Ultrasound simulation on a GPGPU - Reducing memory bottleneck using compression

COMP8750 – Computer Systems Project

Anish Varghese
u5133234
Australian National University
Supervisor: Dr. Eric McCreath
Outline

- Background
- Challenges & Bottlenecks
- Compression Techniques
- OpenCL Introduction
- Implementation Approaches
- Evaluation & Results
- Conclusion
- Questions
Background

- Application of Ultrasound Simulation
  - Medical Ultrasonography - System design and development
  - Delivery of Therapeutic Ultrasound
- Westervelt Equation

Image Source: Andrew A. Haigh, Bradley E. Treeby, and Eric C. McCreath. Ultrasound Simulation on the Cell Broadband Engine Using the Westervelt Equation
Challenges & Bottlenecks

- Large number of floating point computations
- Memory bandwidth bottleneck – Transfer of data to and from CPU for computation of each grid point
- Data Compression
Data Compression Techniques

- FPC Algorithm - Prediction method to compress sequences of values
- GFC Algorithm for GPUs
- Fast Entropy Coding
- Discrete Cosine Transform
- Floating point to fixed point conversion
Floating point to Fixed point compression

- Normalize original number
- Remove exponent
- Shift significand
- Result in 2 or 3 bytes
- Overhead for compression
OpenCL

- Open framework for heterogeneous platforms
- Work-item, work-group
- Memory Hierarchy
Ultrasound Simulation Implementation

- Basic Kernel
  - Access Global Memory
  - 4 grid points per work-item
- Shared Memory Kernel
  - Access Local Memory of Work-group
  - Few elements overlapped across work-groups
  - 4 grid points per work-item
- Memory Accesses aligned to 4 bytes
Evaluation of Data Compression - Performance

- Performance better for large problem sizes
### Evaluation of Data Compression - Accuracy

- Average accuracy of 99.95% with 2 byte compression
- Average accuracy of 99.99% with 3 byte compression

<table>
<thead>
<tr>
<th>Original Float</th>
<th>2 Byte Compressed Float</th>
<th>3 Byte Compressed Float</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.65625000</td>
<td>5.65592480</td>
<td>5.65624237</td>
</tr>
<tr>
<td>-76.56250000</td>
<td>-76.56097412</td>
<td>-76.56250000</td>
</tr>
<tr>
<td>66.56754303</td>
<td>66.56646729</td>
<td>66.56754303</td>
</tr>
<tr>
<td>0.58958769</td>
<td>0.58746338</td>
<td>0.58957934</td>
</tr>
<tr>
<td>-23.00000000</td>
<td>-22.99753952</td>
<td>-22.99999237</td>
</tr>
<tr>
<td>80.23451233</td>
<td>80.23326111</td>
<td>80.23451233</td>
</tr>
<tr>
<td>1.45899999</td>
<td>1.45721436</td>
<td>1.45899260</td>
</tr>
<tr>
<td>-2.49505901</td>
<td>-2.49481201</td>
<td>-2.49505043</td>
</tr>
<tr>
<td>0.35689092</td>
<td>0.35603842</td>
<td>0.35688281</td>
</tr>
<tr>
<td>-0.24923000</td>
<td>-0.24922690</td>
<td>-0.24922690</td>
</tr>
</tbody>
</table>
Evaluation of Ultrasound Simulation - Performance

- Improvement of 20% with 2 byte compression for large grid sizes
Evaluation of Ultrasound Simulation - Bandwidth

- Bandwidth reduced by around 50%

<table>
<thead>
<tr>
<th>Grid Size</th>
<th>Baseline (MB/s)</th>
<th>Basic Kernel</th>
<th>Extension (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>97.0</td>
<td></td>
<td>26.4</td>
</tr>
<tr>
<td>256</td>
<td>168.4</td>
<td></td>
<td>53.2</td>
</tr>
<tr>
<td>512</td>
<td>250.8</td>
<td></td>
<td>110.2</td>
</tr>
<tr>
<td>1024</td>
<td>299.2</td>
<td></td>
<td>141.8</td>
</tr>
<tr>
<td>2048</td>
<td>323.8</td>
<td></td>
<td>153.1</td>
</tr>
<tr>
<td>2304</td>
<td>293.3</td>
<td></td>
<td>156.7</td>
</tr>
<tr>
<td>2560</td>
<td>253.4</td>
<td></td>
<td>156.7</td>
</tr>
<tr>
<td>2816</td>
<td>250.2</td>
<td></td>
<td>157.3</td>
</tr>
</tbody>
</table>
Evaluation of Ultrasound Simulation – Image Quality

With compression

Without compression
Conclusion

• Data compression using floating point to fixed point conversion
• Performance gain of 20% and upto 50% savings in memory bandwidth
• Useful for other domains as well
• Performance on better GPUs
• Future work – 3 dimensional simulation using compression
Questions

u5133234@anu.edu.au