

Justifying Reverse Time Migration Order of Accuracy on NVIDIA GPUs



Marcel Nauta
Software Developer, Acceleware

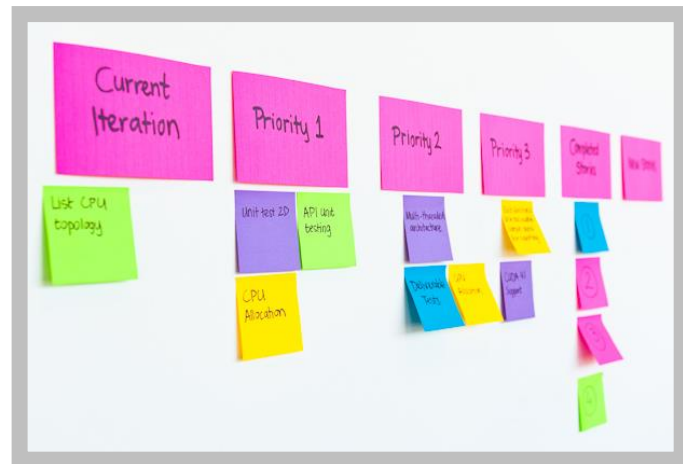
About Acceleware

- Accelerated software
 - Forward Modeling
 - TTI Reverse Time Migration
 - Full Waveform Inversion
- Software acceleration services
 - Feasibility studies
 - Algorithm parallelization and code optimization
 - Migration of applications to heterogeneous platforms
- Programmer training
 - CUDA, OpenCL, OpenMP, MPI



Agenda

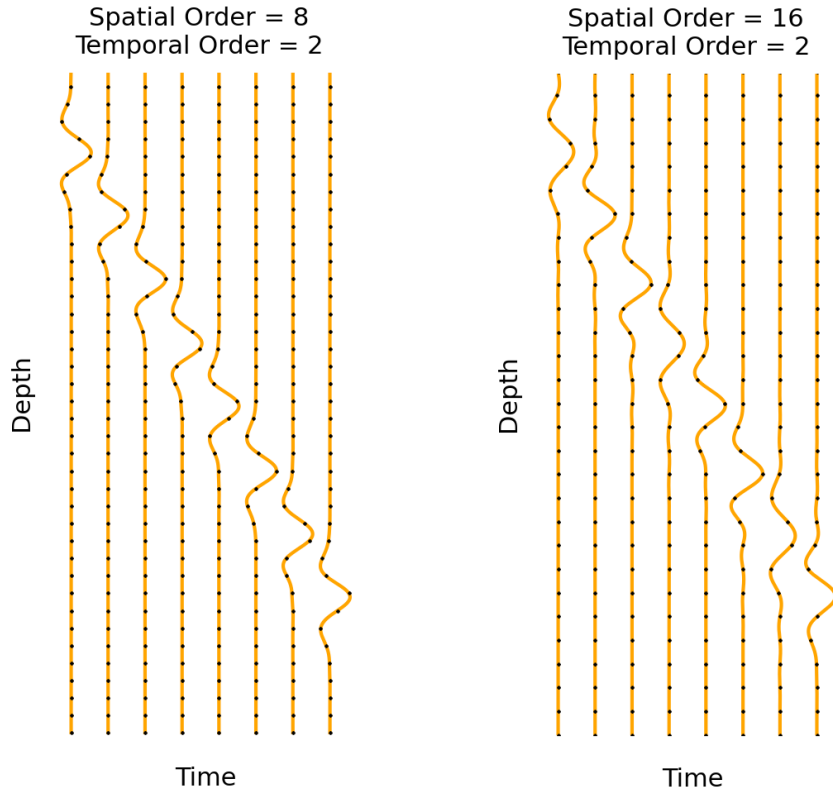
- Introduction to RTM
- RTM Grids and GPU Memory
- Implications of changing spatial and temporal order
- Discussion of trade-offs



Reverse Time Migration Algorithm

- Goal of RTM is to produce an image of the reflectors in the subsurface
- The most computationally expensive step is the 3D simulation of the wavefields

Effect of Spatial Order



- Sparser Grid
- Less Memory
- Less Points to update

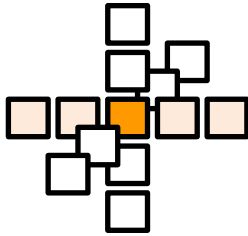
Spatial Order Limitations

- For RTM, the propagation must always be upsampled to the imaging grid
- High orders of accuracy are more prone to subtraction (floating point) errors
- Spatial orders compared using equal phase velocity error (spatial dispersion)

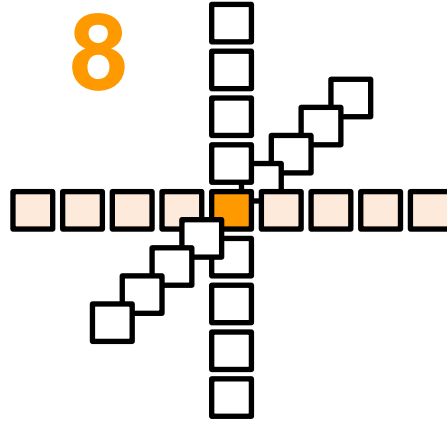
Spatial Order Algorithm

- Every timestep is similar to filtering a 3D volume with the stencil below

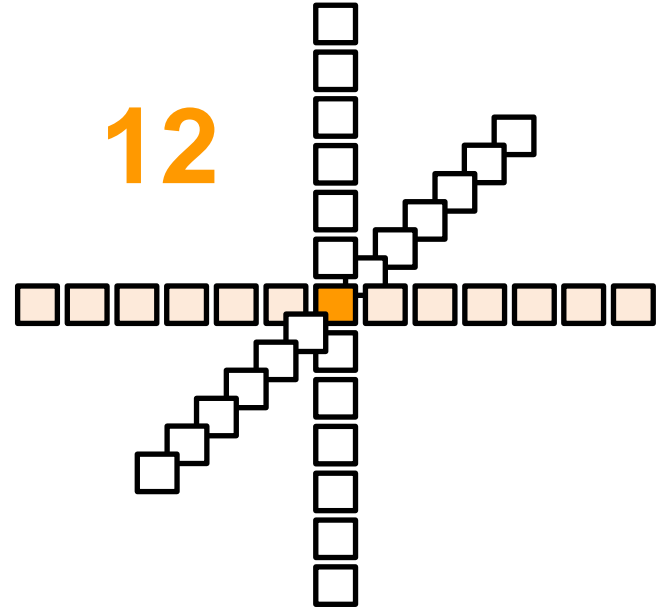
4



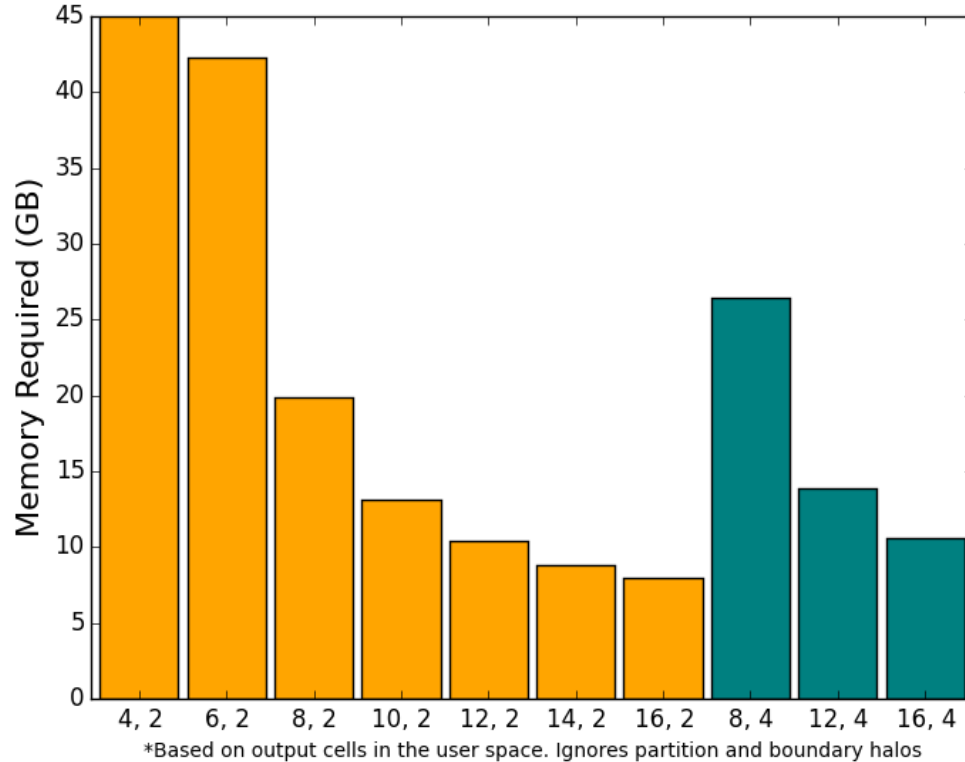
8



12



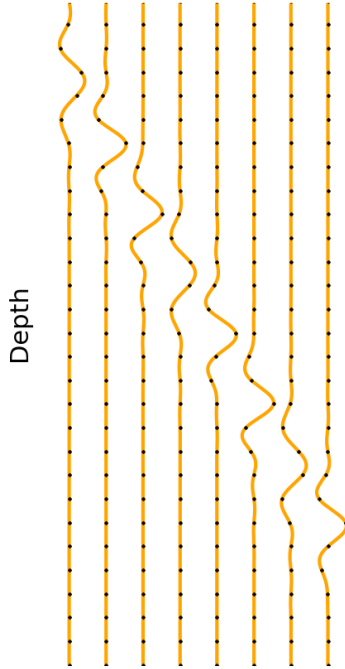
Single Shot Example



- Memory for a single shot Isotropic RTM
- Higher orders converge towards Nyquist

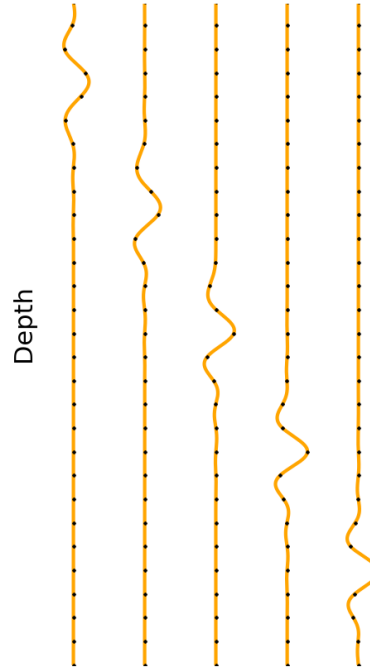
Effect of Temporal Order

Spatial Order = 16
Temporal Order = 2



Time

Spatial Order = 16
Temporal Order = 4



Time

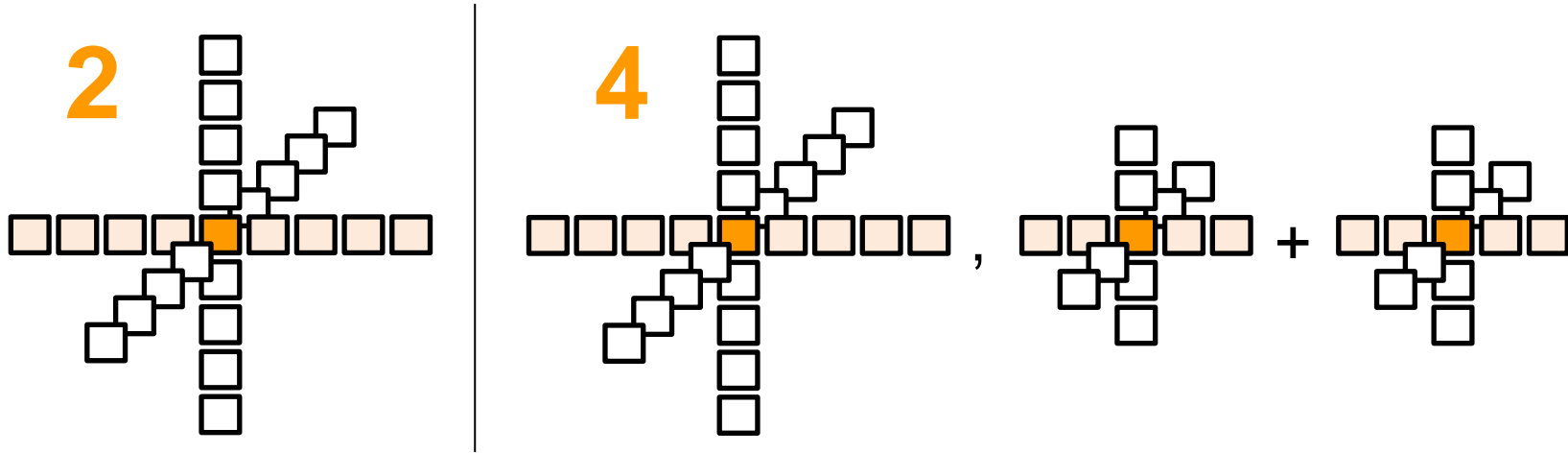
- Larger timestep
- Less iterations

Temporal Order Limitations

- Larger timestep means greater percentage of time is spent imaging
- Spatial orders compared using equal phase velocity error (temporal dispersion)
- 2nd order is always less accurate because timestep is limited by subtraction errors

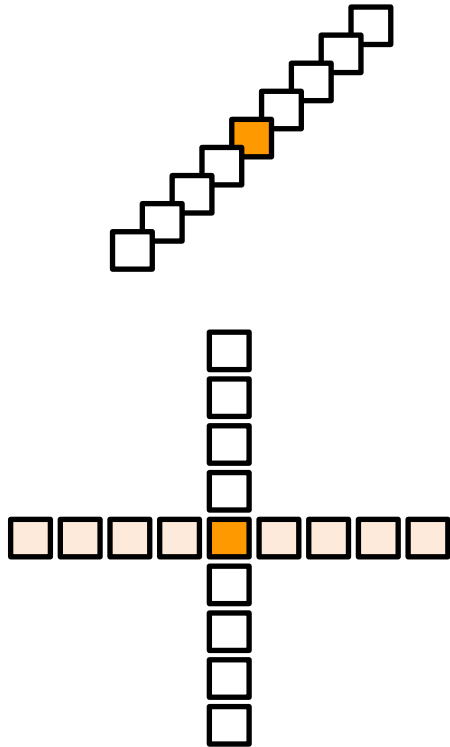
Temporal Order Algorithm

- Fourth order in time requires 3 passes through the volume per timestep



*There are also minor differences in illumination calculation and boundary conditions

CUDA Implementation

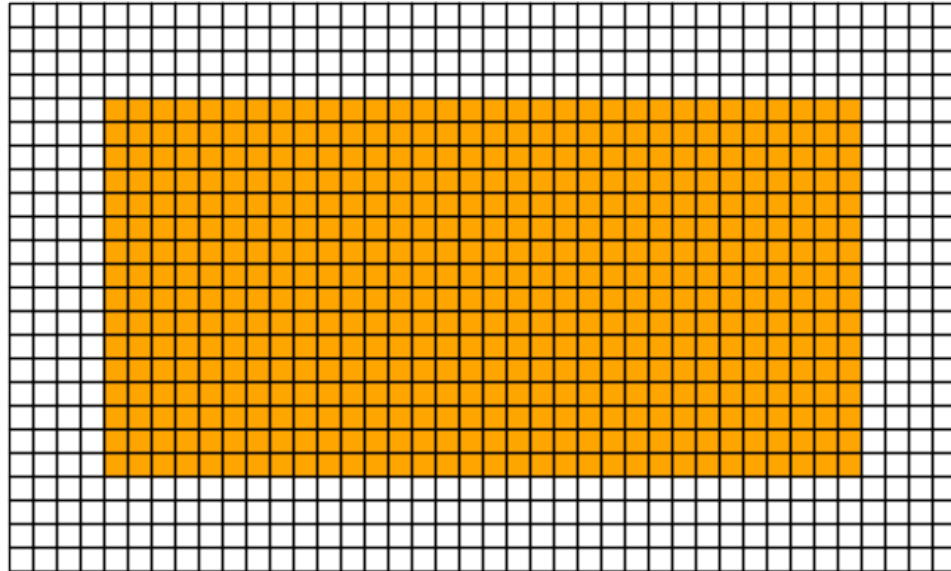


- Circular buffer of registers for slowest indexed axis

- Shared memory for fastest indexed axes

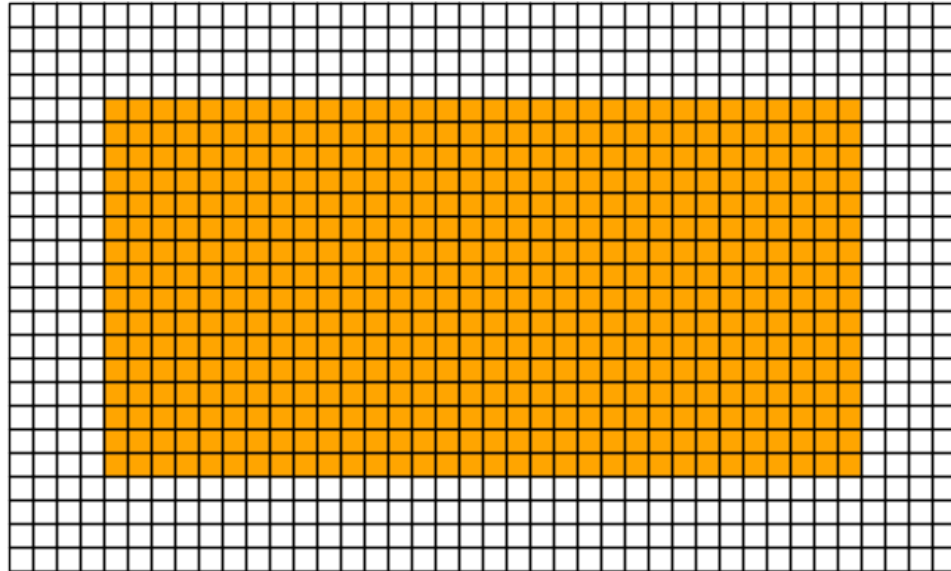
Shared Memory Loads

- Shared memory of a block, orange represents location of useful outputs



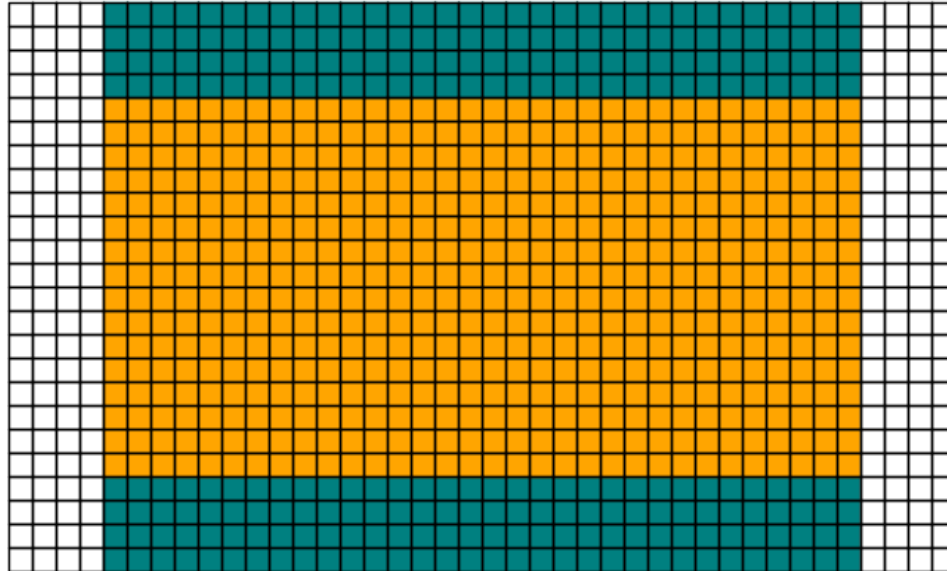
Shared Memory Loads

- Center of shared memory uses coalesced reads



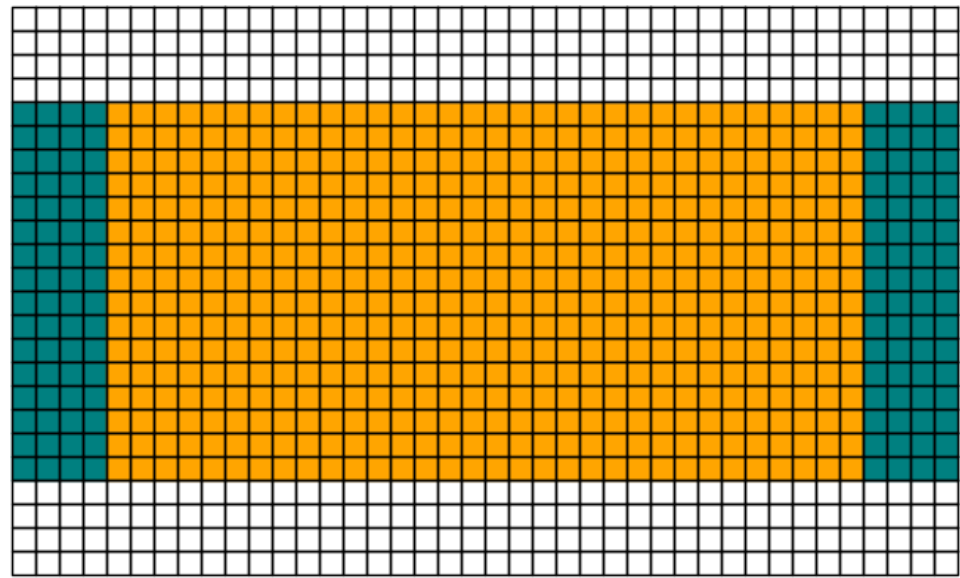
Shared Memory Loads

- Halos in strided direction are coalesced



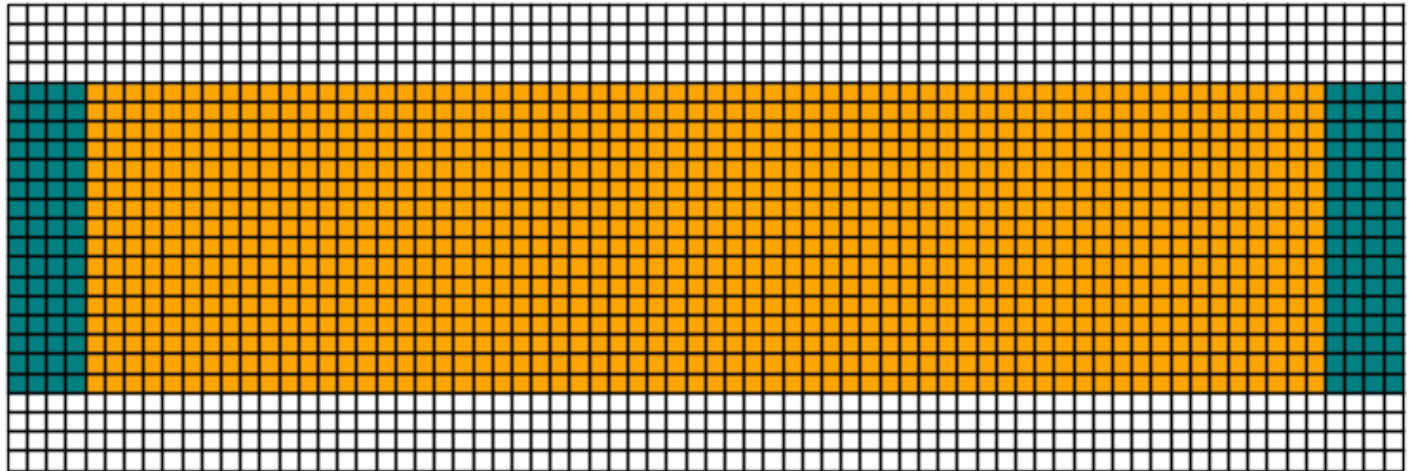
Shared Memory Loads

- Halos in unit stride direction are not coalesced

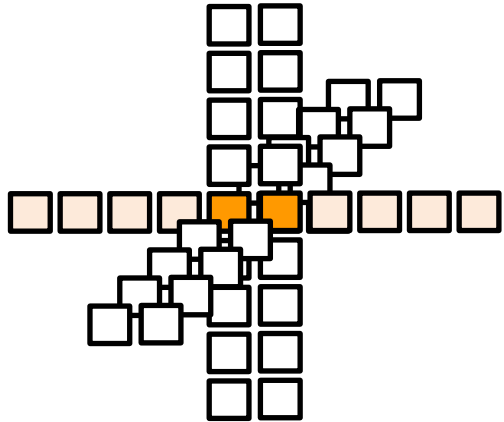


Shared Memory Loads

- Loading float2 reduces uncoalesced loads and the number of memory instructions



Loading With Float2



- Every CUDA thread must update 2 consecutive points
- Better memory access patterns
- Significantly increases register usage

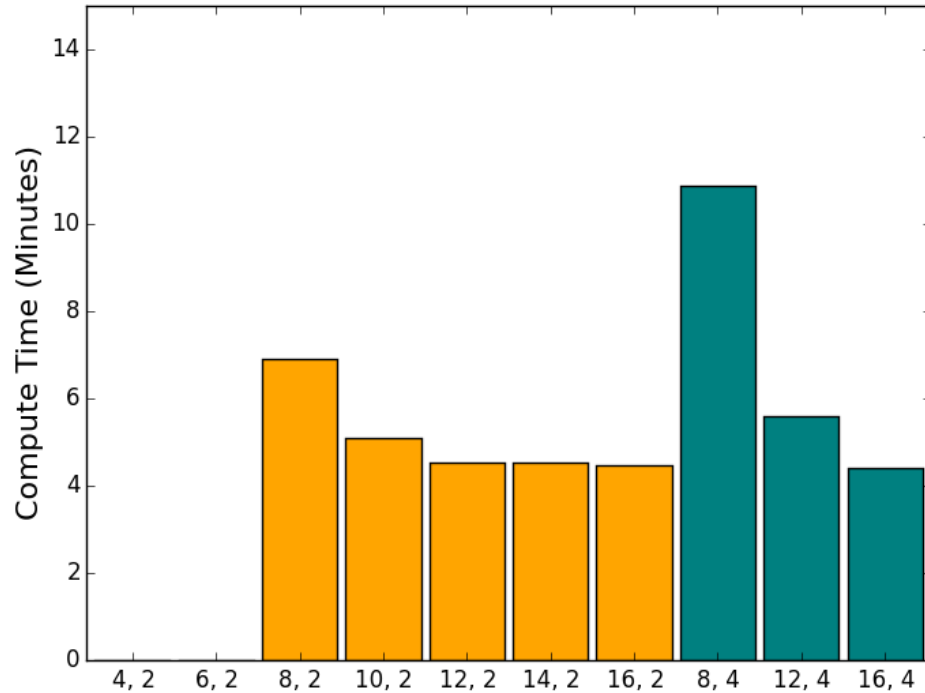
Limiting Register Usage

- High register usage allows the compiler to unroll loops and reduce instruction counts
- Limiting register usage allows better occupancy
- Lower spatial orders can afford higher instruction counts

Benchmarking Variations

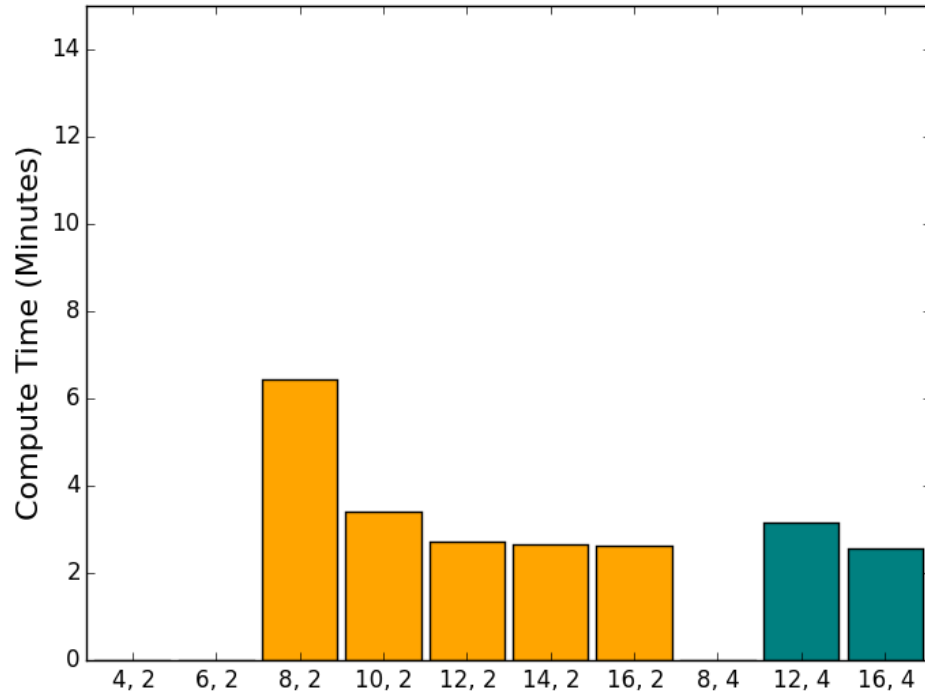
- End goal of RTM is to generate an image in the shortest possible time without compromising accuracy
- Benchmarked throughput time for a single shot with the same theoretical accuracy
- Tested various kernel parameters for each spatial order for each architecture

Benchmarking Variations



- K80
- 35 Hz
- 14 x 28 x 12 km

Benchmarking Variations



- K10
- 25 Hz
- 14 x 28 x 12 km

K10 to K80 Speed up

- K10 simulates 25 Hz shot in 2.5 minutes
- K80 simulates 35 Hz shot in 4.4 minutes
- K80 is 2.23 times faster than K10
 - Correcting for fourth power scaling with frequency
 - Assuming linear scaling of K10s

Conclusions

- 16th order in space and 4th order in time is faster than 8th order in space 2nd order in time
- Increased register and shared memory on new GPUs is making higher orders more affordable
- K80s run 2.23 times faster than K10s if reoptimization is included

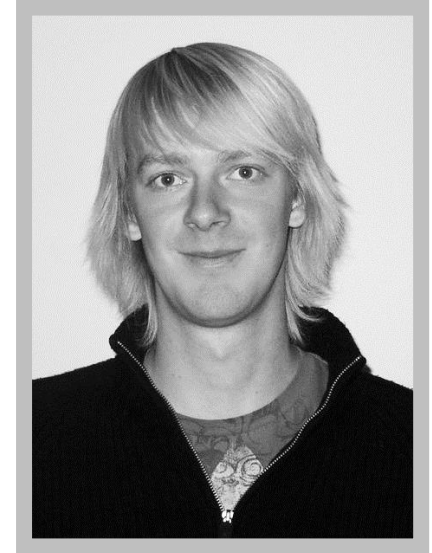
Questions?

Acceleware Ltd.

Tel: +1 403.249.9099

Email: services@acceleware.com

Website: <http://acceleware.com>



Marcel Nauta

marcel.nauta@acceleware.com