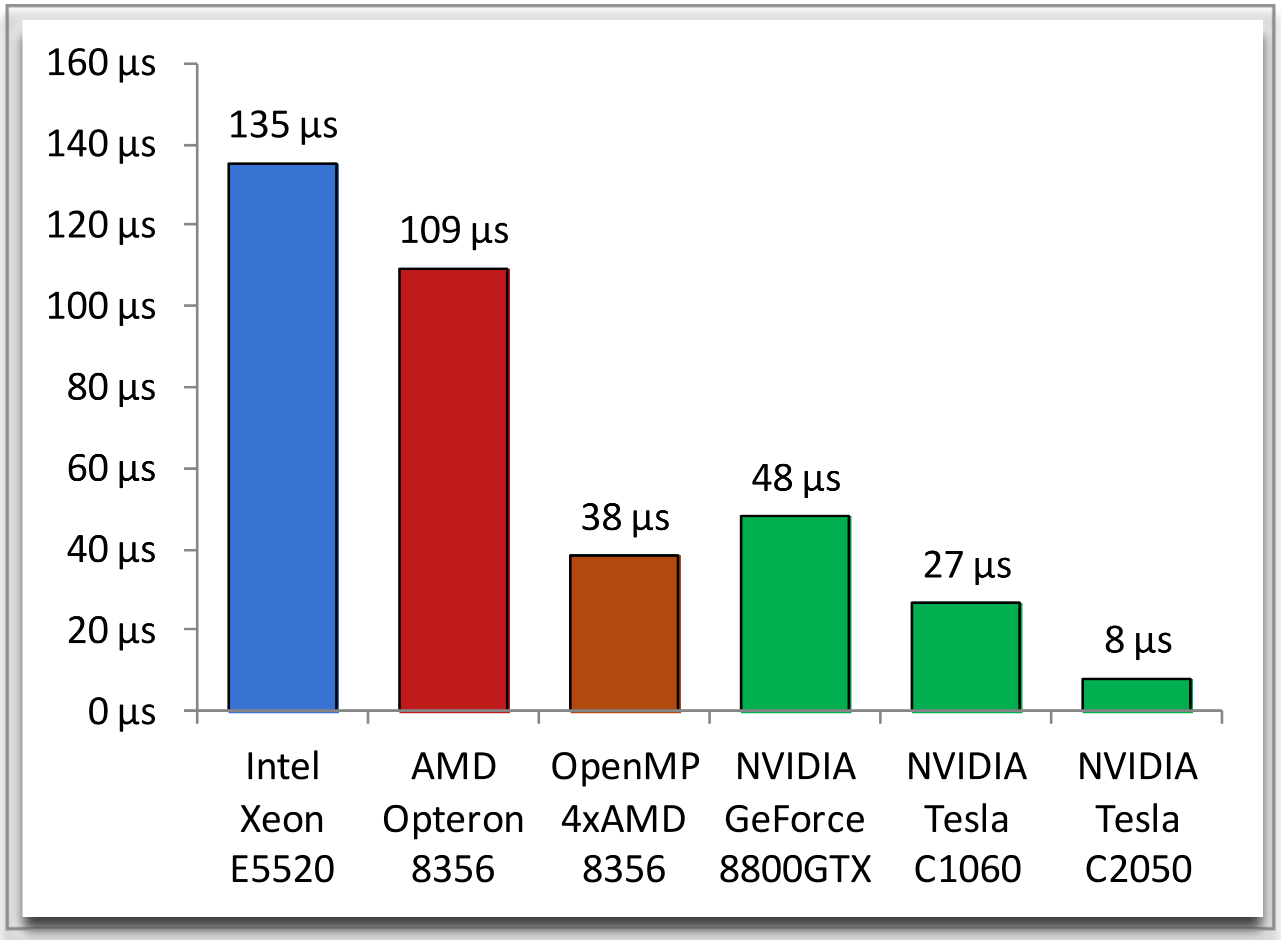



We can now use simple gathers to pull input arrays (density, pressure) into the output mesh.

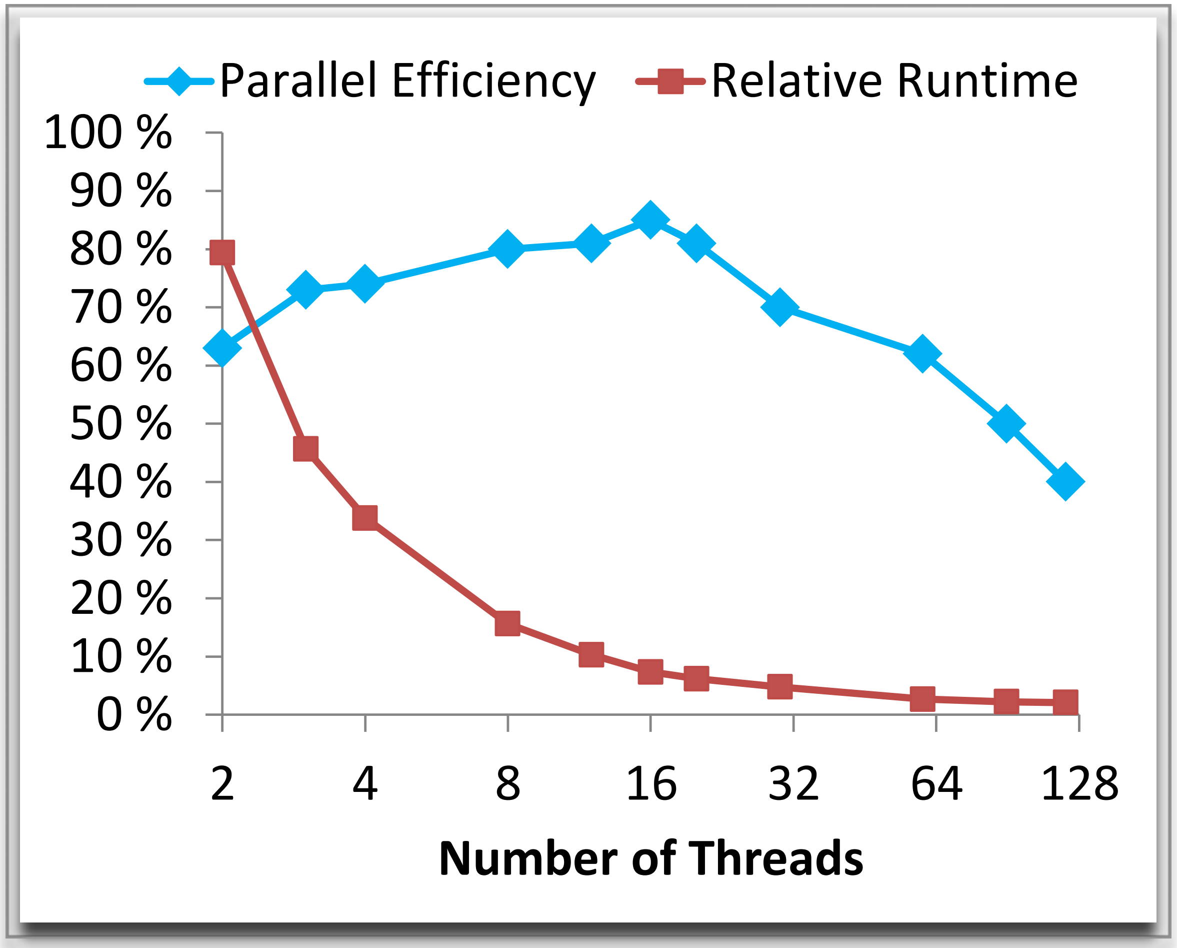


# Heterogeneous Computing

## Runtimes for Surface Normal Calculations




## Surface Normal Scaling on Xeon Phi

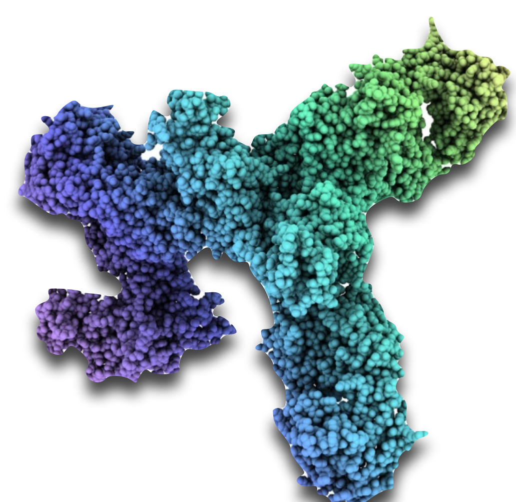



See: "A Distributed Data-Parallel Framework for Analysis and Visualization Algorithm Development", Workshop on General Purpose Processing on Graphics Processing Units (GPGPU5), 2012.

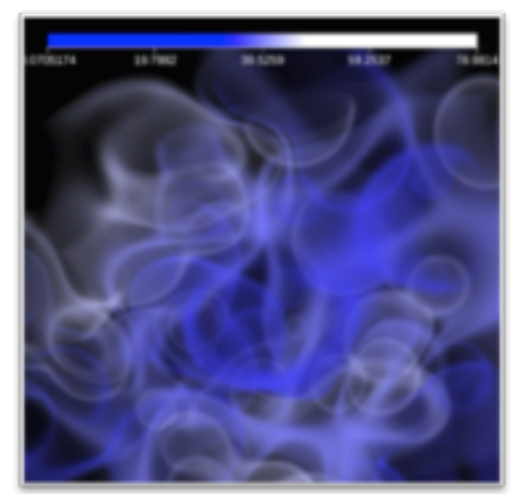
# Advanced Rendering

- Advanced rendering capabilities
  - raster/vector, ray tracing, volume rendering
  - all GPU accelerated using EAVL's data parallel API
  - parallel rendering support via MPI and IceT
- Examples: ambient occlusion lighting effects highlight subtle shape cues for scientific understanding
- Example: direct volume rendering achieves high accuracy images with GPU-accelerated performance

Shear-wave perturbations in SPECFEM3D_GLOBAL code



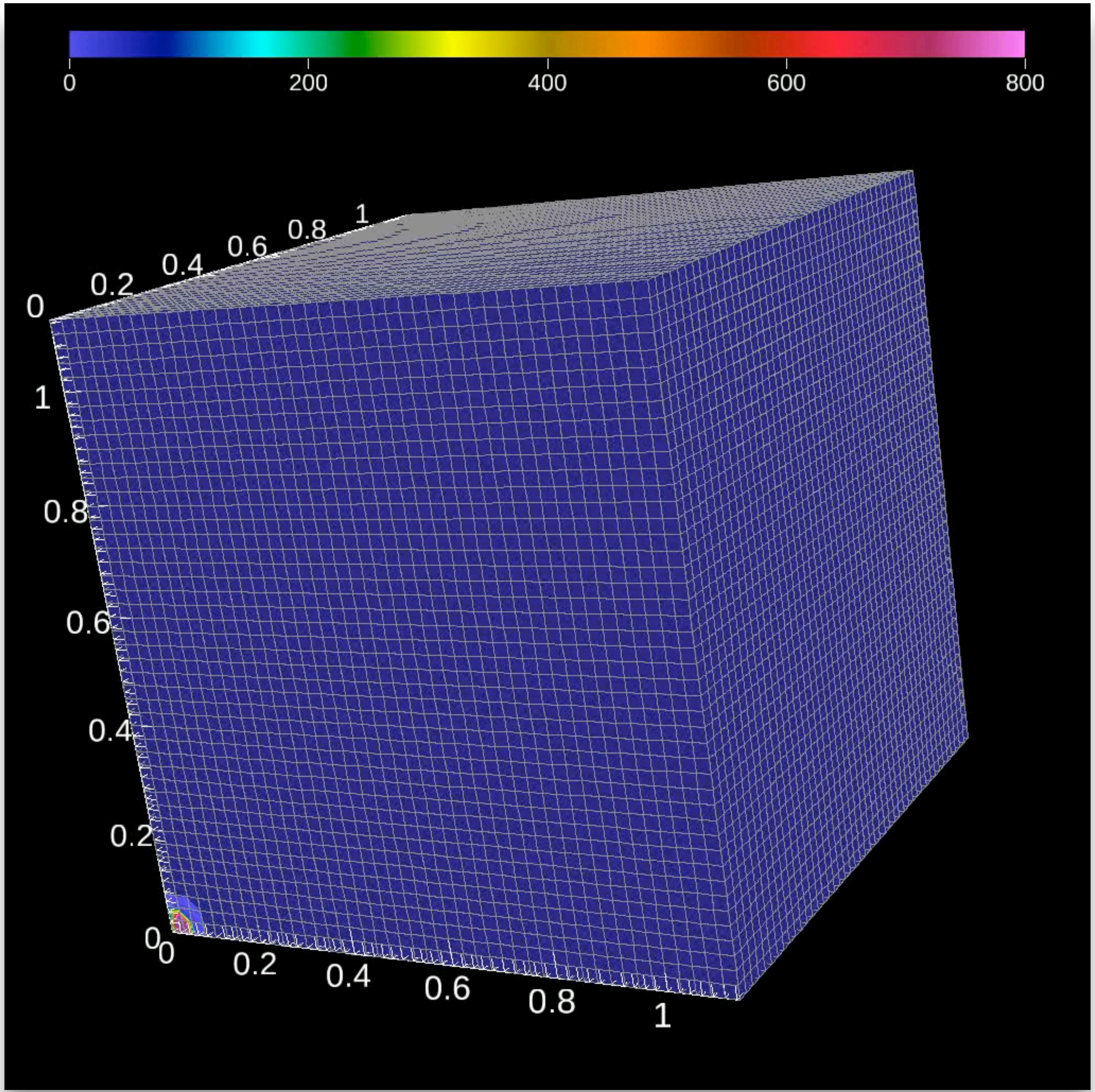
Ebola glycoprotein with proteins from survior



Direct volume rendering from Shepard global interpolant

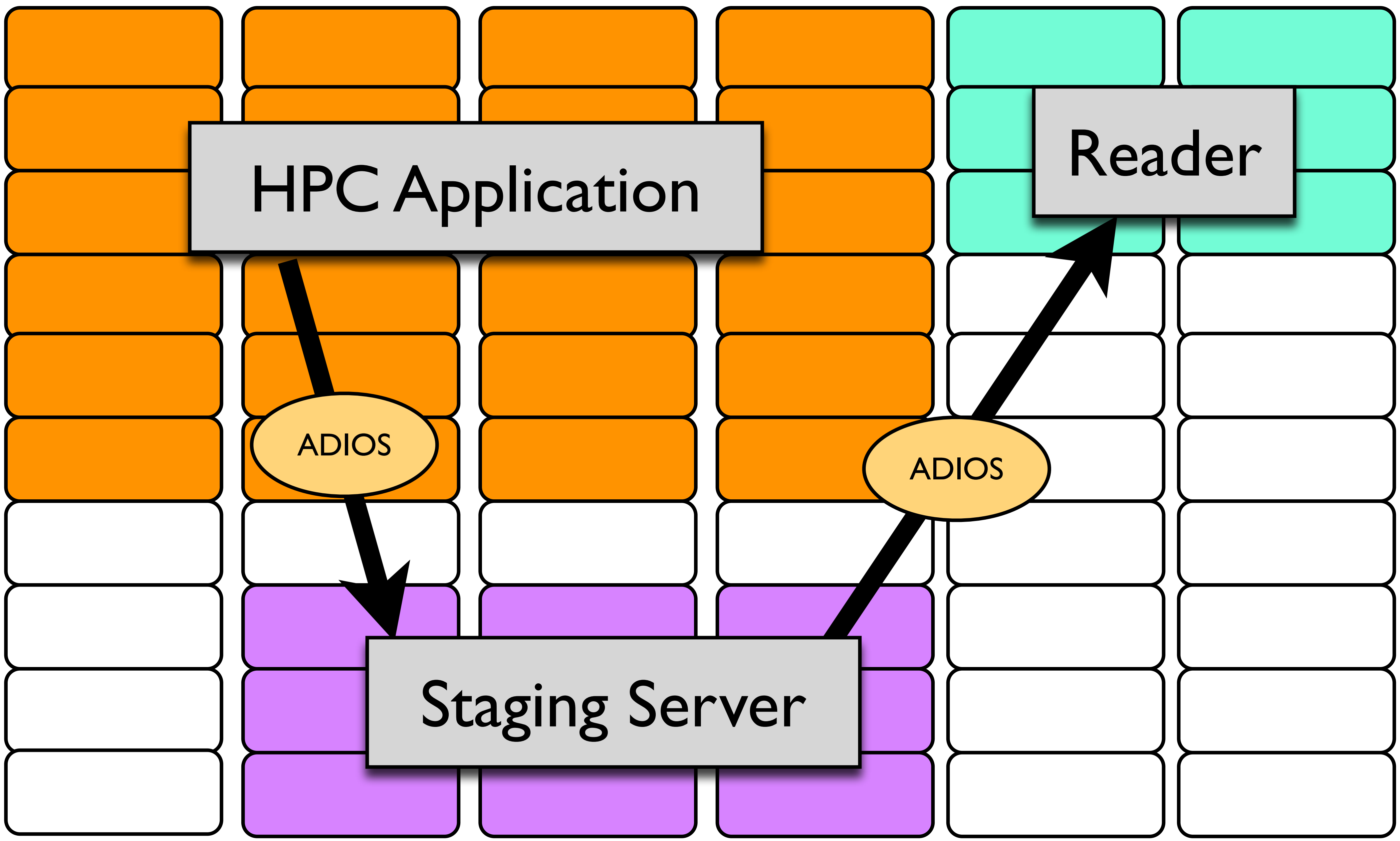
# Tightly-coupled In Situ

- Zero-copy host and device
- Parallel rendering infrastructure
- Examples:
  - LULESH (Hydrodynamics)
  - Xlotal (Fusion)

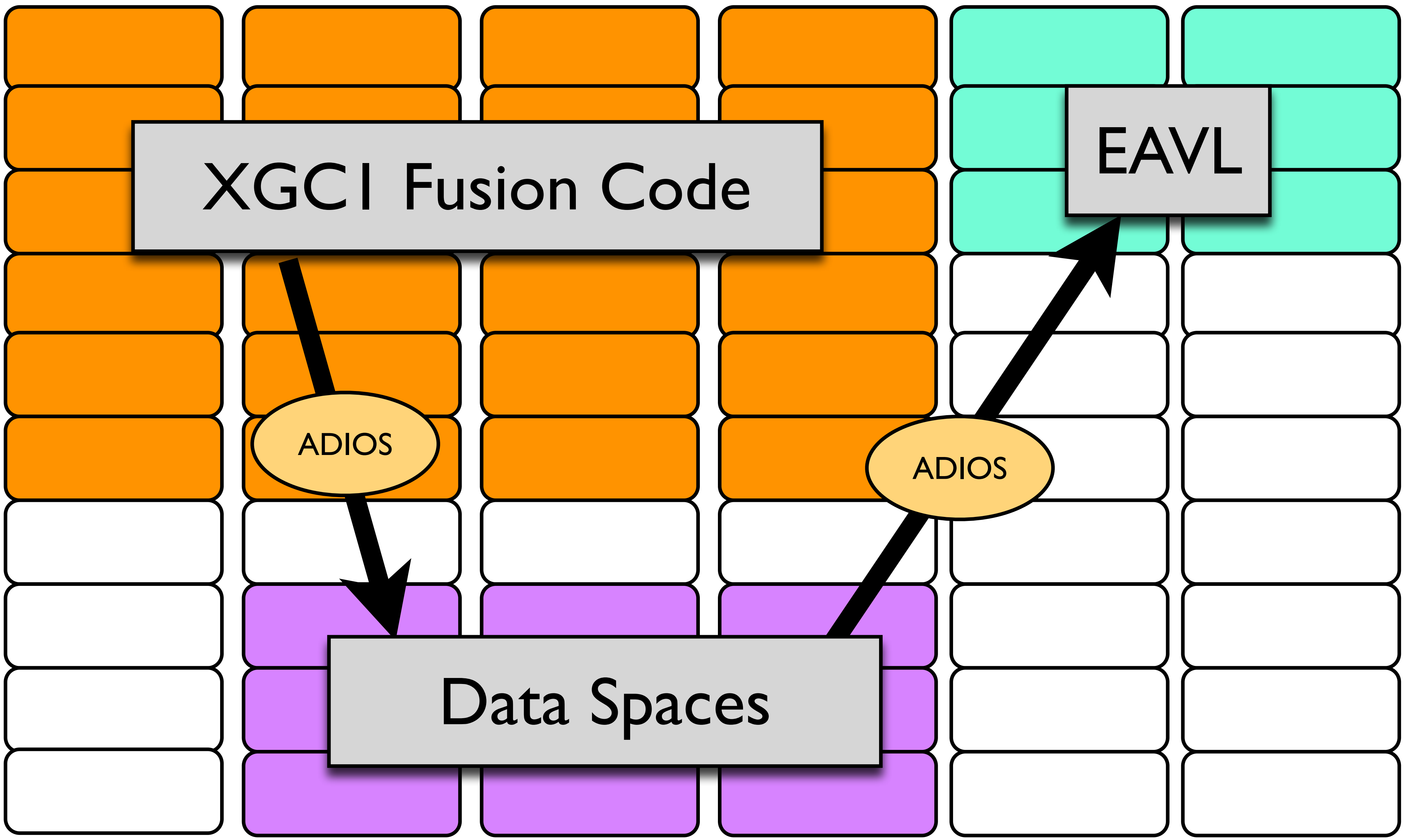



XVis

# Data Staging with ADIOS



# Data Staging with EAVL, ADIOS, and XGC1




OAK RIDGE National Laboratory XVis SDG SDAV OLCF Oak Ridge Leadership Computing Facility

# How to run

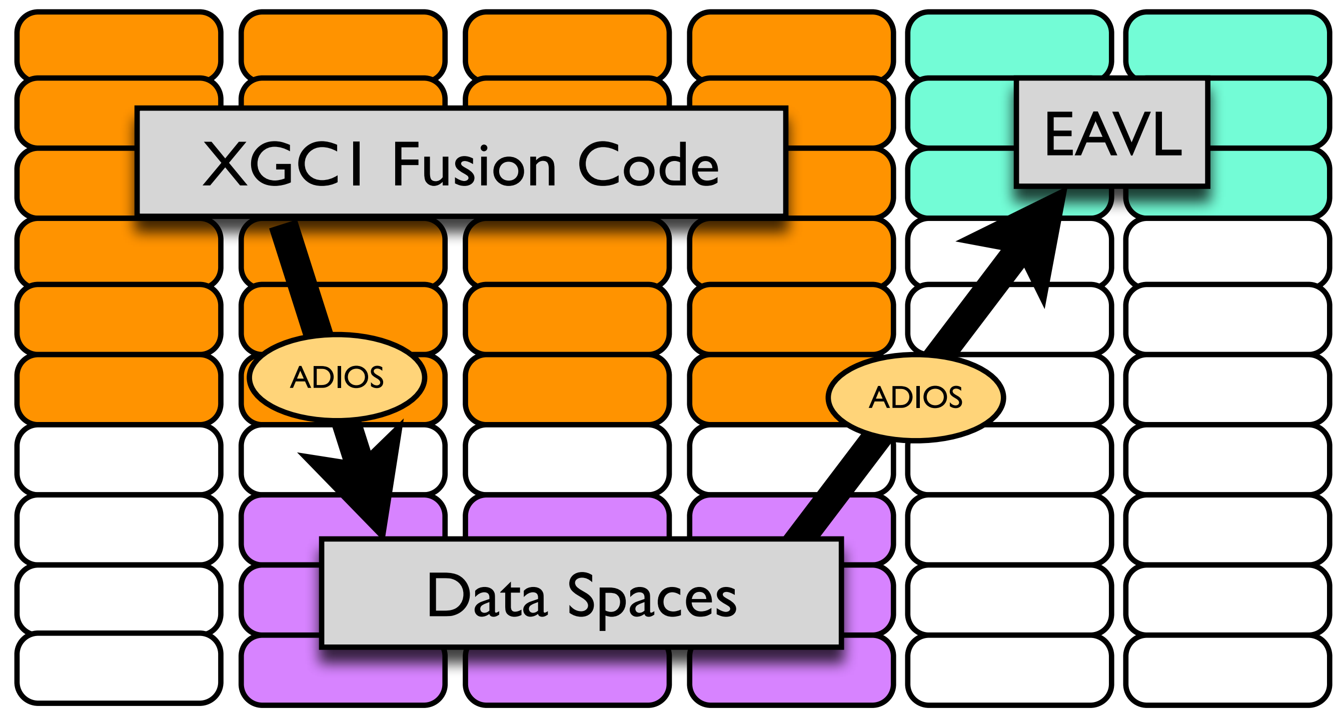
## job script

S = <server configuration>

mpirun -np $N_1$ **app**

mpirun -np $N_2$ **staging_server** S

mpirun -np $N_3$ **vis_app** S



# Source code

## File based visualization

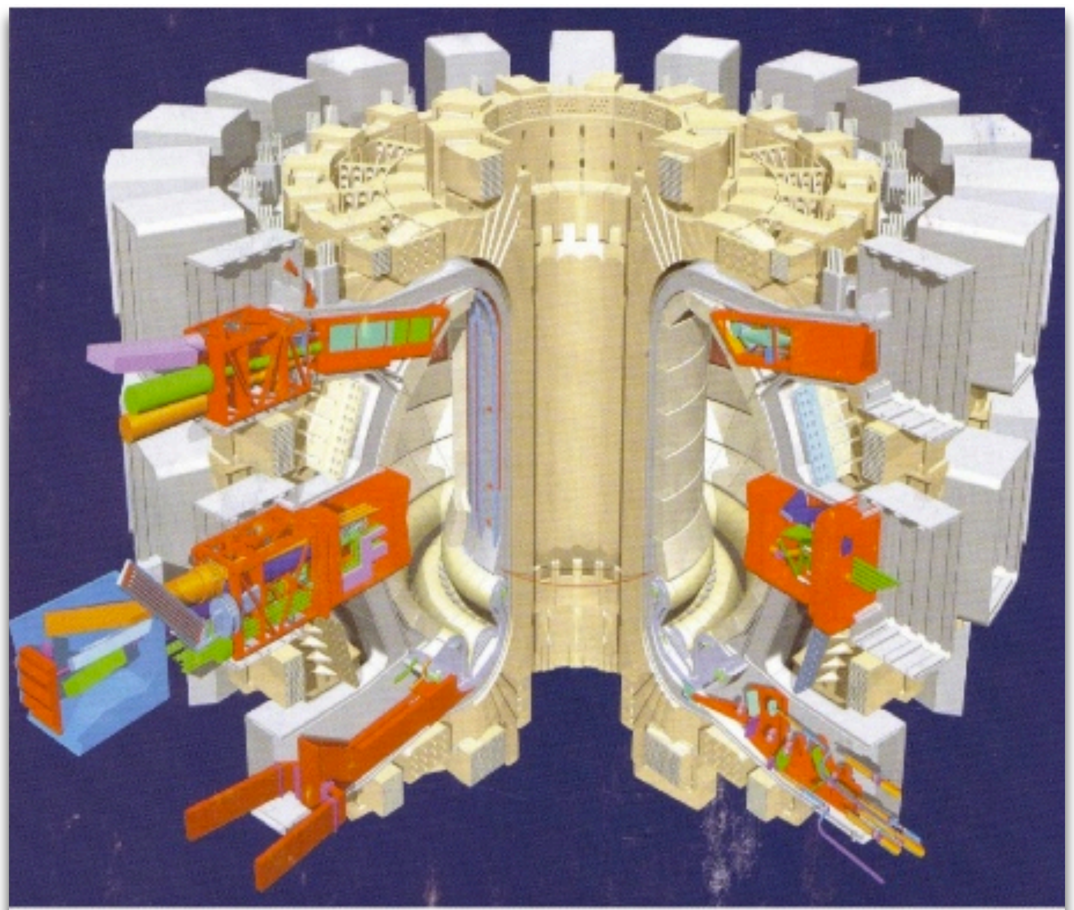
```
f = adios_read_open( "data.bp",
     READ_METHOD_BP,
     MPIComm);

for t in f->nSteps
      adios_schedule_read(f, ...., &data);
      Do_EAVL_Stuff(data);

adios_read_close(f);
```

## Staging based visualization

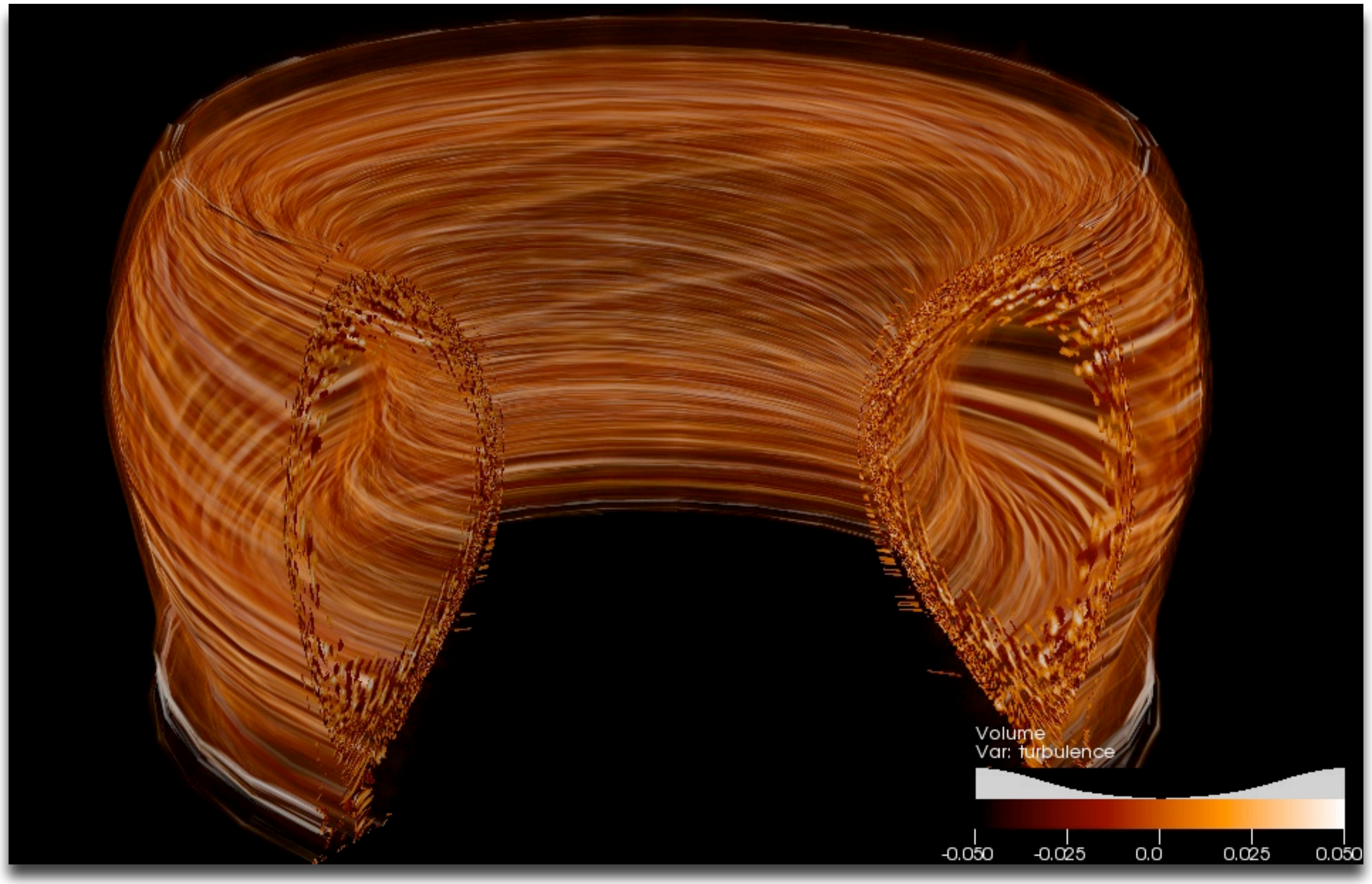
```
f = adios_read_open( "data.bp",
     READ_METHOD_DATASPACES,
     MPIComm);

while ! f->endOfStream
    adios_schedule_read(f, ...., &data);
    Do_EAVL_Stuff(data);

adios_read_close(f);
```

# Fusion Test case

- XGC1: gyrokinetic particle code for simulations of tokamak physics

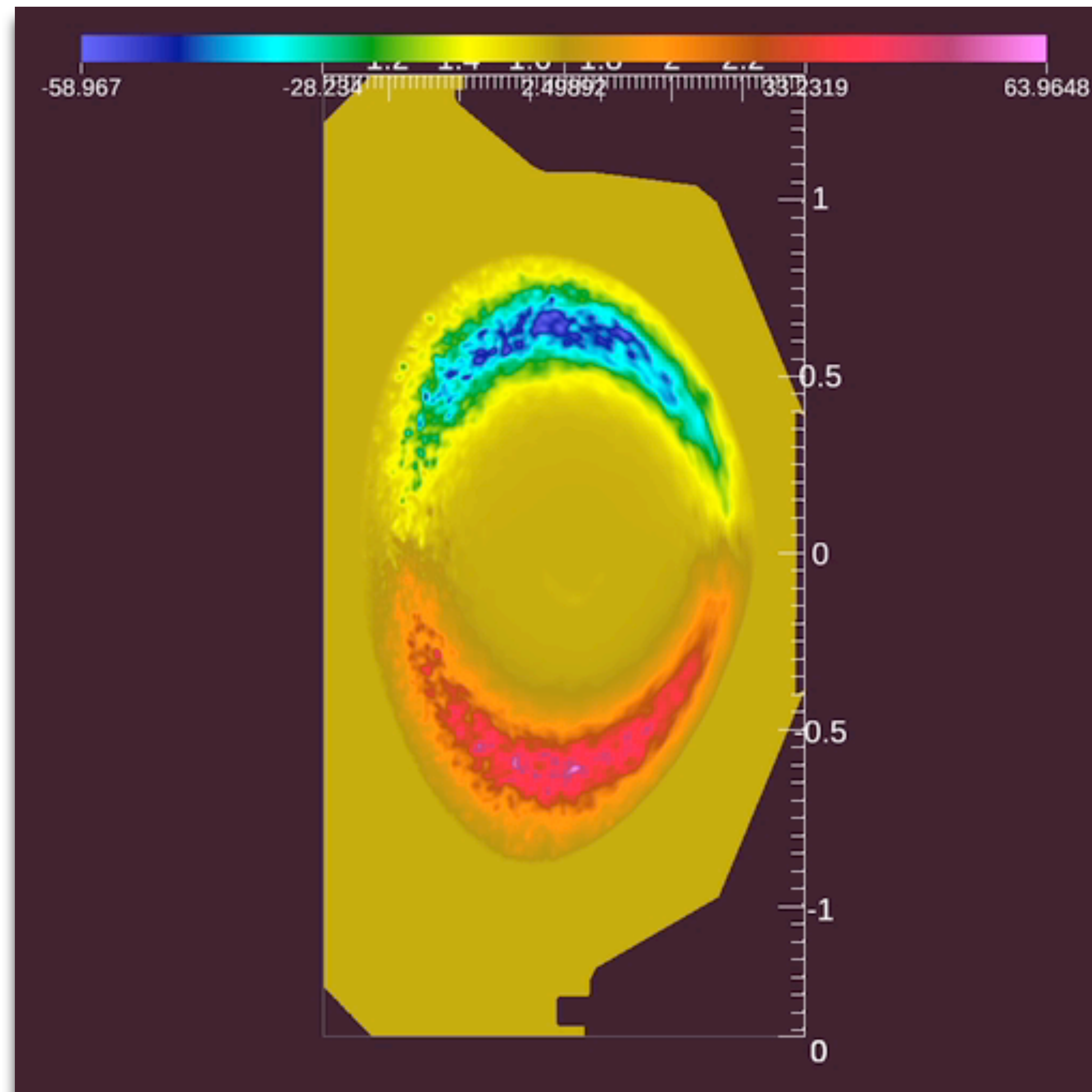


ITER



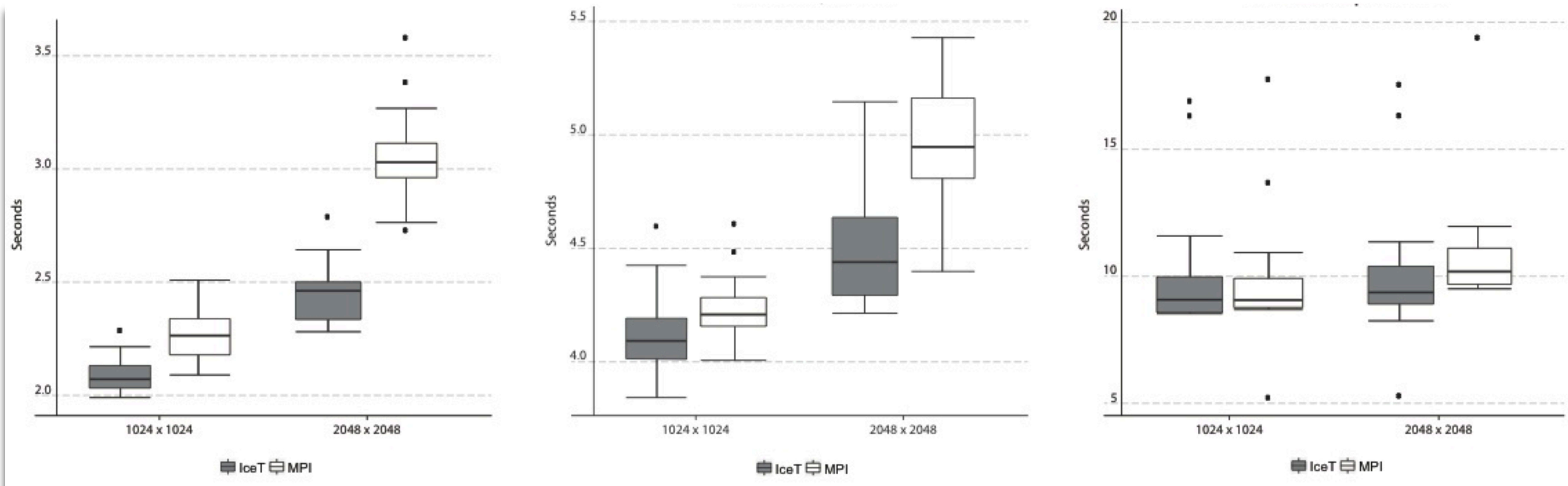
# Loosely coupled In Situ with XGC Field Data

- Application de-coupled from visualization using ADIOS and Data Spaces
  - EAVL plug-in reads data using ADIOS API from staging nodes
  - EAVL plug-in performs visualization operations



# Parallel Rendering of XGC Field Data

Scaling study of parallel rendering of XGC field data using MPI and IceT compositing



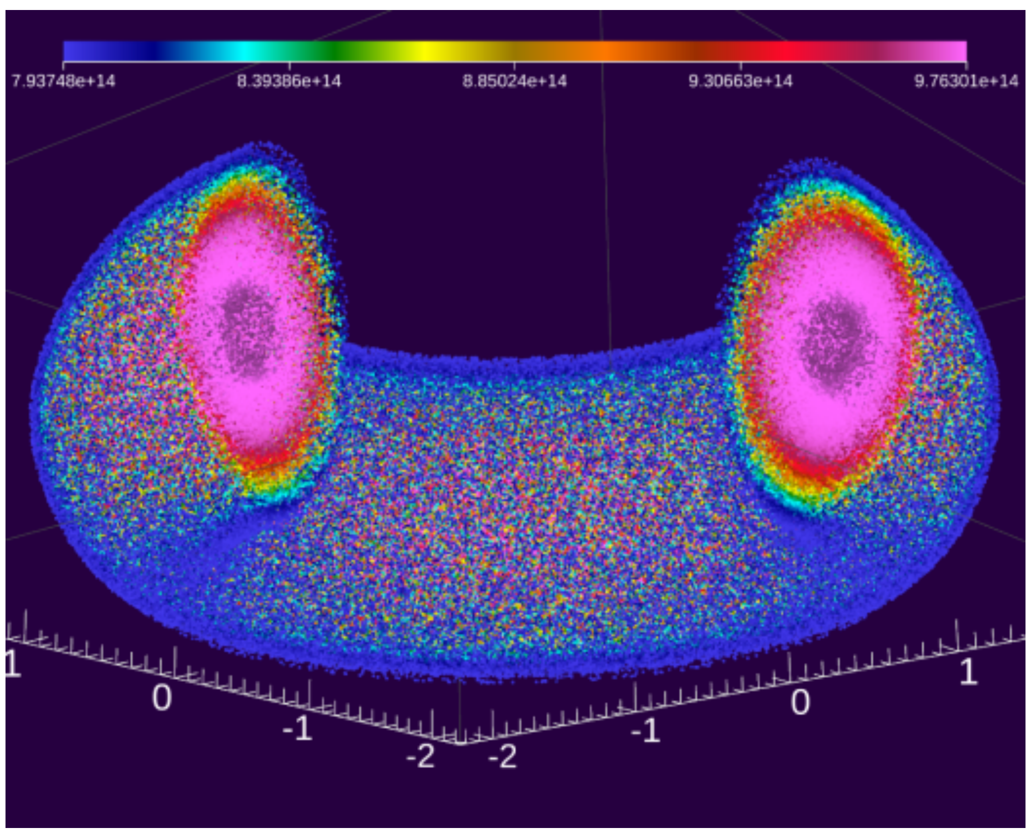

32 tasks 64 tasks 128 tasks

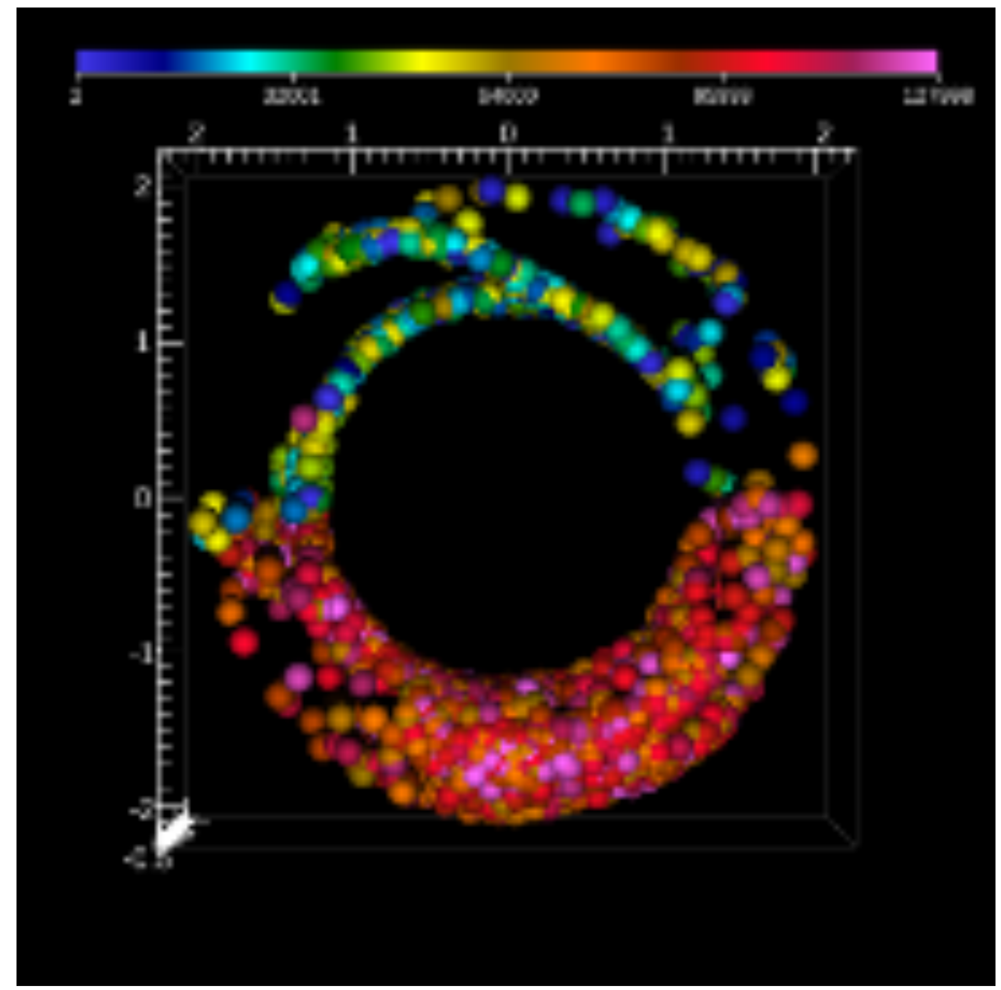
See: "Towards Scalable Visualization Plugins for Data Staging Workflows", SC BDAC Workshop, 2014.

OAK RIDGE National Laboratory XVis SDG SDAV OLCF Oak Ridge Leadership Computing Facility

# Future Work: XGC Particle Data

- Using identical ADIOS, EAVL workflow
- XGC configured to write particles to Data Server
- EAVL plug-in filters particles of interest and renders data



Oak Ridge National Laboratory XVis SDG SDAV OLCF Oak Ridge Leadership Computing Facility

# Conclusions and Future Directions

- EAVL provides a viable path for light weight visualization plugin-ins for in situ environments
  - EAVL to be integrated with Dax and PISTON into new VTK-m initiative
- Early scaling studies show scalability with ADIOS data staging methods
- Extend and study characteristics with different ADIOS staging methods
- Explore ADIOS self describing data streams via the visualization schema
- Continue work in particle visualization

OAK RIDGE National Laboratory XVis SDG SDAV OLCF Oak Ridge Leadership Computing Facility

# Thanks to our funding sources


- Oak Ridge Leadership Computing Facility
- The SciDAC Institute for Scalable Data Mangement, Analysis and Visualization. Funded through DOE ASCR
- XVis. Funded through DOE ASCR