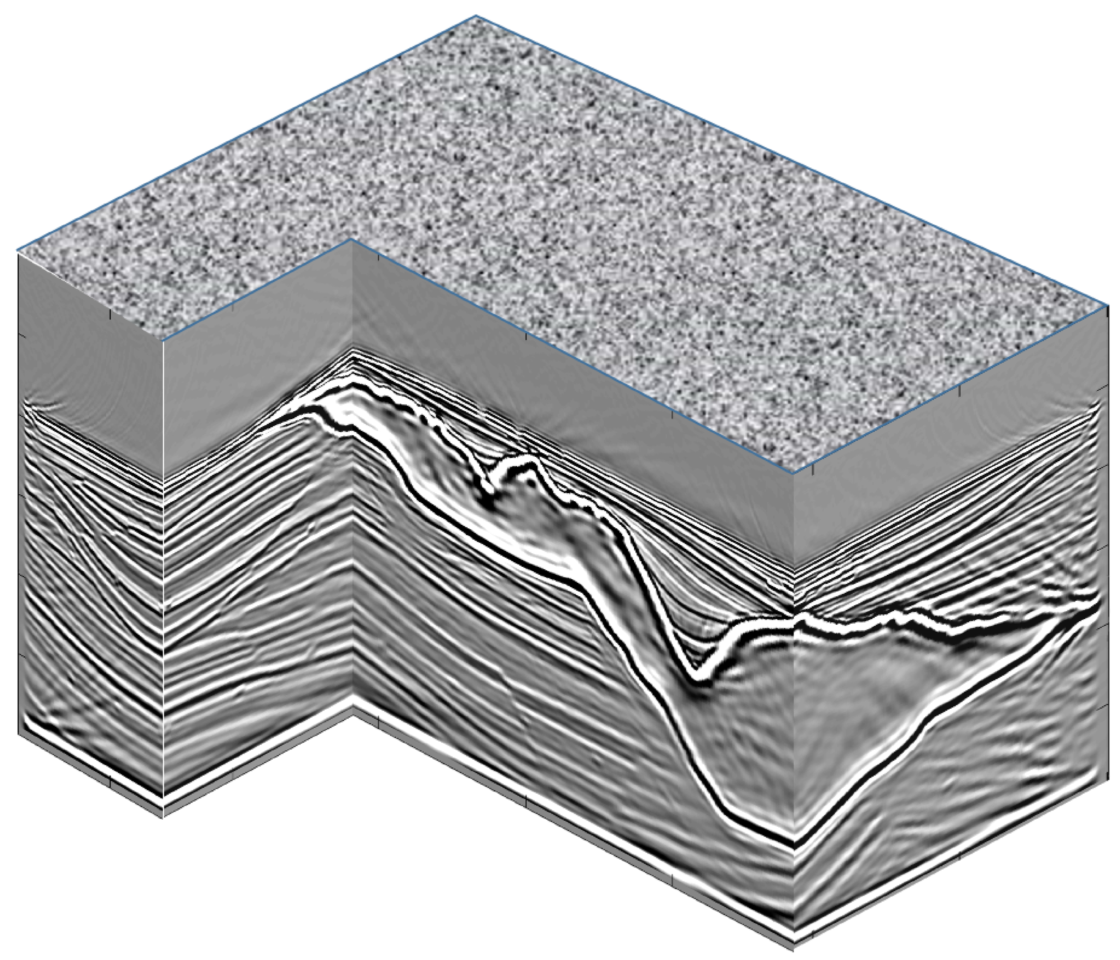
Enabling Performance-Portability of Full Wavefield Migration for Seismic Tomography

Andreas Hadjigeorgiou, The Cyprus Institute, CaSToRC

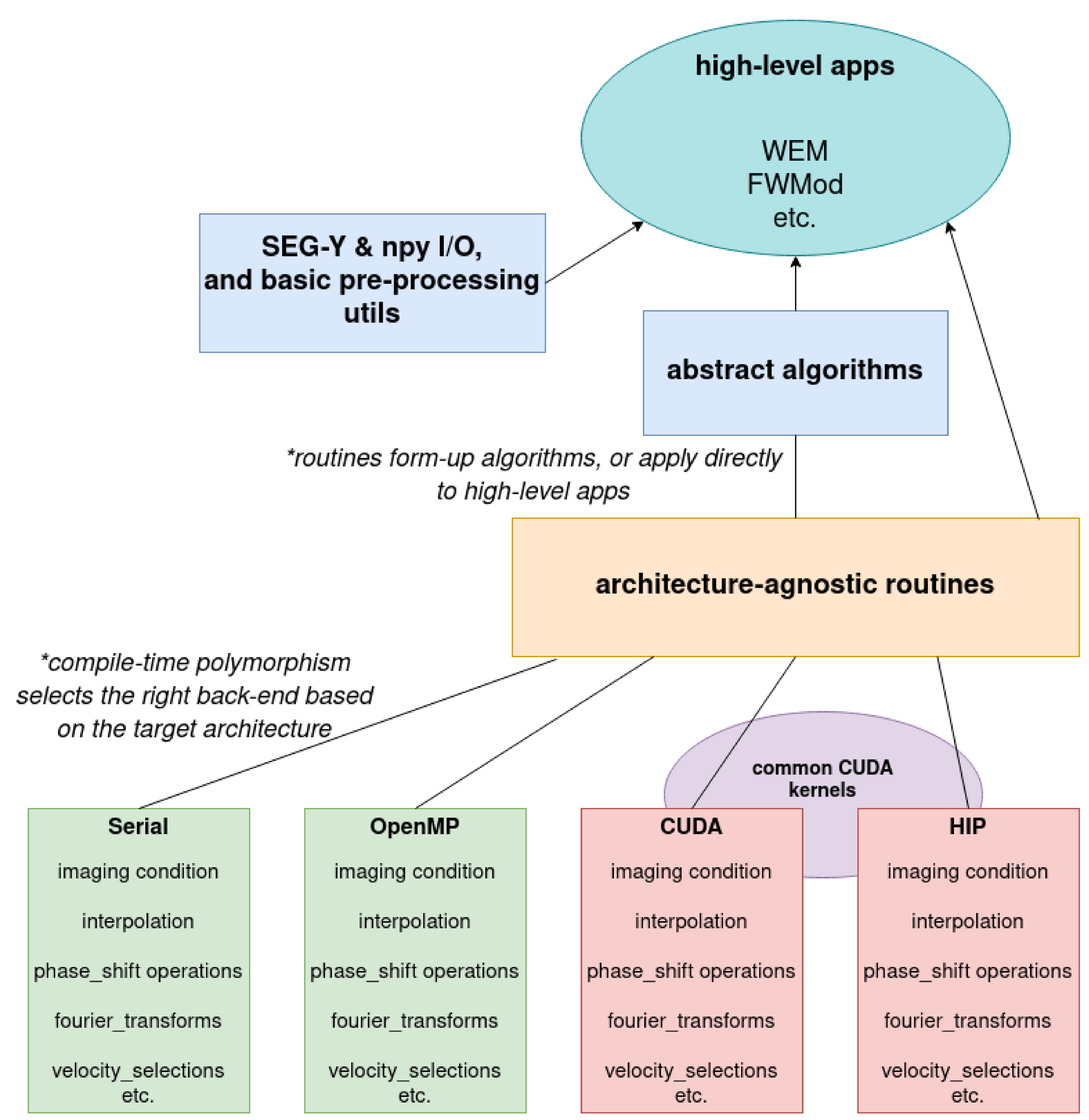
Introduction

Seismic migration is the process of geometrically locating the seismic events to the positions in space (or time) where they occurred. In principle, the migration process relies on modelling wavefields that propagate through the subsurface. Accurate migration requires the knowledge of the subsurface proporties, which is at the same time the subject matter; this makes it an inverse problem. Full Wavefield Migration (FWM) is an iterative migration scheme that optimizes subsurface delineation by employing in the imaging process *primary* as well as *higher-order* scattering effects. However, this comes at higher computational cost. In realistic seismic surveys the amount of data that needs to be processed is in the order of Petabytes. On top of this, the iterative process introduced by the FWM scheme makes the computational complexity of the process such high, that supercomputing level resources are needed in order to make applications practical.



The evolution in HPC from parallel processing on single-core to many-core processing units of different architectures such as CPUs, GPUs, FPGAs, and other emerging hardware, makes portability - without performance loss - a real challenge!

Code design



Portability-approach: Different platform-specific back-end implementations adhere to one abstract interface, enabling portability across a range of CPU/GPU computer architectures for the high-level applications that are built on top of.

Performance-approach: Use the right programming model for each target architecture.

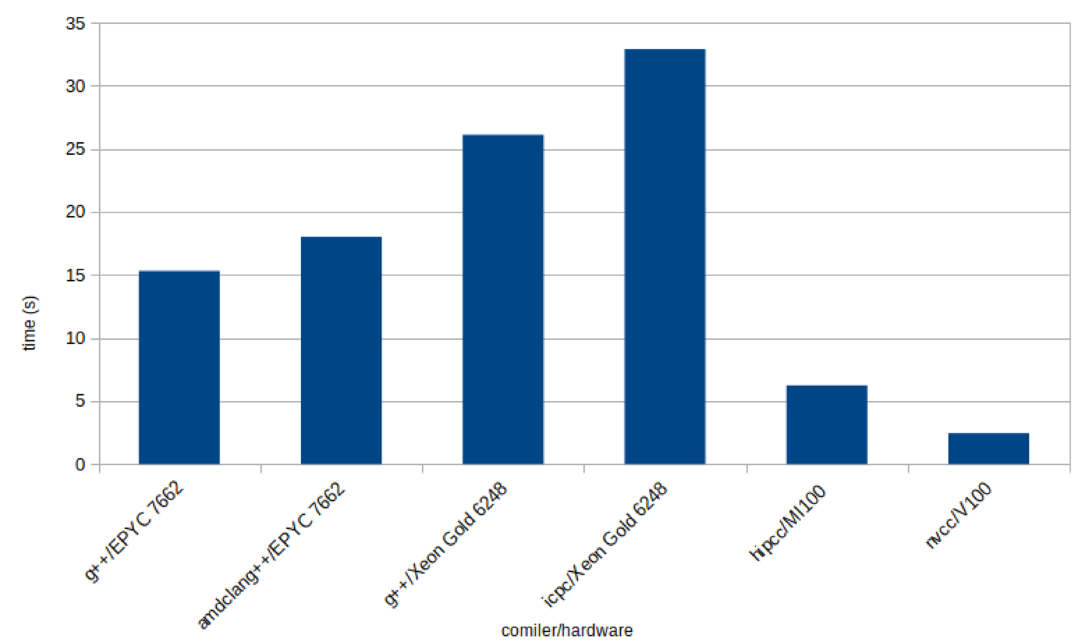
- Develop the *Serial* implementation for reference.
- Use OpenMP threads for CPU parallelism.
- Use CUDA for acceleration on Nvidia GPU.
- Use HIP layer to port CUDA on AMD GPUs.

Objectives

- Optimize the wavefield modelling engine, which is the most computationally complex operation in FWM, for CPU and GPU architectures using platform-specific implementations.
- Develop platform-specific routines that provide additional essential functionalities for FWM.
- Develop an abstraction layer in C++ that makes these platform-specific implementations accessesible through a common interface.
- Develop a set of containers (classes) and routines (funtions) that adhere to the C++ abstraction layer, to ease the development of algorithms.
- Using the C++ abstraction layer develop performance-portable FWM for seismic imaging.

Preliminary results

So far: Effectively ported the wavefield modelling engine [1] on Intel, AMD CPUs, and Nvidia, AMD GPUs. On top of the modelling engine we have developed one-iteration migration algorithms like Wave-Equation Migration [2], and currently working on more sophisticated iterative migration algorithms like FWM [3,4].



Next: MPI+X for distributed parallelism!

Remarks

- Using OpenMP and CUDA we enable portability of FWM to almost all available hardware in todays supercomputers.
- The portability to HIP was generally trivial, including the *hipFFT* library.
- To enable portability with this approach, that additional code-development effort was 30-40 %.
- Unit-testing applies to architecture-agnostic routines level (*see fig.*); it was not necessary to write unit-tests explicitly for each implementation.
-

References

[1] C. Bagaini et. al, *Data parallel implementation of 3D PSPI*, SEG Technical Program Exp. Abstracts, :188-191, 1995.

[2] B. Biondi *3D Seismic Imaging* SEG BOOKS, 2006

[3] A. J. Berkhout *Review Paper: An outlook on the future of seismic imaging, Part II: Full-Wavefield Migration*, Geophys. Prosp., 62(5):931–949, 2014.

[4] M. Davydenko and D. J. Verschuur *Full-Wavefield Migration: using surface and internal multiples in imaging*, Geophys. Prosp., 65(1):7–21, 2017.

Contact Information - Any feedback is welcome!!!

Personal page: https://ahadji05.github.io, **Email**: a.hadjigeorgiou@cyi.ac.cy, **Phone**: +357 99839383