A Modular Approach to Performance, Portability and Productivity for 3D Wave Models Using the Lift Framework

Larisa Stoltzfus
larisa.stoltzfus@ed.ac.uk
http://www.lift-lang.org

Supervisors
Christophe Dubach
Alan Gray
Stefan Bilbao

Motivation

- Parallel programming landscape is becoming more complex and less portable
- Computational scientists should not have to be experts in parallelisation
- New architectures require maintaining multiple specialised code bases
- Re-writing and re-tuning code is tedious

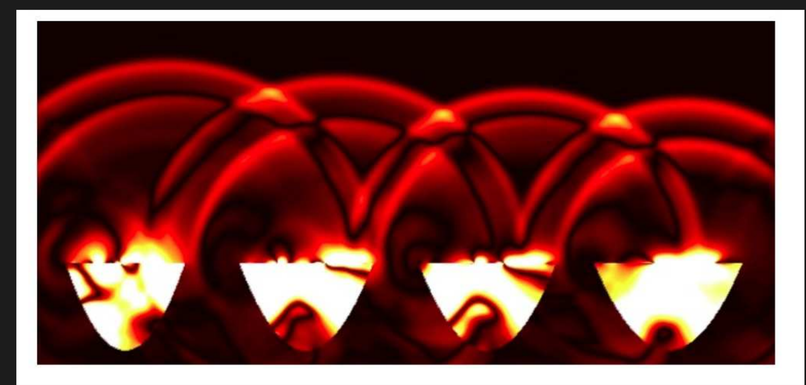
Vision

3D Wave Models

Finite **D**ifference **T**ime **D**omain is a common numerical method for modelling the 3D wave equation: space is discretised into a 3D grid of points and values are updated based on their neighbours
This algorithm is known as a *stencil*

$$\frac{\partial^2 \Psi}{\partial t^2} = c^2 \left(\frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} + \frac{\partial^2 \Psi}{\partial z^2} \right)$$

The 3D wave equation



Snapshot of room acoustic simulation with four timpani drums

The Lift Framework

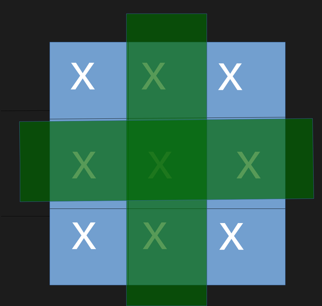
Lift is a parallel intermediary language between productive, portable high-level DSLs and low-level, high performance code

A small number of primitives can be combined to express many different types of applications

A search over a space of different code optimisations using *rewrite rules* produces specialised code for a particular platform

Selection of *Lift* primitives

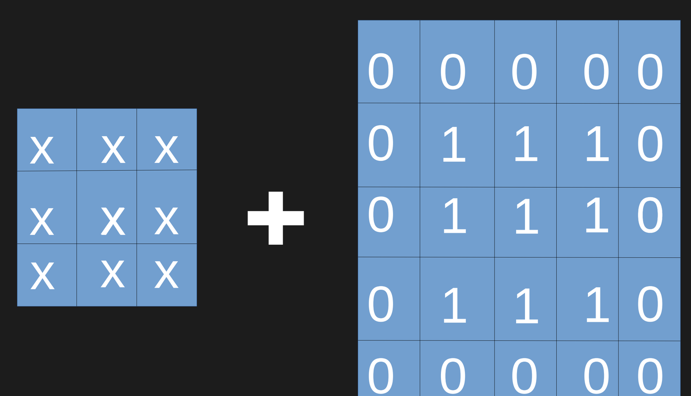
Optimising Stencils in the Lift Framework



Selection of stencil shape (*select*)

Targeting different groups of neighbouring values

```
LIFT:  map(reduce(nbh[-1]+nbh[0]+nbh[1] ∘ slide(size,step)) « input
C:  for(i = 0 → n)  updated[i] = data[i-1]+data[i]+data.at(1)
```



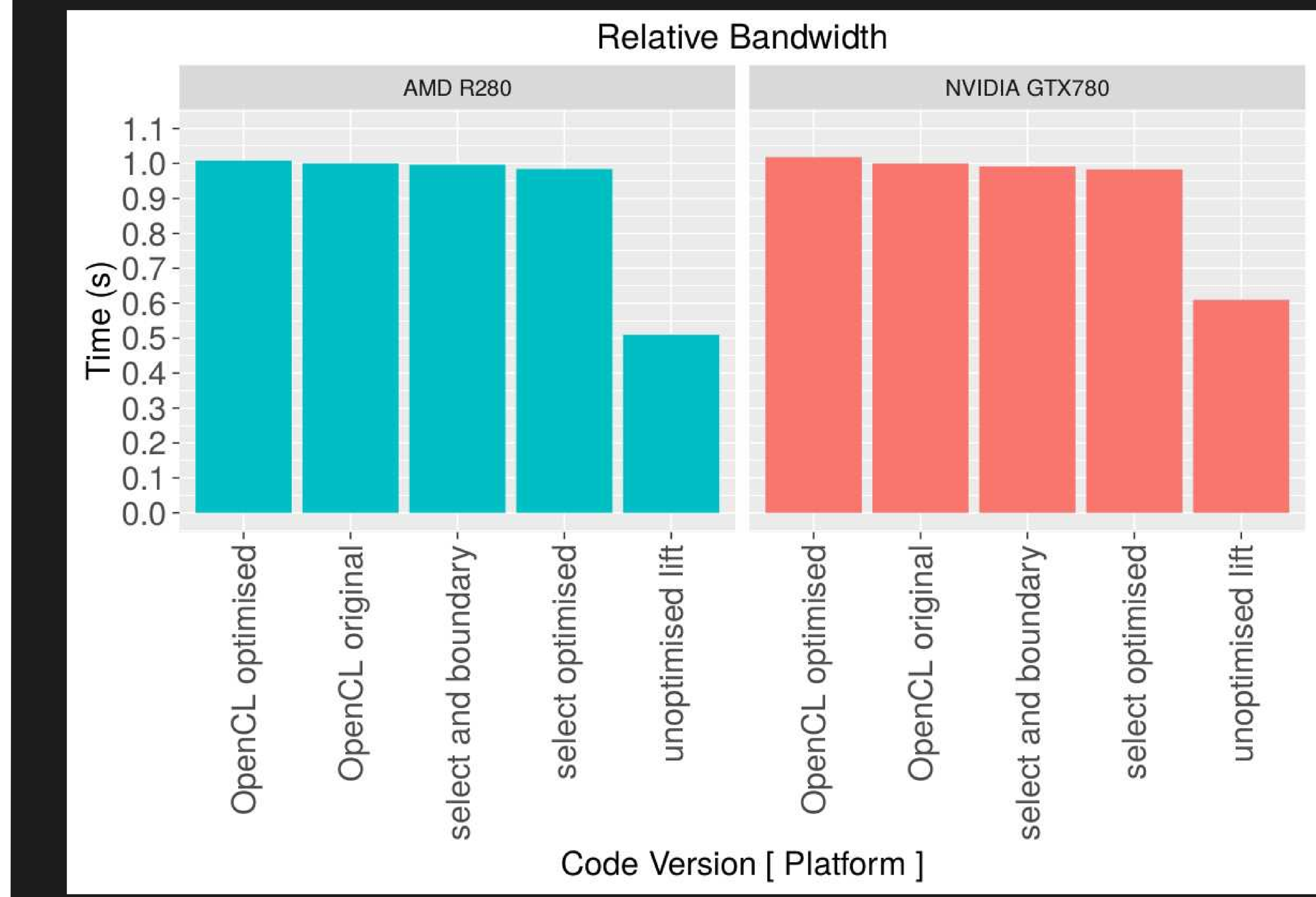
On-the-fly boundary handling (*boundary*)

Creating masking values when needed instead of storing in memory

```
LIFT:  boundaryValue = ArrayGen(idx)*border1 + !ArrayGen(idx)*border2
C:  boundaryValue = if(idx <= M || idx > 0 )  return 1.0  else  return 0.0
```

Results

- Developed simple room acoustics simulations in the *Lift* framework
- After iterations of optimisation, now close to the original benchmark



Future Work

- Investigate abstractions for absorbing boundary conditions
- Express 2.5D Tiling optimisation as a rewrite rule in *Lift*
- Determine and adapt stencil-focused DSL for 3D wave models
- Translate stencil-based DSL into *Lift*

THE UNIVERSITY *of* EDINBURGH informatics
epcc
EPSRC Engineering and Physical Sciences Research Council
EPSRC Centre for Doctoral Training in Pervasive Parallelism