Geophysical Research Abstracts
Vol. 12, EGU2010-7462, 2010
EGU General Assembly 2010

Accelerating a 3D finite-difference wave propagation code by a factor of 50 and a spectral-element code by a factor of 25 using a cluster of GPU graphics cards

Dimitri Komatitsch (1), David Michéa (2), Gordon Erlebacher (3), and Dominik Göddeke (4)

(1) University of Pau, CNRS and INRIA, France (dimitri.komatitsch@univ-pau.fr), (2) Bureau de Recherches Géologiques et Minières, Orléans, France, (3) Department of Scientific Computing, Florida State University, Tallahassee, USA, (4) Institut für Angewandte Mathematik und Numerik, TU Dortmund, Germany

We first accelerate a three-dimensional finite-difference in the time domain (FDTD) wave propagation code by a factor of about 50 using Graphics Processing Unit (GPU) computing on a cheap NVIDIA graphics card with the CUDA programming language. We implement the code in CUDA in the case of the fully heterogeneous elastic wave equation. We also implement Convolution Perfectly Matched Layers (CPMLs) on the graphics card to efficiently absorb outgoing waves on the fictitious edges of the grid. We show that the code that runs on the graphics card gives the expected results by comparing our results to those obtained by running the same simulation on a classical processor core. The methodology that we present can be used for Maxwell's equations as well because their form is similar to that of the seismic wave equation written in velocity vector and stress tensor.

We then implement a high-order finite-element (spectral-element) application, which performs the numerical simulation of seismic wave propagation resulting for instance from earthquakes at the scale of a continent or from active seismic acquisition experiments in the oil industry, on a cluster of NVIDIA Tesla graphics cards using the CUDA programming language and non blocking message passing based on MPI. We compare it to the implementation in C language and MPI on a classical cluster of CPU nodes. We use mesh coloring to efficiently handle summation operations over degrees of freedom on an unstructured mesh, and we exchange information between nodes using non blocking MPI messages. Using non-blocking communications allows us to overlap the communications across the network and the data transfer between the GPU card and the CPU node on which it is installed with calculations on that GPU card. We perform a number of numerical tests to validate the single-precision CUDA and MPI implementation and assess its accuracy. We then analyze performance measurements and in average we obtain a speedup of 20x to 25x.