Beyond Traditional Microprocessors for Geoscience HPC Applications

This is a summary of a paper published in IEEE Micro, vol. 31, no. 2, in March/April 2011.

About the Authors

Olav Lindtjorn is a visiting scholar at Stanford University. Haohuan Fu is an associate professor in the Center of Earth System Science in Tsinghua University. Bob Clapp is with the Center for Computational Earth & Environmental Science (CEES) at Stanford University. Oliver Pell is VP of Engineering, Oskar Mencer is CEO and Michael J. Flynn is Chairman at Maxeler Technologies and Professor Emeritus of Electrical Engineering at Stanford University.

Summary

High Performance Computing (HPC) plays a very important role in current applications of gas and oil extraction. Yet, as the demands of algorithms continue to increase, the viability of conventional HPC becomes questionable. Dataflow Engine (DFE) Technology is able to meet these increasing demands because it is capable of circumventing performance scaling issues experienced by conventional CPUs.

Fluid flow and Imaging are some of the most computationally demanding geoscience applications. The fluid flow problem is typically solved using a finite-element or finite-volume approach. This approach has become increasingly memory bound due to the fact that sparse matrix multiplication has come to rely on large caches. The imaging problem is approached through sparse matrix multiplication, fast Fourier transforms, or convolutions. Current techniques have focused on convolving with a small filter in the time domain or multiplying in the Fourier domain. Various tests have confirmed convolution the most effective method due to better defined parallelism and higher regularity in computational and memory access, facilitating more scalable performance. Reverse Time Migration (RTM) requires considerable sampling over the desired terrain, and the amount of data needed becomes very costly. Sampling requirements are significantly reduced using higher accuracy approximations to space and time derivatives.

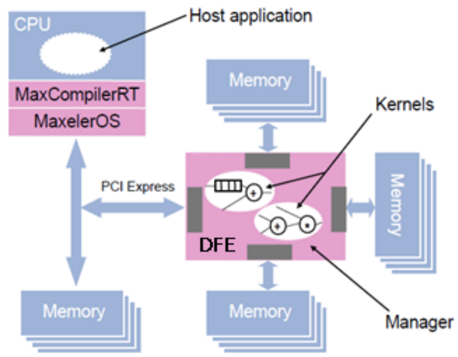

Figure 1: Maxeler accelerator architecture equipped with a host application, kernels and a manager.

The computational model for acceleration of application algorithms involves implementing these algorithms on one or more dataflow machines. This model consists of a host processing node, and an accelerator which uses a kernel to handle the mathematical computations. The kernel has potential for substantial acceleration. The kernel's code is simplified and flattened, objects are expanded and dynamic loops are converted to static loops. The altered kernel is compiled into a dataflow graph and the dataflow graph is compiled into a synchronous dataflow machine.

Maxeler's Maxcompiler is utilized to build dataflow machines to address application challenges and ultimately create DFE accelerators. Maxeler hardware acceleration systems contain one or more DFEs with local memories connected to a host CPU and a Maxring interconnect joining multiple DFEs. Figure 1 displays the architecture of a Maxeler accelerator containing one DFE. The Maxeler accelerated system consists of a host application, kernels and a manager. Separation of computation and communication allows for kernels to be deeply pipelined, resulting in high performance.

Sparse-matrix solving is heavily dependent on the computational cost of matrix-vector multiplication. This problem is effectively addressed using the conjugate gradient method and the benefits of DFEs. DFEs provide a high level of flexibility when encoding data and efficient parallelization when coupled with a Maxring interconnect across multiple DFEs. Custom hardware allows representation of matrices and vector data using fewer bytes in memory. Parallelization of multiple DFEs allows the total memory bandwidth to increase with the number of DFEs used.