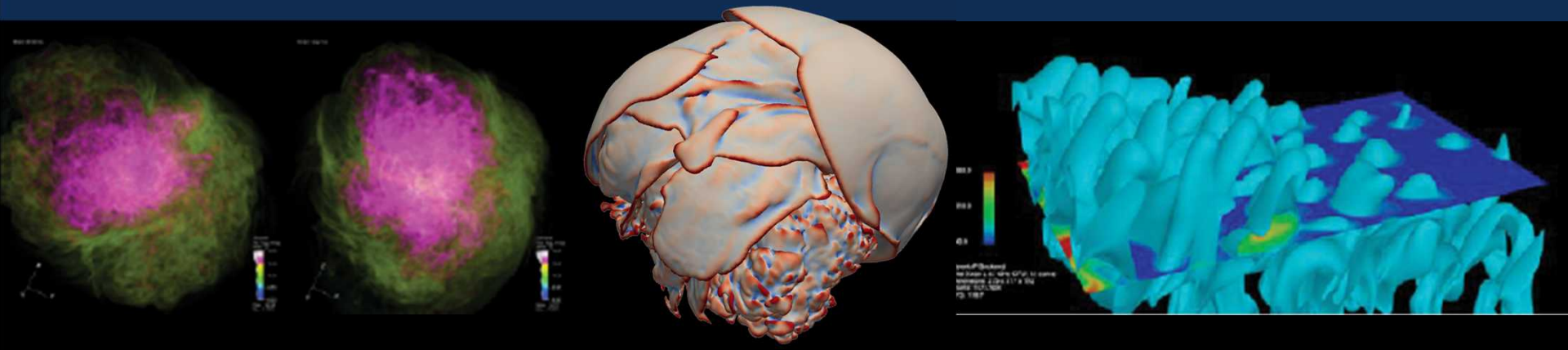
SAND2015-3267PE

Exceptional service in the national interest

Sandia National Laboratories



VTK-m

DOECGF

April 29, 2015

Kenneth Moreland Sandia National Laboratories

With contributions from Jeremy Meredith (ORNL), Chris Sewell (LANL), and Robert Maynard (Kitware)

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. 2011-XXXXP

Supercomputers!

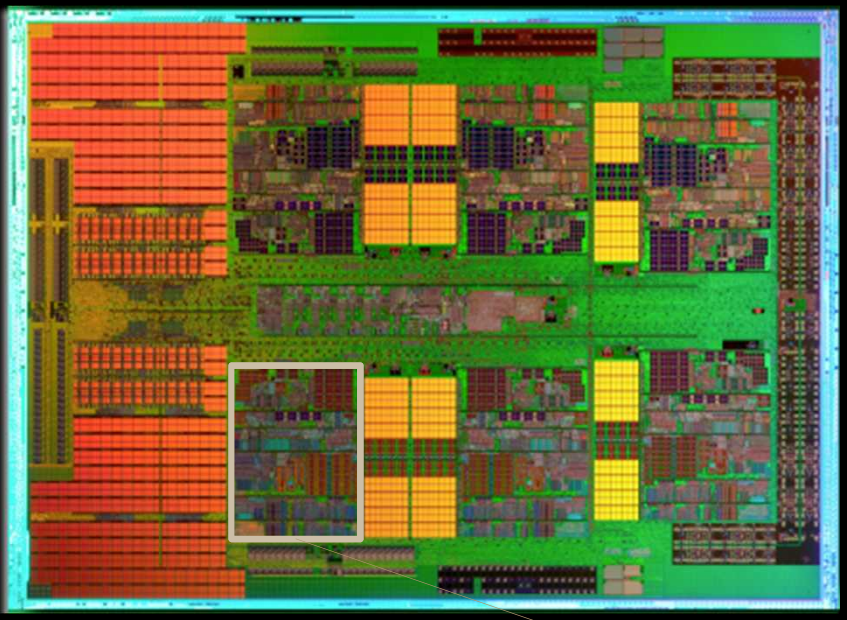
Sandia National Laboratories

- A clear trend in supercomputing is ever increasing parallelism
- Clock increases are long gone
  - "The Free Lunch Is Over" (Herb Sutter)

	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

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

“Everybody who learns concurrency thinks they understand it, ends up finding mysterious races they thought weren’t possible, and discovers that they didn’t actually understand it yet after all.” Herb Sutter



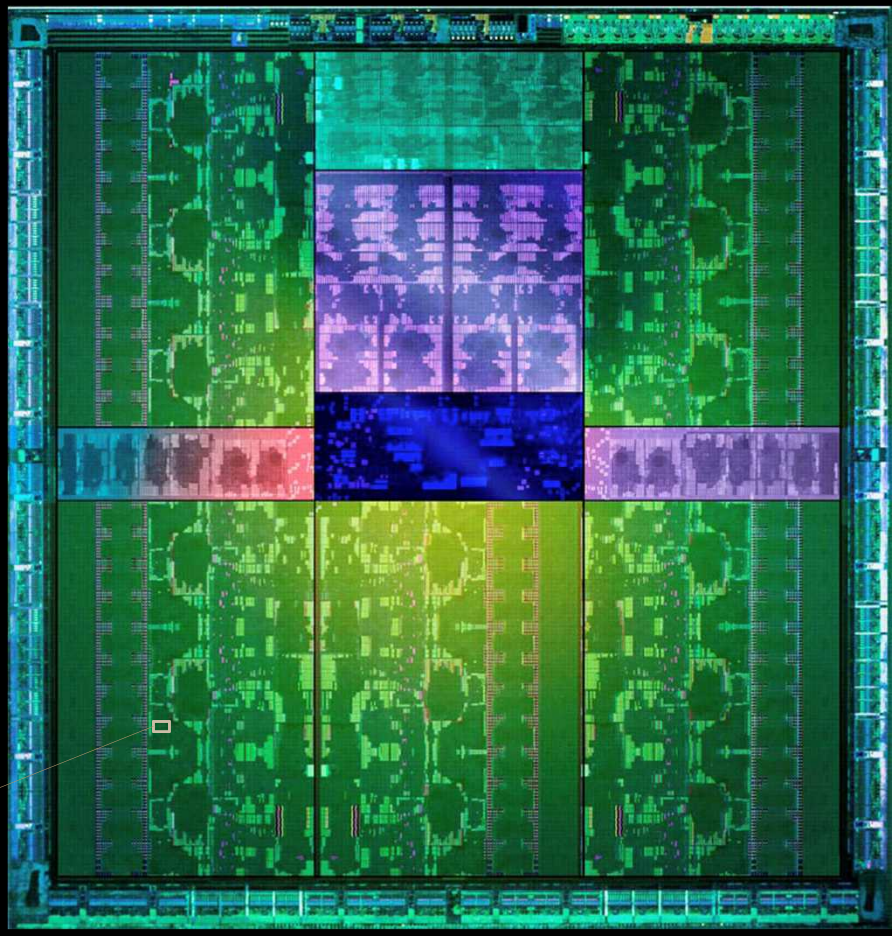
1mm

AMD x86

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

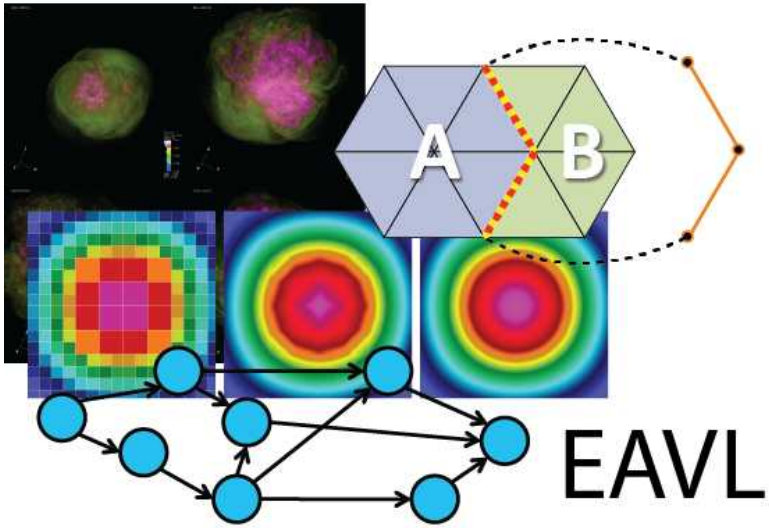
1 x86 core

1 Kepler "core"

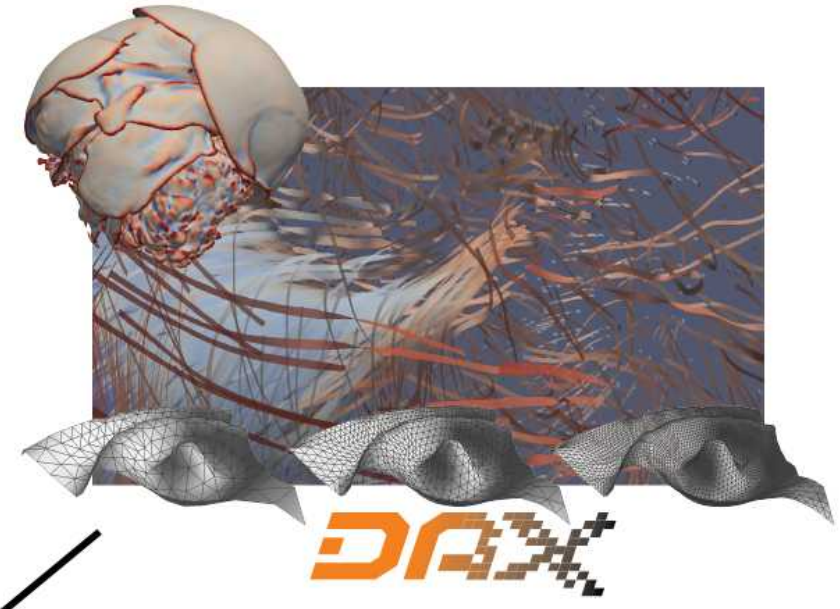


NVIDIA GPU

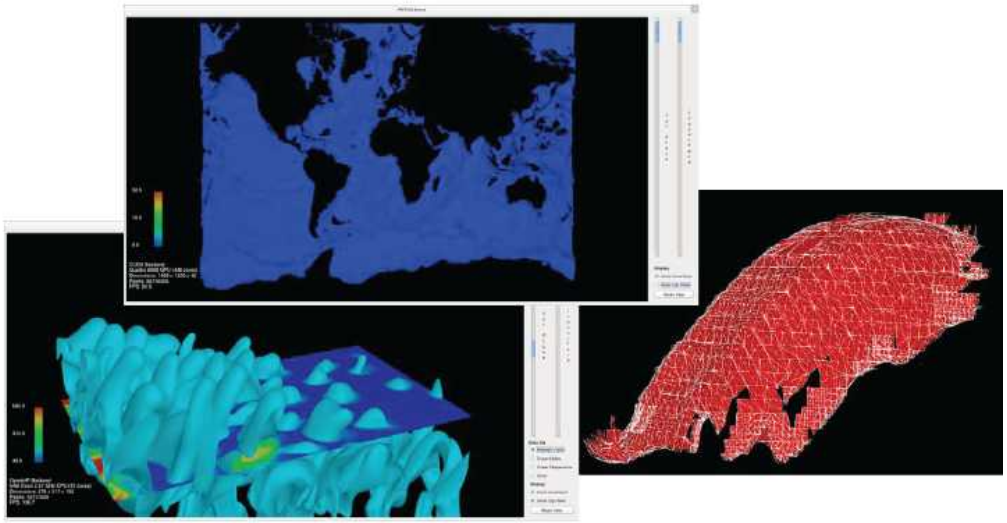
2,880 cores collected in 15 SMX
Shared PC, Cache, Mem Fetches
Reduced control logic
MPI-Only not feasible



EAVL



DAX



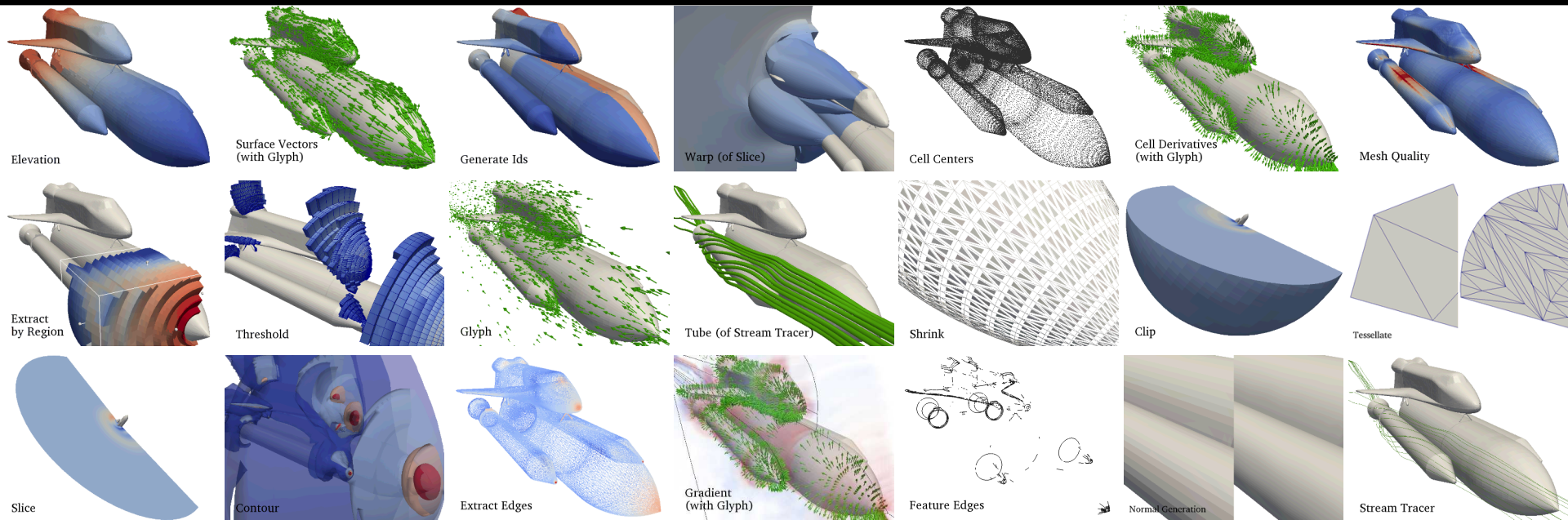
PISTON

VTK-m

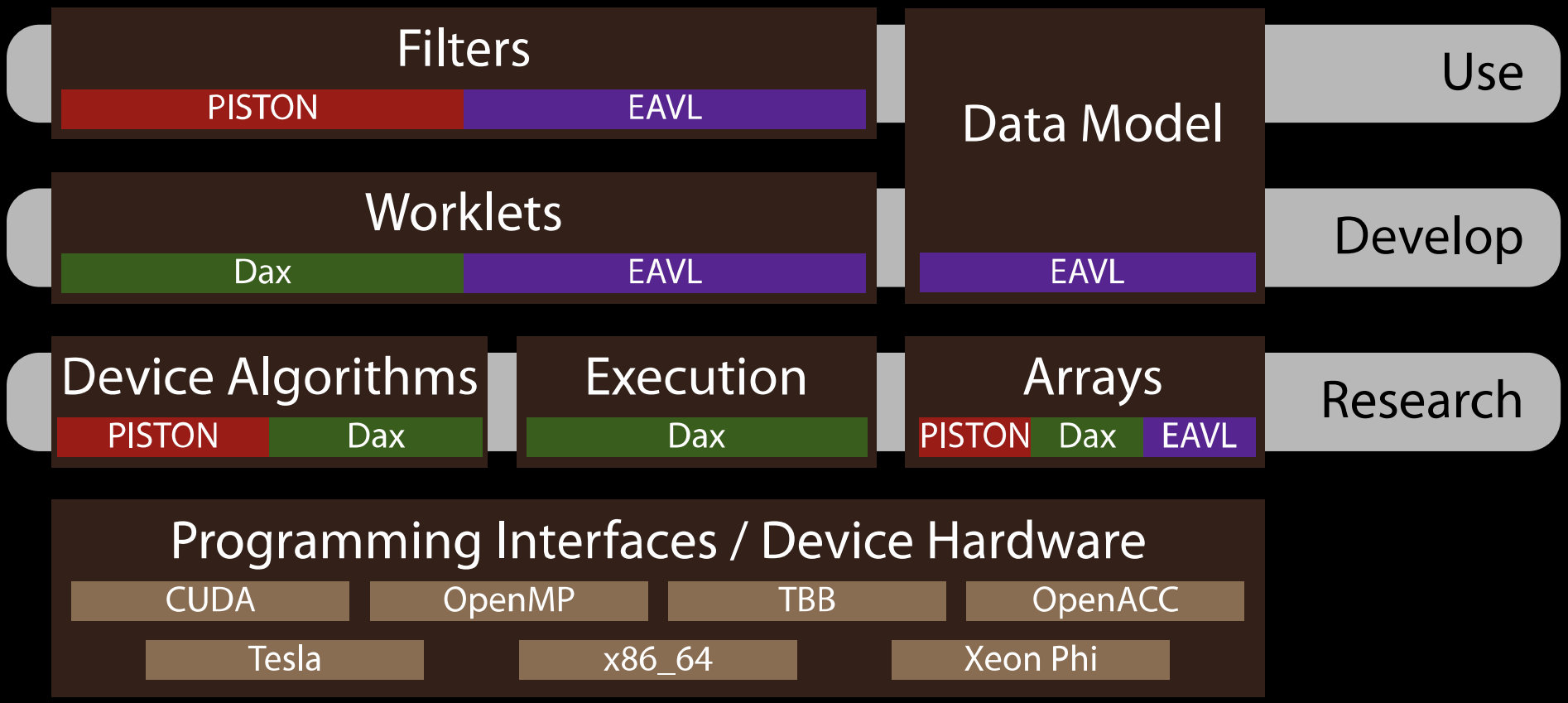
VTK-m Goals

Sandia National Laboratories

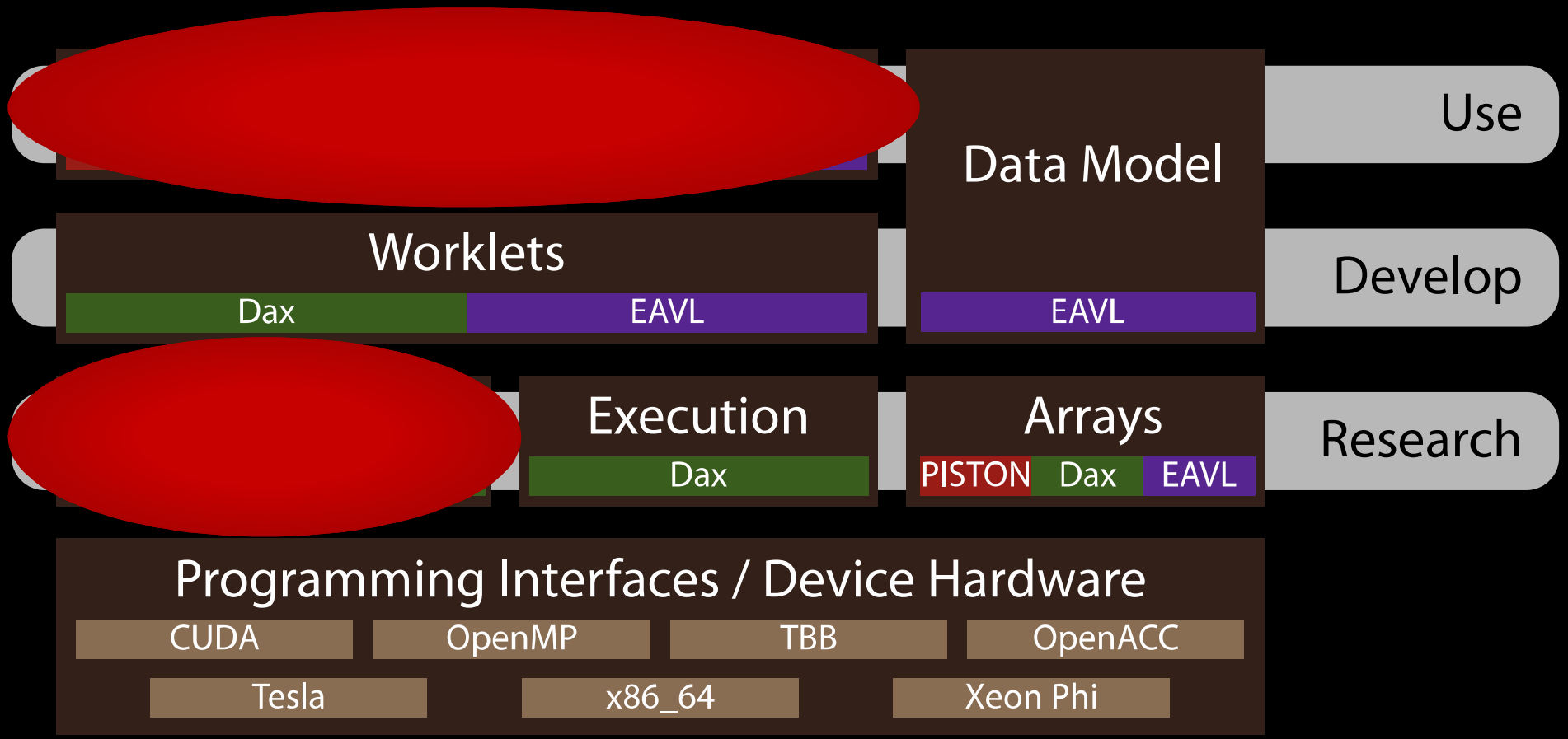
- A single place for the visualization community to collaborate, contribute, and leverage massively threaded algorithms
- Reduce the challenges of writing highly concurrent algorithms by using data parallel algorithms
- Update DOE visualization tools (VTK/ParaView/VisIt) to modern processor technology
- Make these algorithms accessible to simulation codes



VTK-m Architecture



VTK-m Architecture

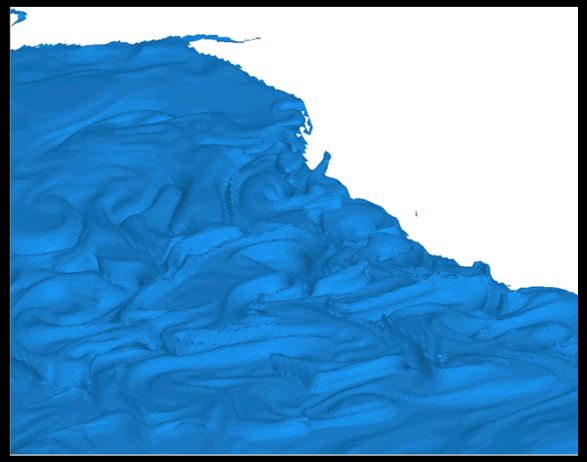


Piston

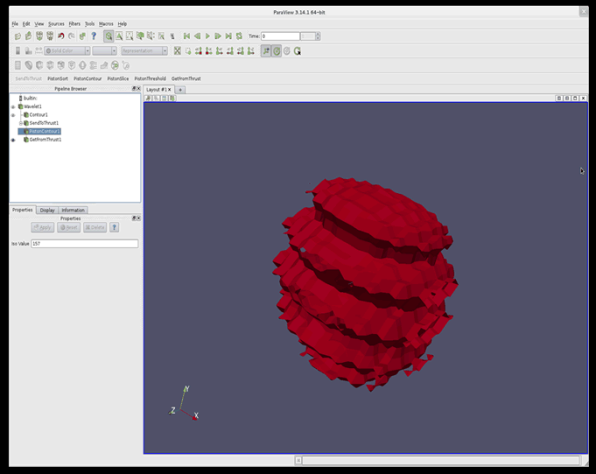
Sandia National Laboratories

- Focuses on developing data-parallel algorithms that are portable across multi-core and many-core architectures for use by LCF codes of interest
- Algorithms are integrated into LCF codes in-situ either directly or though integration with ParaView Catalyst

PISTON isosurface with curvilinear coordinates



Ocean temperature isosurface generated across four GPUs using distributed PISTON



PISTON integration with VTK and ParaView

1. input

transform(classify_cell)

2. caseNums

3. numVertices

transform_inclusive_scan(is_valid_cell)

4. validCellEnum

5. CountingIterator

upper_bound

6. validCellIndices

make_permutation_iterator

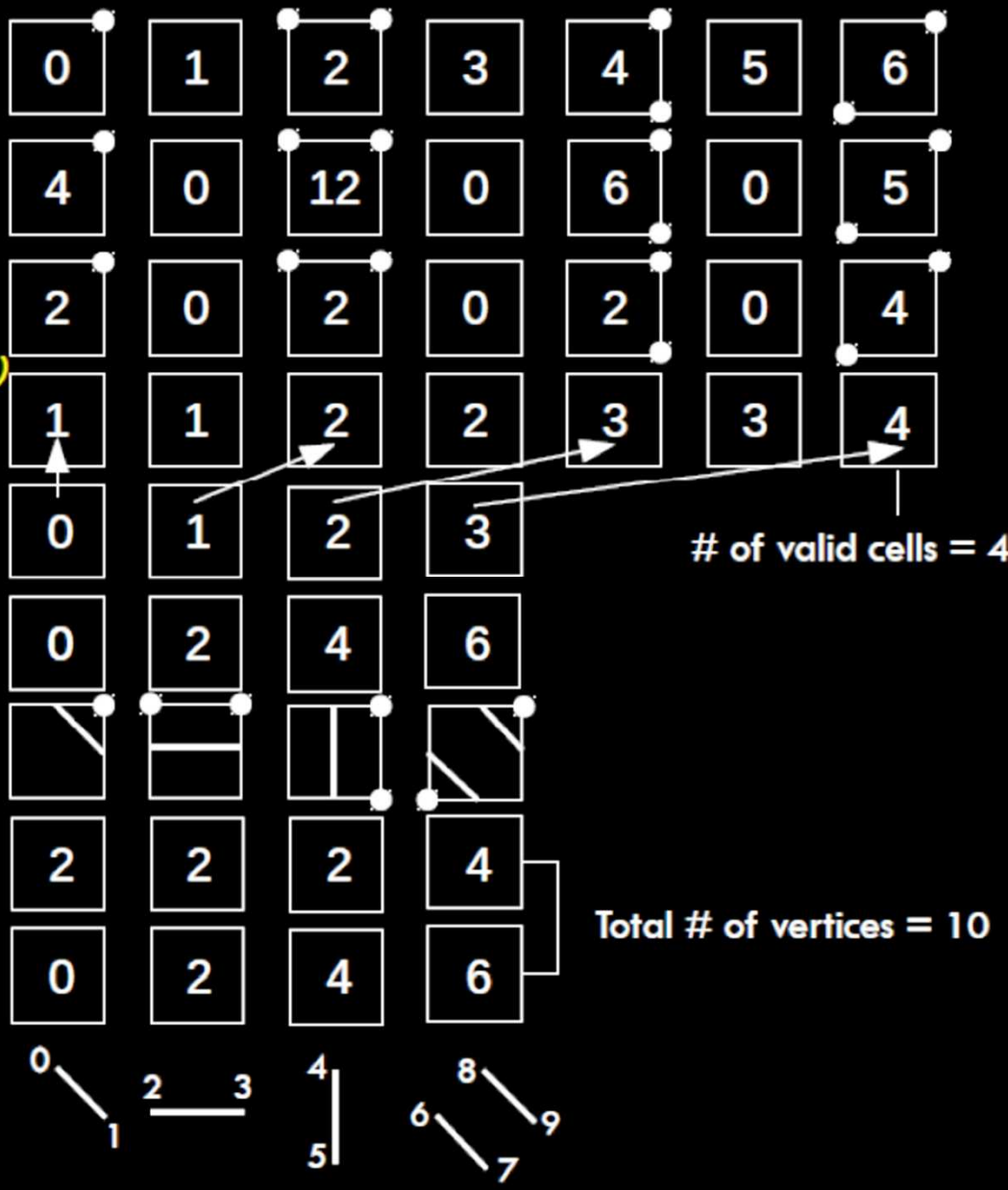
7. numVerticesCompacted

exclusive_scan

8. numVerticesEnum

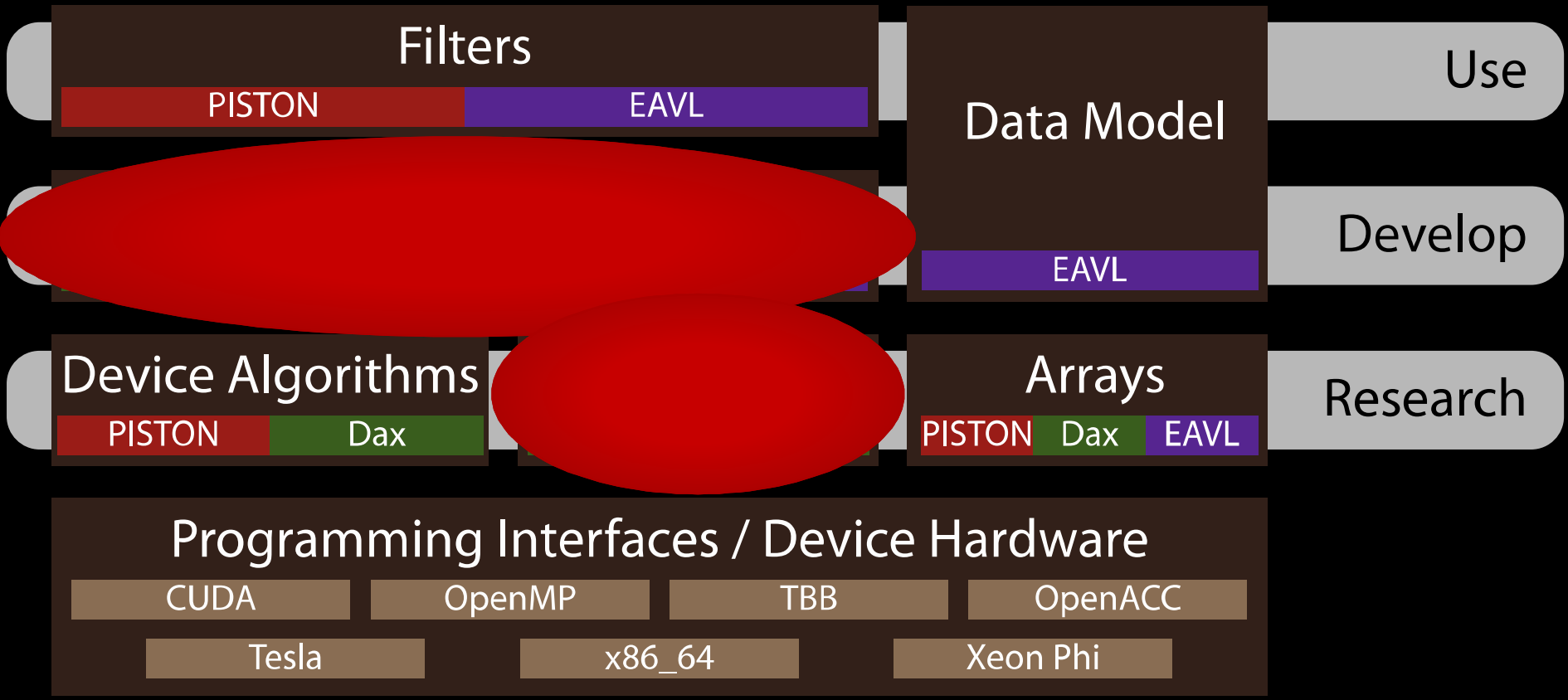
for_each(isosurface_functor)

9. outputVertices



VTK-m Architecture

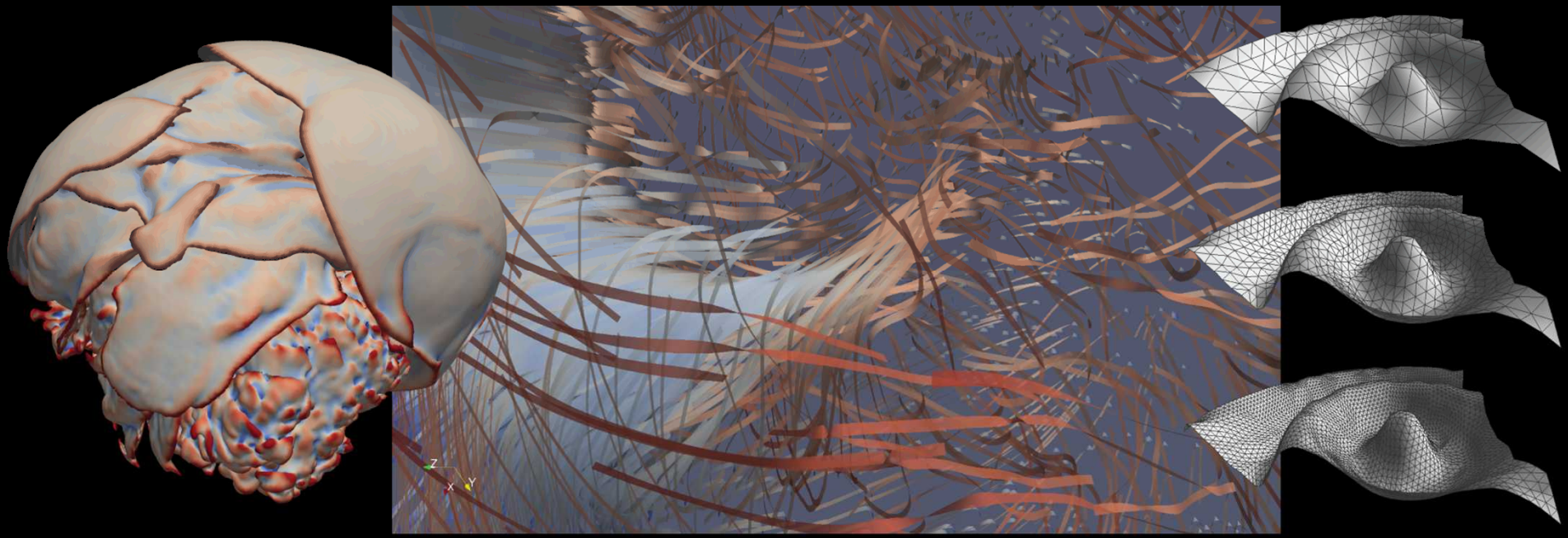
Sandia National Laboratories



Dax

Sandia National Laboratories

- Create a toolkit that well suited to design of visualization operations with a great number of shared memory threads.
- Develop a framework that adapts to emerging processor and compiler technologies.
- Design multi-purpose algorithms that can be applied to a variety of visualization operations.



```
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]...
}
```

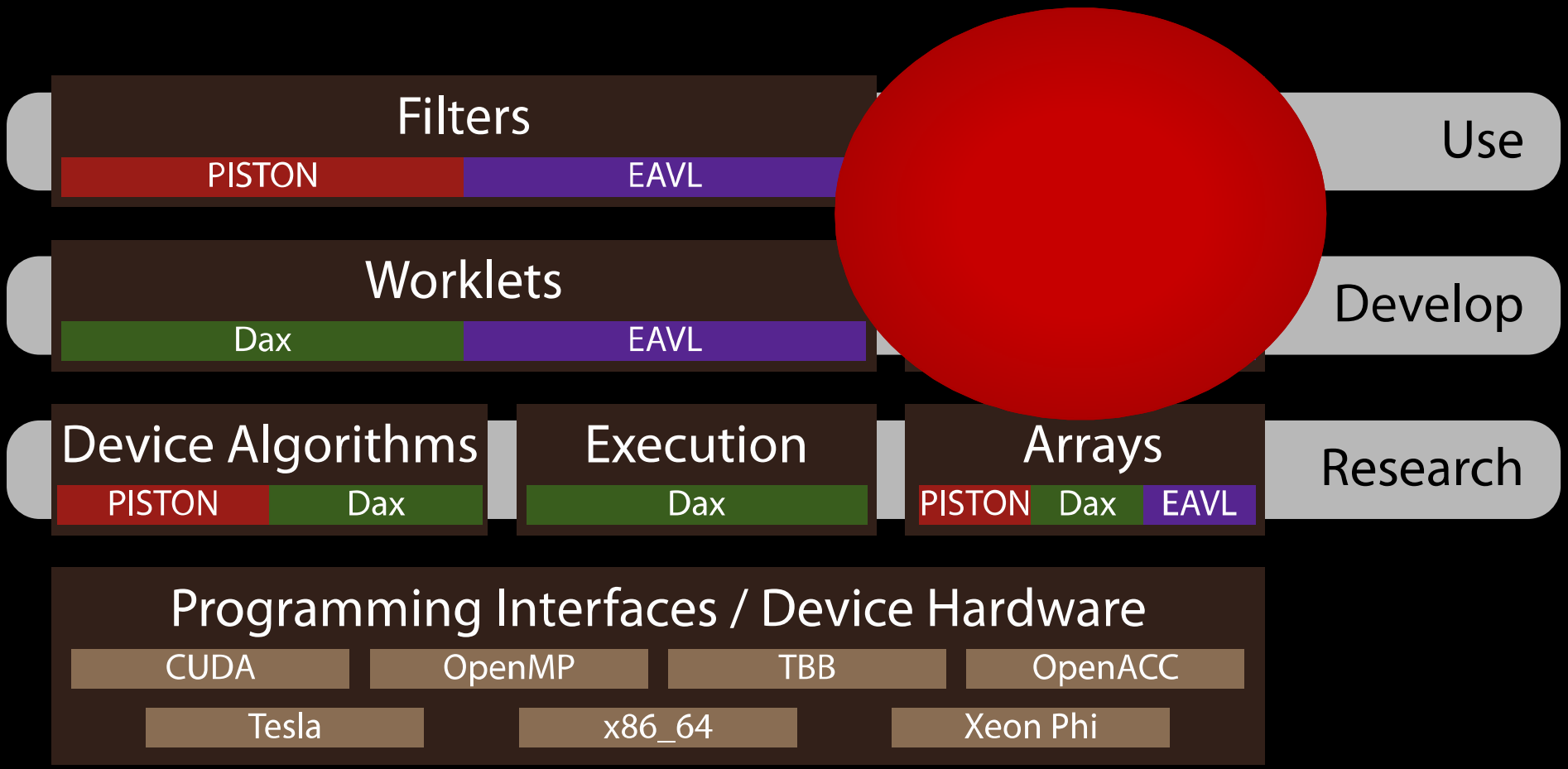
VTK Code

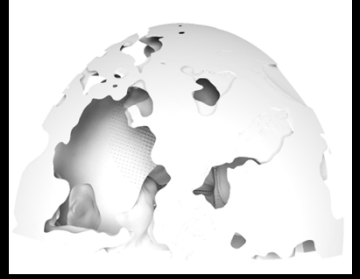
```
struct CellGradient : public WorkletMapCell
{
  typedef void ControlSignature(
      Topology, Field(Point),
      Field(Point), Field(Out));
  typedef _4 ExecutionSignature(_1,_2,_3);

  template<typename CellTag> DAX_EXEC_EXPORT
  Vector3 operator()(...)
  {
    Vector3 parametricCellCenter =
        ParametricCoordinates<CellTag>::Center();

    return CellDerivative(parametricCellCenter,
                          coords,
                          pointField,
                          cellTag);
  }

};
```

Dax Code

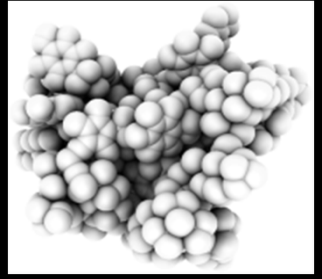
VTK-m Architecture

Sandia National Laboratories





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



EAVL enables advanced visualization and analysis for the next generation of 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

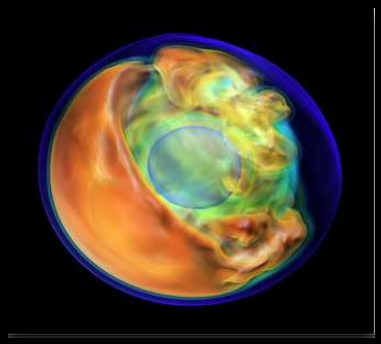
- Direct zero-copy mapping of data from simulation to analysis codes
- Heterogeneous processing models

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.

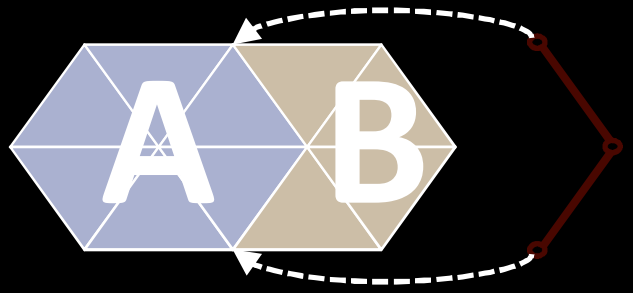
http://ft.ornl.gov/eavl

Data Models Supported by EAVL

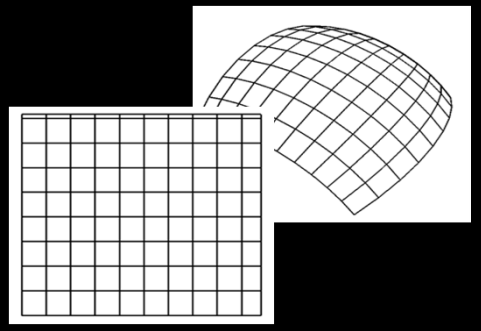
Sandia National Laboratories



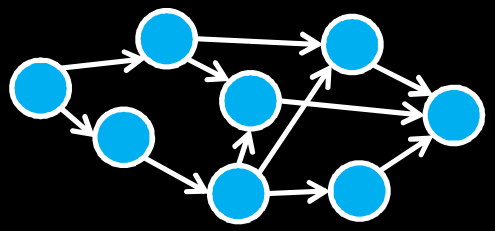
Low/high dimensional data
(9D mesh in GenASiS)



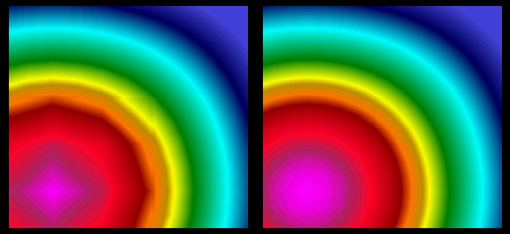
Multiple cell groups in one mesh
(E.g. subsets, face sets, flux surfaces)



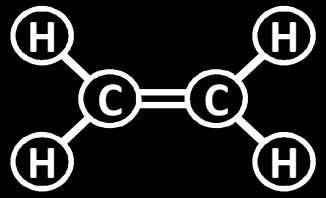
Multiple simultaneous
coordinate systems
(lat/lon + Cartesian xyz)



Non-physical data (graph,
sensor, performance data)



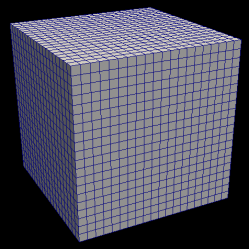
Novel and hybrid mesh types
(quadtree grid from MADNESS)



Mixed topology meshes
(atoms + bonds, sidesets)

VTK-m Framework

Sandia National Laboratories



Control Environment

Grid Topology
Array Handle
Invoke

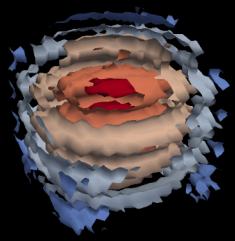
Device Adapter

Allocate
Transfer
Schedule
Sort
…

Execution Environment

Cell Operations
Field Operations
Basic Math
Make Cells

Worklet



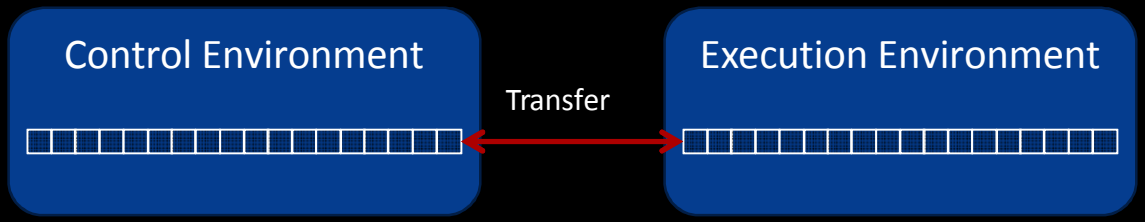
vtkm::cont

vtkm::exec

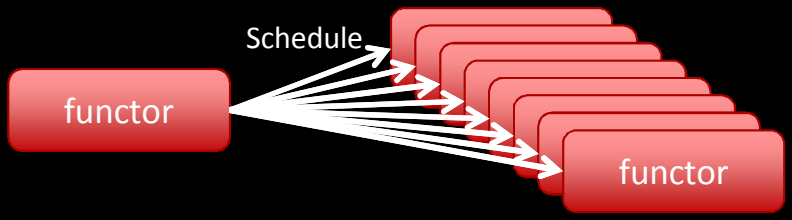
Device Adapter Contents

Sandia National Laboratories

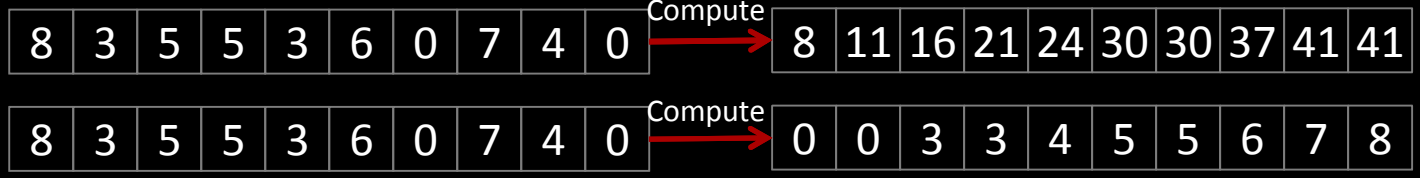
- Tag (`struct DeviceAdapterFoo {  };`)
- Execution Array Manager



- Schedule



- Scan
- Sort

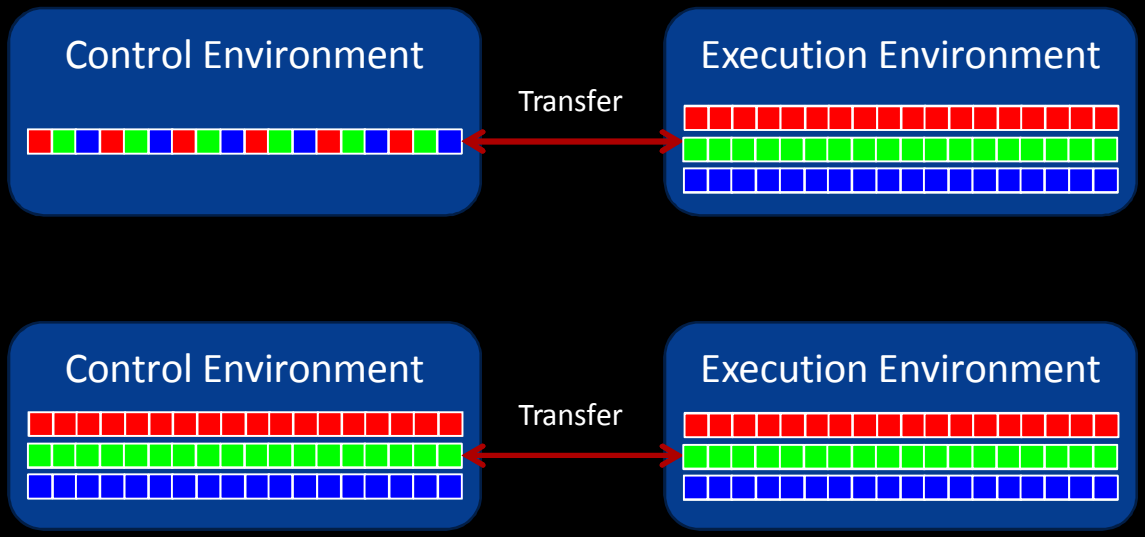


- Other Support algorithms
  - Stream compact, copy, parallel find, unique

VTK-m Arbitrary Composition

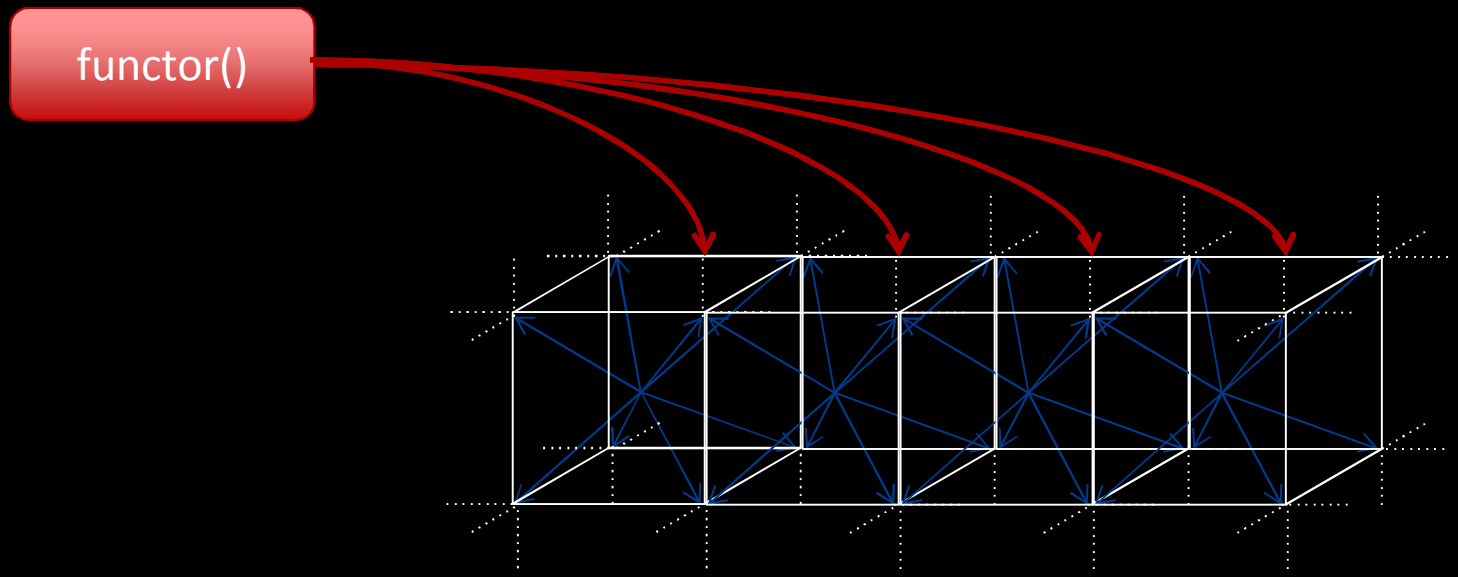
Sandia National Laboratories

- VTK-m allows clients to access different memory layouts through the Array Handle and Dynamic Array Handle.
  - –Allows for efficient in-situ integration
  - –Allows for reduced data transfer



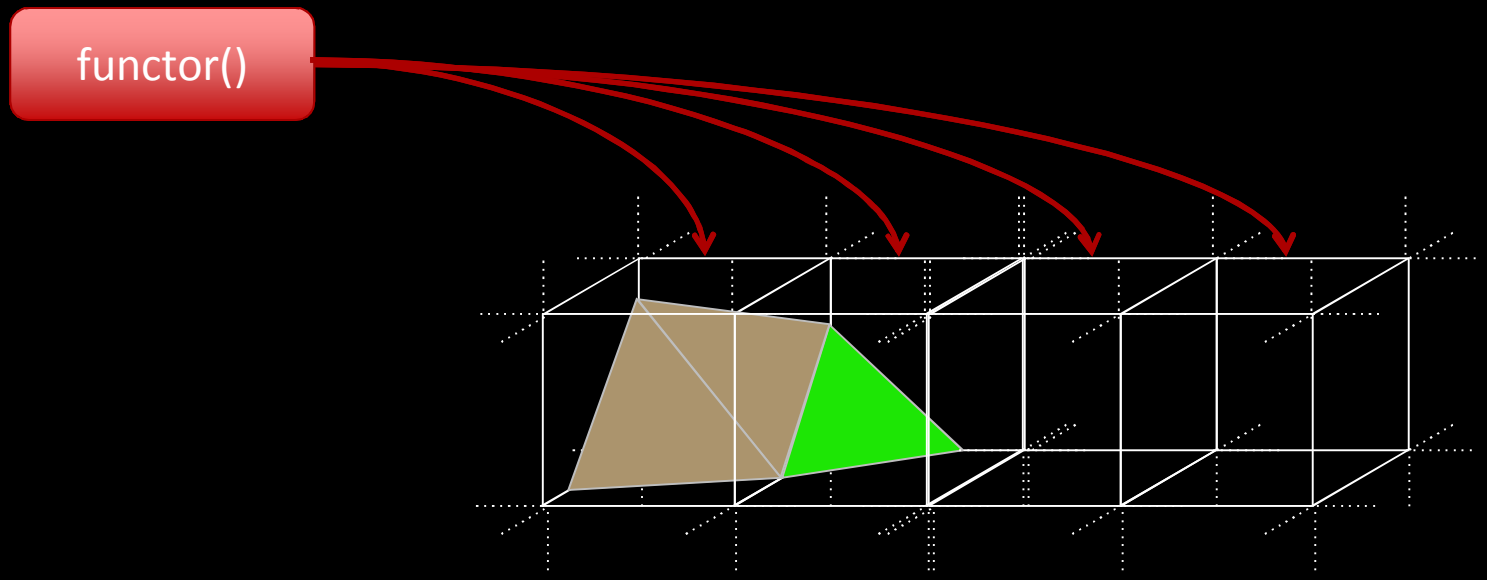
Functor Mapping Applied to Topologies

Sandia National Laboratories

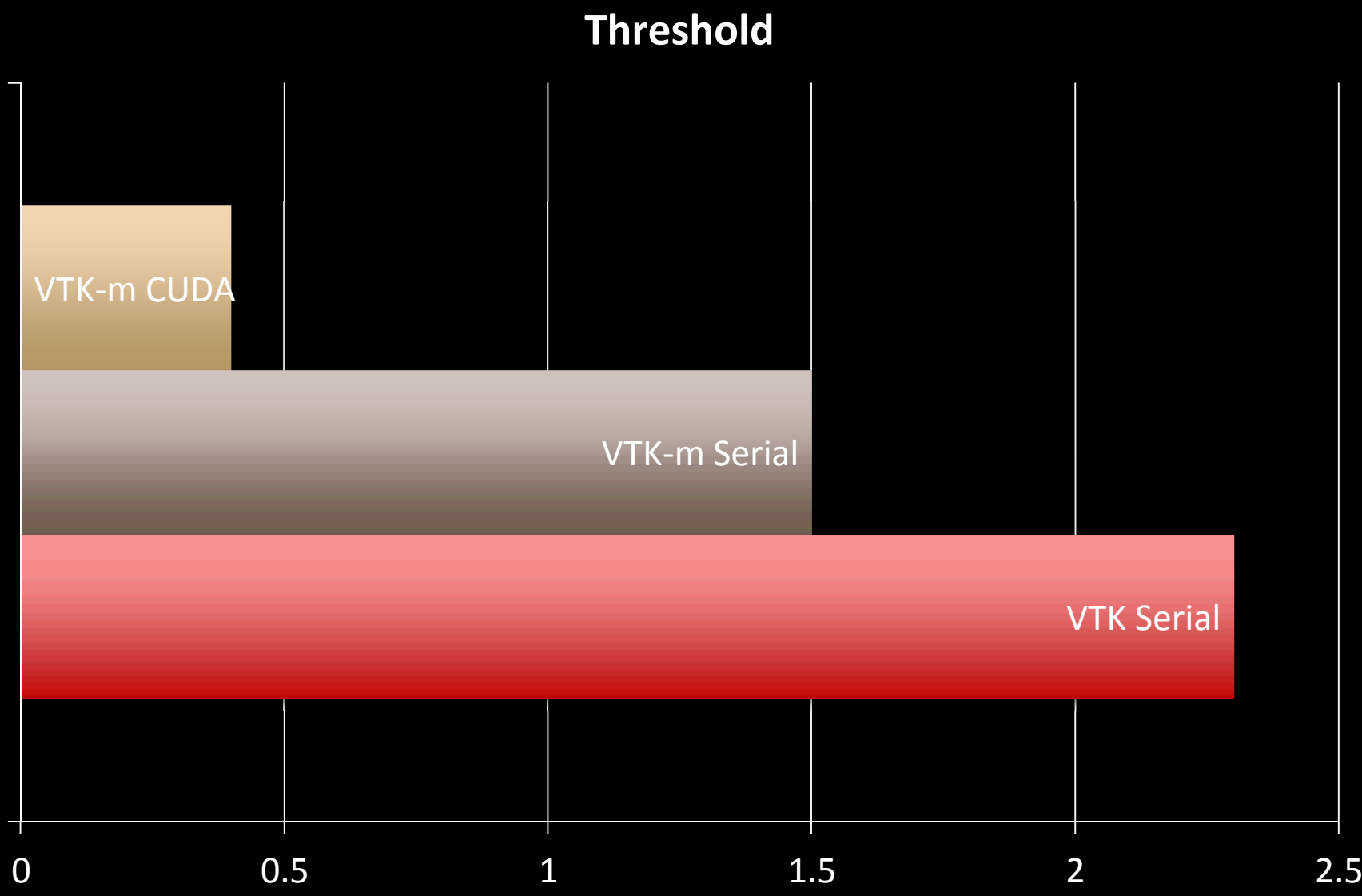


Functor Mapping Applied to Topologies

Sandia National Laboratories

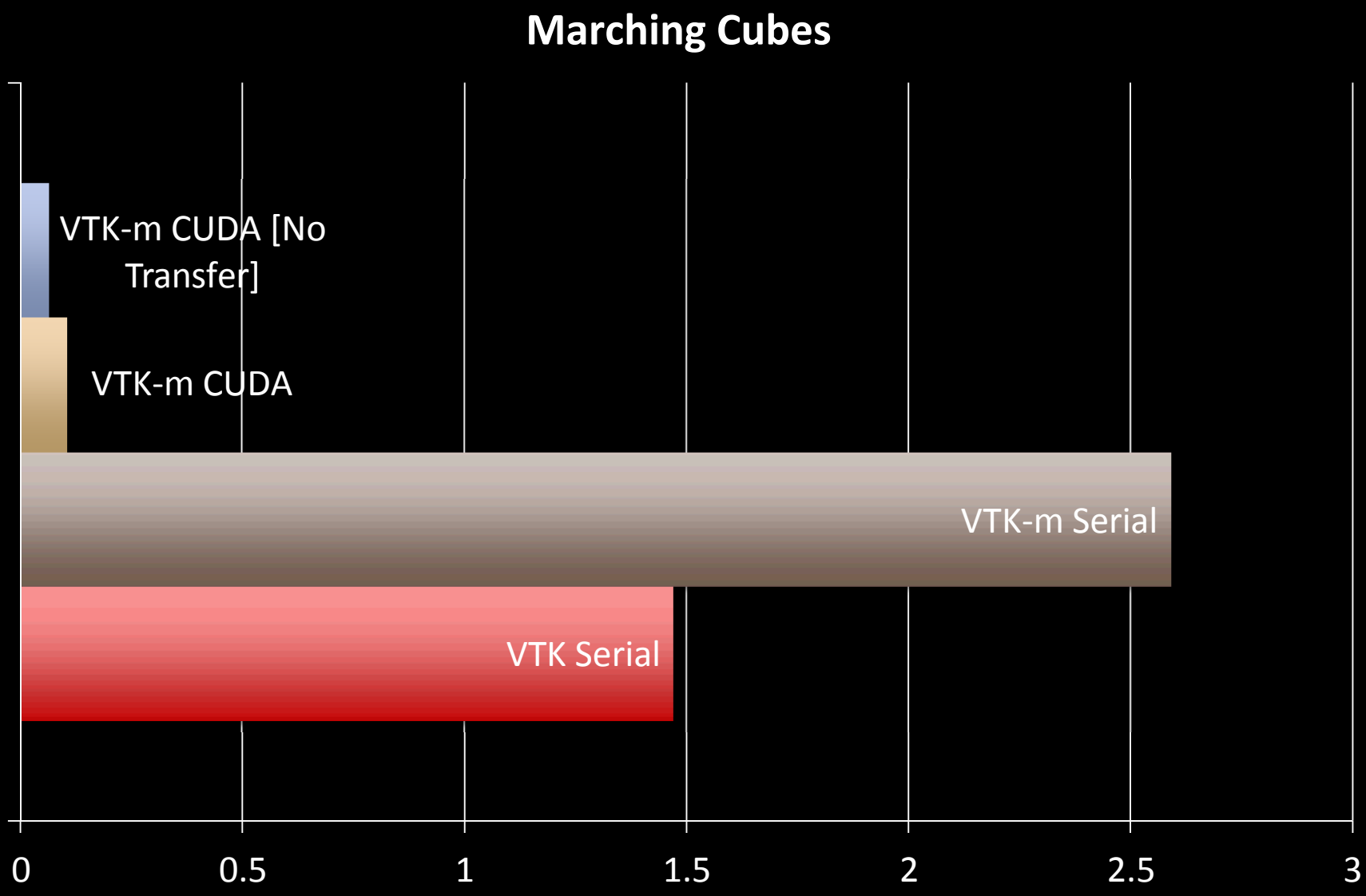


2 x Intel Xeon CPU E5-2620 v3 @ 2.40GHz + NVIDIA Tesla K40c



2 x Intel Xeon CPU E5-2620 v3 @ 2.40GHz + NVIDIA Tesla K40c

Data: 432^3



What We Have So Far

Sandia National Laboratories

- Features
  - Core Types
  - Statically Typed Arrays
  - Dynamically Typed Arrays
  - Device Interface (Serial, CUDA)
  - Basic Worklet and Dispatcher
- Compiles with
  - gcc (4.8+), clang, msvc (2010+), icc, pgi, nvcc
- User Guide work in progress
- Ready for larger collaboration

What We Have So Far

Sandia National Laboratories

- Compiles with
  - gcc (4.8+), clang, msvc (2010+), icc, and pgi

- User Guide work in progress
- Ready for larger collaboration

Questions?

Sandia National Laboratories

m.vtk.org