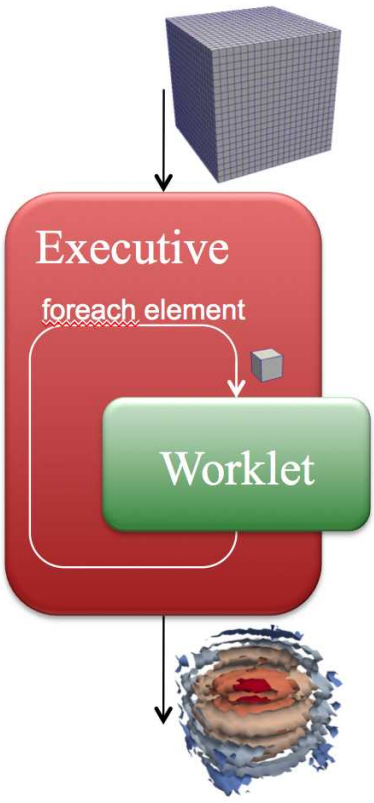
SAND2011-6964P

Data Analysis at Extreme: The Dax Toolkit

The transition to exascale machines represents a fundamental change in computing architecture. Efficient computation on exascale machines requires a massive amount of concurrent threads, at least 1000× more concurrency than existing systems. Current visualization solutions cannot support this extreme level of concurrency. Exascale systems require a new programming model and a fundamental change in how we design fundamental algorithms. To address these issues, our project, titled "A Pervasive Parallel Processing Framework for Data Visualization and Analysis at Extreme Scale," builds the Data Analysis at Extreme (Dax) Toolkit.

A Toolkit for Exascale Data Analysis

The Dax Toolkit supports the fine-grained concurrency for data analysis and visualization algorithms required to drive exascale computing. The basic computational unit of the Dax Toolkit is a worklet, a function that implements the algorithm's behavior on an element of a mesh (that is a point, edge, face, or cell) or a small local neighborhood. The worklet is constrained to be serial and stateless; it can access only the element passed to and from the invocation. With this constraint, the serial worklet function can be concurrently scheduled on an unlimited number of threads without fear of memory clashes or other race conditions.



The Dax Toolkit provides a unit called an executive that accepts a mesh, iterates over all elements in the mesh, invokes one or more worklet on each element, and collects the resulting values for each element. Conceptually we can think of this iteration as a serial operation, but of course in practice the executive will schedule the operation on multiple threads.

Sandia National Laboratories · Kitware · UC DAVIS UNIVERSITY OF CALIFORNIA · NNSA National Nuclear Security Administration

```
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

```
DAX_WORKLET void CellGradient(...)
{


  dax::exec::Cell cell(work);
  dax::Vector3 parametric_cell_center
    = dax::make_Vector3(0.5, 0.5, 0.5);


  dax::Vector3 value = cell.Derivative(
    parametric_cell_center,
    points,
    point_attribute,
    0);
  cell_attribute.Set(work, value);


}
```

Dax Code

Simplified Parallel Programming

The Dax Toolkit simplifies the development of parallel visualization algorithms. Consider the code samples above that come from the Visualization Toolkit (VTK) on the left and our Dax Toolkit on the right. Both implementations perform the same operation; they estimate gradients using finite differences. Both toolkits provide similar classes and functions, and consequently the code looks remarkably similar.

However, because the Dax Toolkit is structured such that it can schedule its execution on a GPU, we measure that it performs this operation over 100 times faster than the VTK code running on a single CPU. Furthermore, although the Dax code above is compiled and run using CUDA, all CUDA-specific constructs all encapsulated with the Dax Toolkit, simplifying the programmer's job and easing transitions to new programming models.

For more information and updates, please visit http://daxtoolkit.org.

Contacts

Kenneth Moreland, PI
Sandia National Laboratories
kmorel@sandia.gov

Kwan-Liu Ma, Co-PI
University of California at Davis
ma@cs.ucdavis.edu

Berk Geveci, Co-PI
Kitware, Inc.
berk.geveci@kitware.com

Utkarsh Ayachit, Technical Lead
Kitware, Inc.
utkarsh.ayachit@kitware.com

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.

Sandia National Laboratories · Kitware · UC DAVIS UNIVERSITY OF CALIFORNIA · NNSA National Nuclear Security Administration