Anisotropic Reverse-Time Migration Using Co-Processors

MAXELER
Technologies

About the Authors

Wei Liu, Tamas Nemeth, Joseph Stefani and Ray Ergas are Geophysicists at Chevron. Alexander Loddoch, Ling Zhuo and Bill Volz are with Technical Computing at Chevron.
Oliver Pell is VP of Engineering and James Huggett is an Acceleration Architect at Maxeler Technologies.

Summary

Co-processors offer attractive acceleration opportunities to waveform-based imaging and inversion applications in challenging exploration and production environments. Unlike seismic forward modeling, the large amount of data involved in seismic imaging and inversion can pose a significant challenge to scalable acceleration in both GPUs and dataflow engines (DFEs).

For adjoint state problems such as reverse-time migration, the source and receiver wavefields propagate forward and backward respectively in time and at any timestep both wavefields need to be available simultaneously. The requirement for the backward-propagated wavefield poses an architectural challenge: as the algorithm progresses backwards over decrementing values of time-step index n, the wavefield must either be dynamically recalculated via forward propagation from 0 to n (the computational approach), or instead derived from a stored sequence of pre-calculated forward propagation time-steps (the storage-driven approach). Traditional CPU-based implementations typically avoid the computational approach
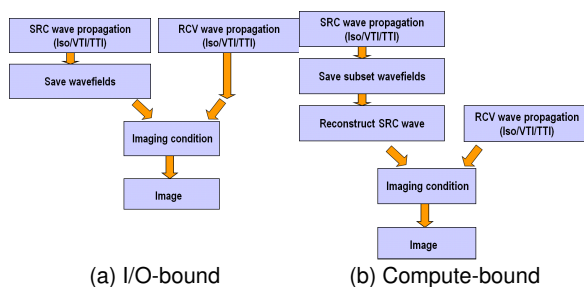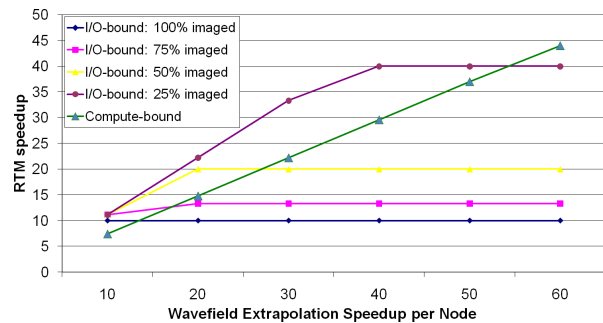
Figure 2: Relationship between RTM speedups (involving data access) and forward modeling (wavefield extrapolation only) speedups per node.

due to its associated $O(N^2)$ cost, and instead favor the storage-driven approach.

In this paper, we evaluate several RTM algorithms, and explore their limitations for GPU and DFE architectures as the level of DFE parallelization is increased. The large data volume required for the storage-based approach to anisotropic RTM causes it to become I/O-bound beyond a certain level of computational parallelization.

First, the modeling computation for a storage-based (I/O-bound) approach (*Figure 1a*) is offloaded onto Maxeler's DFE-based accelerator cards. The DFE implementation is optimized to maximize data throughput to and from the host, and minimize latency. We observe a 4x speedup overall, whereas our DFE implementation of forward modeling is capable of 20-50x speedup (vs. a 2.6GHz 8-core Intel Xeon). This figure will not be increased by further DFE parallelization, nor by simply optimizing the computation.

We develop an alternative, computational (compute-bound) approach (*Figure 1b*), trading reduced I/O volume for extra computation. *Figure 2* shows how RTM speedup relates to forward modeling speedup for the two RTM acceleration options in a DFE analysis.

For the storage-based approach, RTM performance can be limited by wavefield compression and imaging costs, so the speedup achievable varies depending on the proportion of the domain that is imaged. With the computational approach, however, RTM speedup is a linear function of the forward modeling speedup because most of the storage cost is transformed into computation cost by modifying the algorithm.

Figure 1: Workflows of storage-based (I/O-bound) and computational (compute-bound) approaches to RTM.