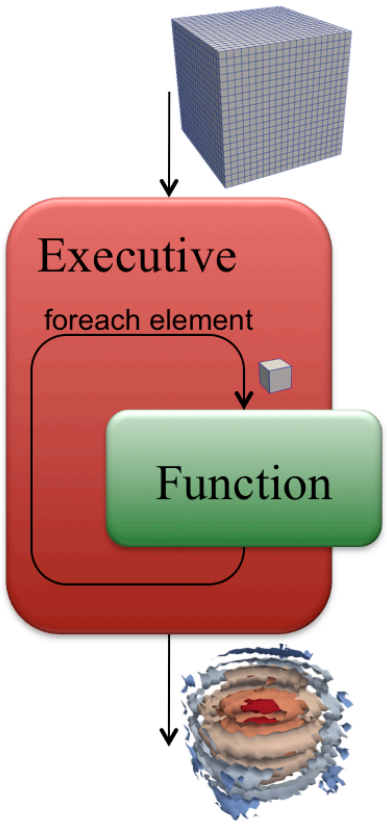
Data Analysis at Extreme: The Dax Toolkit

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 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 function is constrained to be serial and stateless; it can access only the element passed to and from the invocation. With this constraint, the serial 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 of these stateless functions 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.

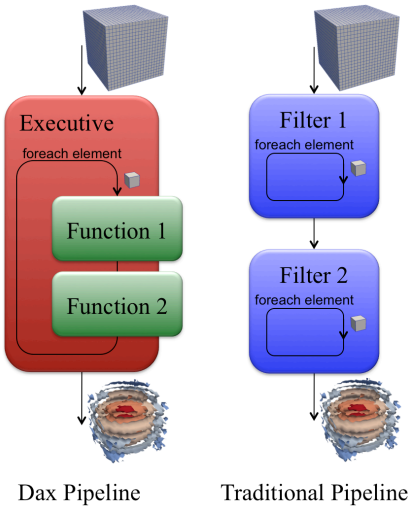
Sandia National Laboratories · Kitware · UC DAVIS UNIVERSITY OF CALIFORNIA · NNSA National Nuclear Security Administration

Features

When designing the Dax Toolkit, algorithms are expressed as serial functions. The complexity of the design, implementation, and debugging of parallel algorithms is removed in the Dax Toolkit. Thus, a developer becomes more efficient with the Dax Toolkit by focusing on the details of the algorithm rather than the intricacies of the parallel system.

With the Dax Toolkit, a single implementation of an algorithm can be adapted to multiple execution environments by applying different executives to the same function. For example, different implementations of an engine can schedule a function in a serial loop (for debugging purposes), on multiple CPU cores, or a GPU-type accelerator.

An executive of the Dax Toolkit has the potential of chaining multiple functions together within a single iteration of the data. Consequently, the executive can pull a single element of the mesh from DRAM and complete an entire chain of operations before ever having to write resulting data back to DRAM. Such execution behavior maximizes the instruction-to-memory fetch ratio, a vital metric when hundreds of instructions can be performed in the time of a single memory fetch. In comparison, each filter in a traditional visualization pipeline must independently iterate over an entire data set and store its results in a DRAM memory bank until the next filter reloads the data again.



For more information and further updates on development, 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.