SAND2012-3234P

Next-Generation Codes/Portability

Dax Perspective

DOECGF 2012

April 25, 2012

Kenneth Moreland Sandia National Laboratories

Release Marking (e.g. Not Approved for Release, SAND XXXX, etc.)

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.

What fundamental problem are you trying to solve?

Exascale requires fine threaded (billion-way parallel) and minimized RAM movement.

Parallel programming is frickin' hard.

Developer 1: The VTK weak pointer implementation is not thread safe because it can be used just like a regular pointer, even though the object itself might be deleted by a separate thread at any time. [Fix involving locking proposed.]

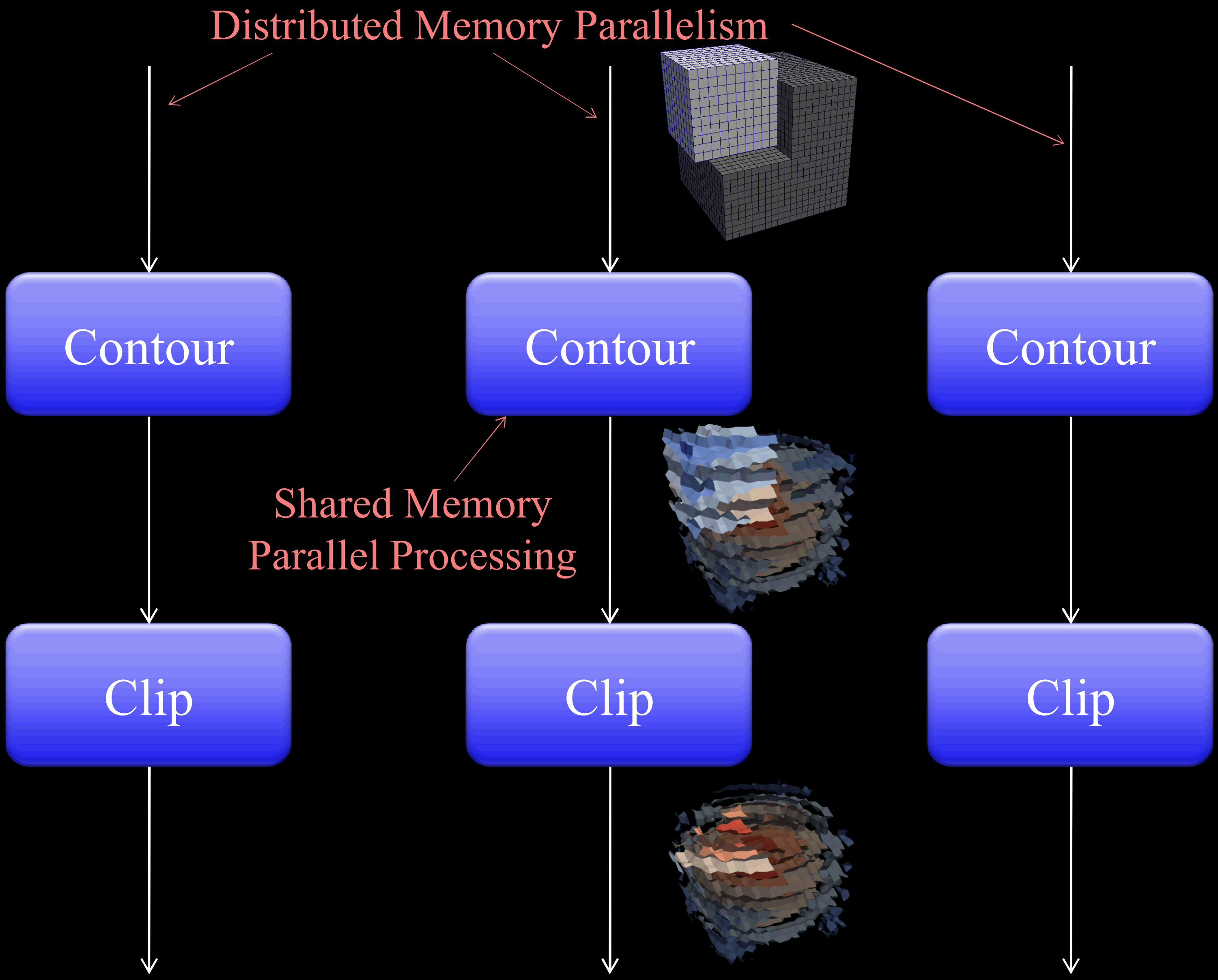
Developer 2: That sounds like a good idea. You have my vote.

Developer 3: How would this impact performance? Would it improve it do you think?

Developer 4: I'm skeptical that this code actually provides thread safety. Your critical section is only applied within a weak pointer. It is not applied within vtkObjectBase itself. While the critical section is running some other thread could call unreference and delete the object after the weak pointer checks the reference but before the smart pointer increments the reference.

Developer 1: I see your point. Indeed, applying a lock just to the WeakPointer doesn't really do anything.

Hybrid Parallel Pipeline



Distributed Memory Parallelism

Contour

Contour

Contour

Shared Memory
Parallel Processing

Clip

Clip

Clip

What are your plans to deal with… Massive Concurrency?

Build a framework to make the implementation easy.

What are your plans to deal with… Distributed Memory?

Do what we are currently doing.

What are your plans to deal with…
Memory Overhead?

Uhh…

Adaptable templated array iterators.

Incremental algorithms?

What are your plans to deal with… Fault Tolerance?

Nothing directly…

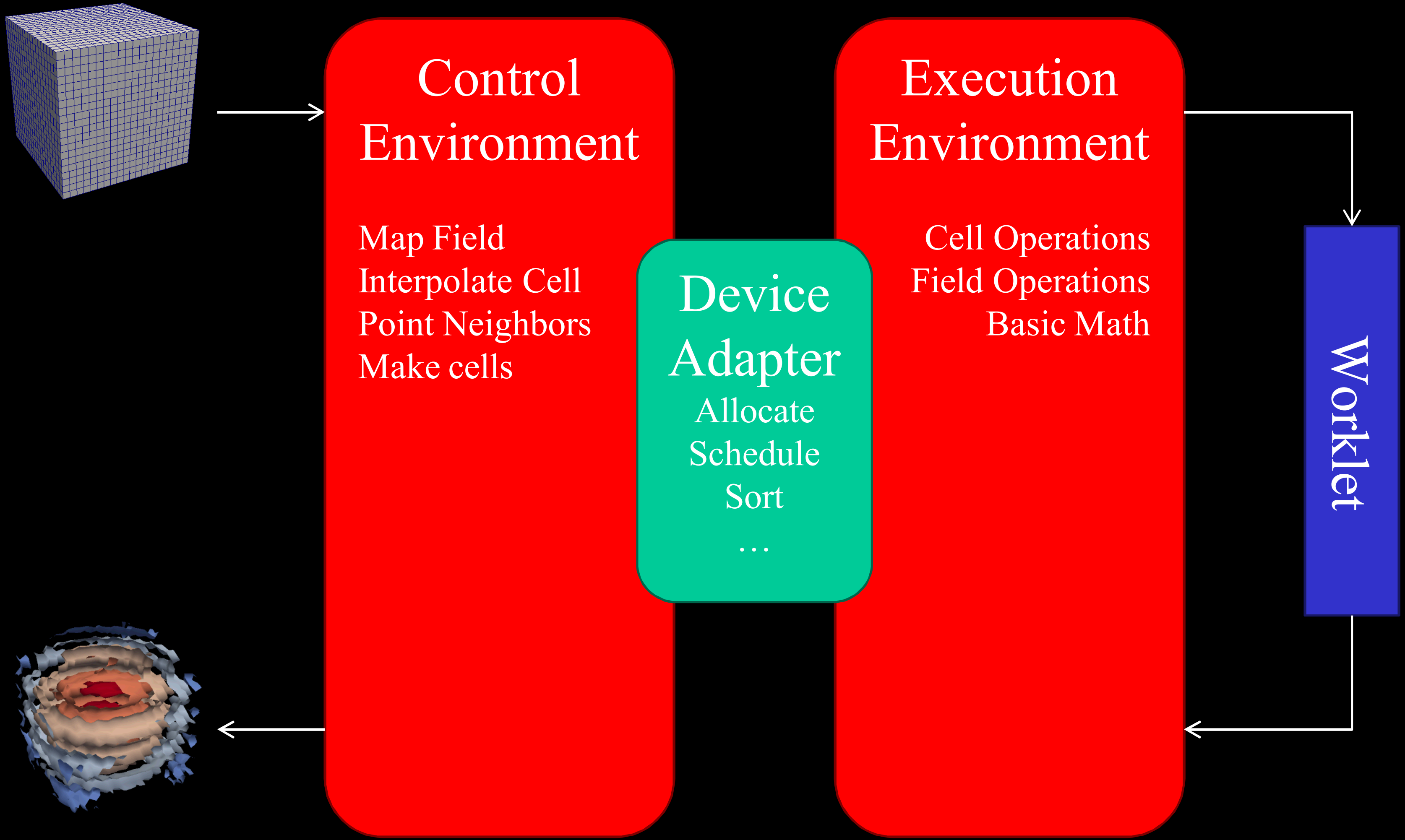
…but could potentially leverage programming models with fault tolerance.

Sandia
National
Laboratories

What is your philosophy for dealing with ambiguity of the exascale architecture.

The *right* thing to do is isolate and minimize the parallel-specific code.

Also the logical place to isolate the porting.



Control Environment

Map Field
Interpolate Cell
Point Neighbors
Make cells

Device Adapter

Allocate
Schedule
Sort
…

Execution Environment

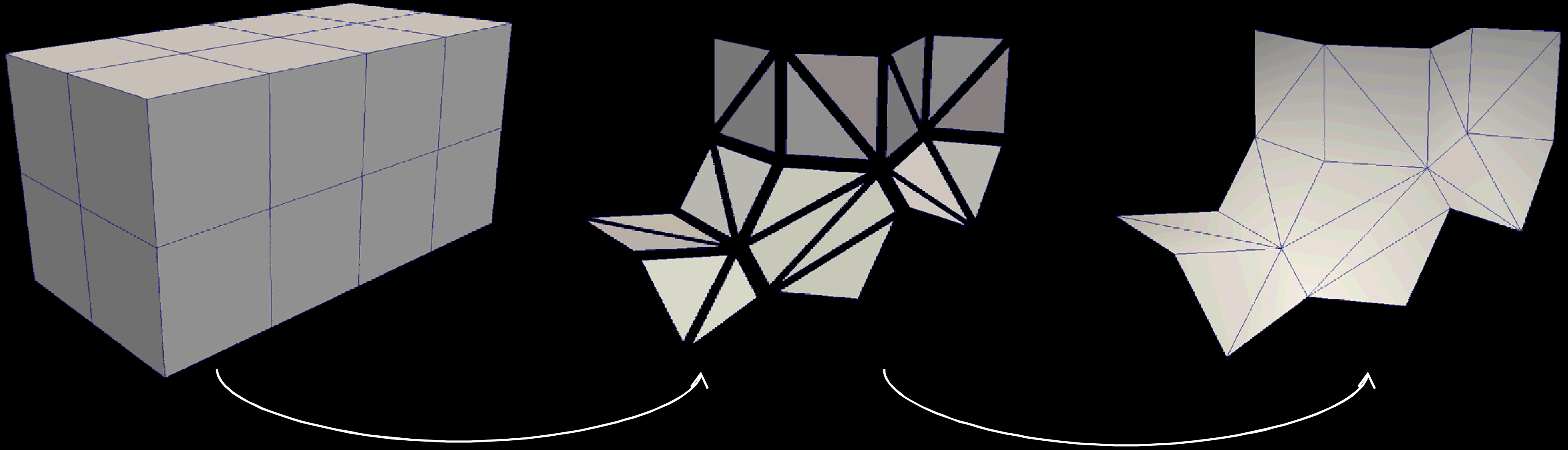
Cell Operations
Field Operations
Basic Math

Worklet

How is your technology implemented?

Worklets make it easy to design local operations.

Building blocks of communicative operations complete the algorithms.



Generate Triangles (Local)

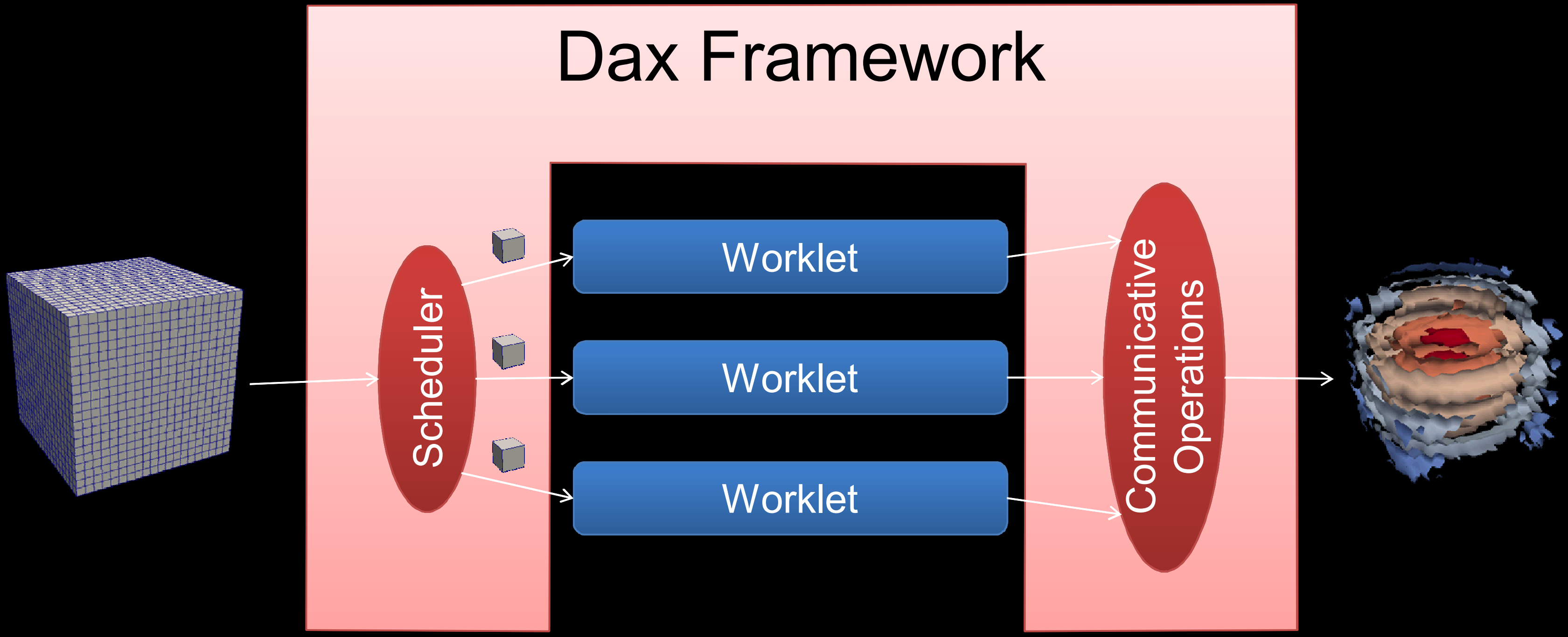
Connect Surface (Communicative)

Templated to adapt to different cell structures.

Familiar vis structres (vectors, points, cells, coordinates).

Hazard-free data access.

```
template<class CellType>
DAX_WORKLET void CellGradient(...)
{
  dax::Vector3 parametricCellCenter = dax::make_Vector3(0.5, 0.5, 0.5);
  CellType cell = work.GetCell();
  dax::Vector3 value = dax::exec::cellDerivative(work,
                                                 cell,
                                                 parametricCellCenter,
                                                 points,
                                                 point_attribute);
  work.SetFieldValue(cell_attribute, value);
}
```



Dax Framework

Scheduler

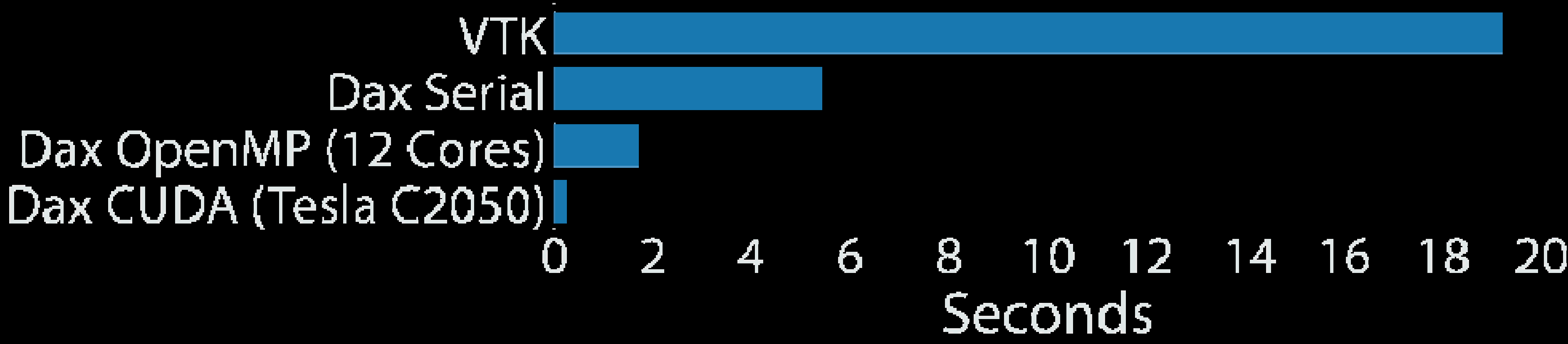
Worklet

Worklet

Worklet

Communicative Operations

Threshold Operation with Unused Point Removal



What is the long-term result for this effort?

Production software.