FDTD Based Seismic Modeling and Reverse Time Migration on a GPU Cluster

Rached Abdelkhalek, Henri Calandra, Olivier Coulaud, Guillaume Latu, Jean Roman

FDTD Based Seismic Modeling and Reverse Time Migration on a GPU Cluster

R. Abdelkhalek[1*] , H. Calandra[1] , O. Coulaud[2] , G. Latu[3] , J. Roman[2]

[1] TOTAL, avenue Larribau, F-64000 Pau
[2] INRIA/Scalapplix project, 341 cours Libération, F-33405 Talence Cedex
[3] Strasbourg University & INRIA/Calvi project, LSIIT, bd Seb. Brant, F-67400 Illkirch
[*] Email: rached.abdelkhalek@total.com

Abstract

We have designed a fast parallel simulator that solves the acoustic wave equation on a GPU cluster. Solving the acoustic wave equation in an oil exploration industrial context aims at speeding up seismic modeling and Reverse Time Migration. We use a finite difference approach on a regular mesh, in both 2D and 3D cases. The acoustic wave equation is solved in either a constant density or a variable density domain. We use CUDA to take advantage of the GPUs computational power. We study different implementations and their impact on the application performance. We obtain a speedup of 11 for Reverse Time Migration and up to 30 for the modeling application over a sequential code running on general purpose CPU.

Key words: Seismic modeling, Reverse Time Migration, Finite Difference, CUDA.

1 Background

1.1 Seismic Modeling

Numerical seismic modeling aims at simulating seismic wave propagation in a geological medium in order to generate synthetic seismograms that are the seismograms that a set of sensors would record, given an assumed structure of the subsurface. Among the numerous approaches to seismic modeling, direct methods based on approximating the geological model by a numerical mesh are of particular interest. Infact, this approch can give very accurate results. However, a disadvantage of this approach is its high computational demand [3].

1.2 Reverse Time Migration

Reverse Time Migration (RTM)[2] is a technique for creating seismic images in areas of complex wave propagation, providing imaging of so called turning and prismatic waves. Its main limitation is its high computational cost. RTM is based on the simulation of waves propagation. Both source and receivers wave fields are propagated respectively forward and backward in time. These wave fields are then cross-correlated for corresponding time steps in order to form the subsoil image.

1.3 Governing Equations

The three dimensional acoustic wave equation (1) links the pressure field *u(x,y,z,t)* to the density $\rho(x, y, z)$ and the velocity *c(x,y,z)*[4].

$$\frac{1}{c^2\rho}\frac{\partial^2 u}{\partial t^2} = \nabla.(\frac{1}{\rho}\nabla u) \tag{1}$$

Using finite difference (FD) methods to solve the wave equation is one way among others to tackle direct methods. The way this equation is derived among a regularly meshed domain is described in [5]: we write a cascaded first order spatial difference expression to compute the second time difference of the wave field:

$$\frac{\partial^2 u}{\partial t^2} = K[\frac{\partial_-}{\partial x}(\frac{1}{\rho}\frac{\partial_+ u}{\partial x})+\frac{\partial_-}{\partial z}(\frac{1}{\rho}\frac{\partial_+ u}{\partial y})+\frac{\partial_-}{\partial z}(\frac{1}{\rho}\frac{\partial_+ u}{\partial z})] \tag{2}$$

where $K = c^2\rho$ is the bulk modulus. The ∂_- and ∂_+ symbols denote the spatial difference operators that are centered halfway between grid points either forward or backward in the direction of the spatial difference.

When the density is considered to be constant in all the domain, equation (1) is simplified to equation (3).

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = \Delta u \tag{3}$$

This approximation is especially done during migration process. The discretization of this equation is done with a second order scheme in time and a eighth-order central differences in space.

1.4 GPU cluster testbed

Our GPU cluster testbed is composed of 10 Xeon bi-socket quadcore nodes (2.0 GHz clock frequency, 16 GB of RAM), coupled with 5 Nvidia Tesla S1070

servers. Each node is connected via one PCIe gen2 bus to the Tesla server. The Tesla server is composed of 4 T10 GPUs. Each pair is connected via a switch to a PCIe 2.0 connection. Thus each node has access to 2 GPUs via the same bus.

T10 GPUs can be seen as a set of 30 *multiprocessors*. Each multiprocessor comprises 8 *streaming processors* running in a single instruction multiple data (SIMD) like way. These processors can execute 3 single precision floating point operations per clock cycle. Our T10 GPUs have a clock rate of 1.44 GHz, providing thus a theoretical peak performance of $30 \times 8 \times 3 \times 1.44 = 1$ TFlops per GPU and 4 TFlops per S1070 blade. To exploit the GPU computing power and make it available for non graphics programmers, NVIDIA introduced the Compute Unified Device Architecture (CUDA)[1]. CUDA defines a novel GPU architecture and a programming model based on a Single Program Multiple Data (SPMD) paradigm: the programmer writes a portion of code (the *kernel*) to be executed by several *threads* in parallel on different data.

2 Performance and Numerical Results

Figure 2 shows the speedup and times obtained using our GPU implementation of the variable density modeling, for different subdomain decompositions. Each subdomain is treated on a host node. Reported times represent the whole application cost, including host-GPU communications. Speedup decreases with increasing number of subdomains because the computing time per subdomain decreases and the communication time becomes more predominant. Yet, only computing times are reduced when using GPUs. We report in figure 2 times and speedup for the Reverse Time Migration. The RTM implementation

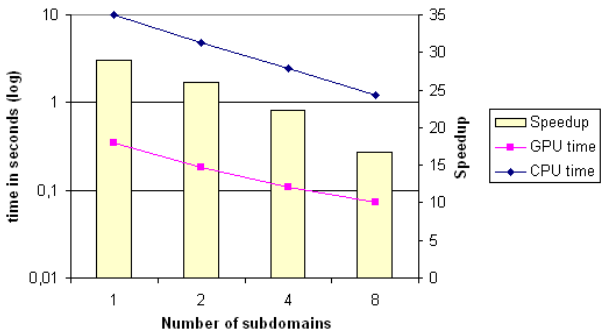

Figure 1: Variable Density Modeling averaged times in seconds for one iteration on a $521 \times 254 \times 1067$ test case.

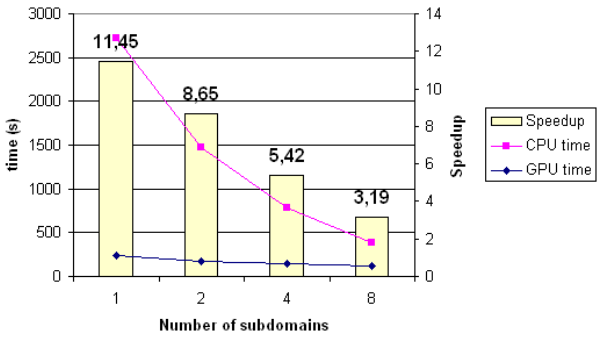

Figure 2: Reverse Time Migration times in seconds for on a $288 \times 118 \times 338$ test case.

involves more host-GPU communications than the modeling. This reduces the obtained speedup.

The difference between synthetic seismograms produced using the CPU and the GPU implementations for a realistic data set used in production process is proportional to the wave field amplitude but remains very low (0.1% in terms of percentage error). This validates our implementation.

References

[1] Jose M. Carcione, Gerard C. Herman, and A. P. E. ten Kroode. Seismic modeling. *Geophysics*, 67(4):1304–1325, 2002.

[2] Biondo L. Biondi. *3D Seismic Imaging*. SEG, 2006.

[3] Jon F. Claerbout. *Imaging the Earth's Interior*. Blackwell Scientific Publications, 1985.

[4] John T. Etgen and Michael J. O'Brien. Computational methods for large-scale 3d acoustic finite-difference modeling: A tutorial. *Geophysics*, 72(5):SM223–SM230, 2007.

[5] *NVIDIA CUDA Compute Unified Device Architecture - Programming Guide*, 2008.