Performance, Portability and Productivity for 3D Wave Models

Larisa Stoltzfus
PPar 2015 Intake
Let's break that down:
Performance (n.)
For computationally intensive algorithms (like many physical models), obtaining results in a timely manner is critical.
Portability \textit{(n.)}

With the plethora of HPC architectures available, programmers should be able to run their codes on newer, more performant platforms without rewriting and retuning.
In addition to performance and portability, it is critical to retain Performance Portability: high performance across different architectures.
Productivity ($n.$)
Computational scientists shouldn't need to know how to perform low-level optimisations to write HPC codes.
3D Wave Models

Use discretised wave equations (modelled as *stencils*) to simulate wave phenomena.

Can be difficult to model absorbing boundary conditions, which are not currently expressible in higher-level abstractions.
There are many frameworks currently available that focus on stencils.

- Halide
- Pochoir
- Exastencils
However, none of these provide a solution which provides all three of performance, portability and productivity for 3D wave models.
Instead: Use a Modular Approach

Computational scientists write their codes in an existing, easily programmable DSL

DSL writers implement their abstractions with a hardware-agnostic intermediary language

A compiler handles the low-level details of specialising for a particular hardware
The good news is that this goal is not far from realisation!
Lift aims to serve as an intermediary language between portable and productive DSLs and high performance code.
Proposed Workflow

computational scientist

productive, portable high-level abstractions

DSL

intermediary language

search optimisation space for best model version

rewrite rules

performant, portable, productive 3D wave model code
Current Work

Adapt *Lift* to abstract absorbing boundary conditions for 3D wave models like room acoustics and ground penetrating radar simulations.

Implement optimisations for 3D stencils in *Lift*.

Extend existing stencil DSL to compile 3D wave models into *Lift*.
Preliminary Results

![Graph showing performance comparison between two platforms (AMD R280 and NVIDIA GTX780) for different code versions. The bars represent time in milliseconds, with 'we are here' and 'and here' indicating specific points of interest.]
Conclusions

- There are currently no frameworks that provide performance, portability and productivity for 3D wave-based models.
- By using the *Lift* framework as an intermediary stage, current stencil based DSLs could be adopted to create performant, portable and productive 3D wave simulations.