Fast Evaluation of Helmholtz Potential on Graphics Processing Units (GPUs)

Shaojing Li*, Boris Livshitz* and Vitaliy Lomakin*
Department of Electrical and Computer Engineering, University of California, San Diego

Problem Formulation

Interactions between N elements in Electromagnetics

\[ u_j = \sum_{i=1}^{N} \frac{q_i q_j}{|r_i - r_j|^\alpha}, \quad j = 1, \ldots, N \]

- High computational cost of \(O(N)\)
- Needs to be accelerated!

Fast Methods:
- Fast Fourier Transform (FFT): \(O(N \log N)\) complexity
- Fast Multiple Method (FMM)\(^1\): \(O(N)\) or \(O(N \log N)\) complexity

Advanced Hardware and Software Techniques:
- Parallelization of fast methods is vital
- Multi-core/multi-CPU clusters have constraints
- Emerging systems: Graphics Processing Unit systems

Significance
- No GPU implementations of fast Electromagnetic integral-type have been reported (implementations of static FFT and FMM exist)
- Applications in Magnetics, Microwave, Astrophysics, Acoustics, Optics, Bioengineering, Chemistry

Non-uniform Grid Interpolation Method (NGIM)\(^3\)\(^4\)\(^5\)

Approach:
- Divide the computational domain into boxes on various levels
- Separate near and far fields calculations
- Use direct method for near-fields
- Field outside a source domain is smooth → Use sparse Non-uniform Grids (NG) to compute it

\(O(N)\) complexity is achieved for static and low-frequency fields
- Multilevel scheme is used

Advantages:
- Automatically adaptive to non-uniform source distributions
- Can handle low- and high-frequency problems → wide applicability
- No special functions required → Simplified code structure and faster
- Very fast, with GPU-CPU acceleration gains of 70-200
- Can be extended to different types of interactions (general kernels)

Computational Time

Low frequency, \(\lambda = 10^3 \Omega D\), \(L_2\) error = e-1

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>GPU NGIM Time (secs)</th>
<th>CPU NGIM Time (secs)</th>
<th>Speedup</th>
<th>Time (secs)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>65,536</td>
<td>0.0326</td>
<td>2.54</td>
<td>77.9</td>
<td>394</td>
<td>1.21e4</td>
</tr>
<tr>
<td>1,048,576</td>
<td>0.815</td>
<td>52.9</td>
<td>64.9</td>
<td>1.695e+05</td>
<td>2.07e5</td>
</tr>
<tr>
<td>16,777,072</td>
<td>8.90</td>
<td>666</td>
<td>74.8</td>
<td>4.386e+07</td>
<td>4.96e6</td>
</tr>
</tbody>
</table>

4900000x speed-up!
75x ~ 200x speed-up!
(Almost the same computational time for any source distribution!)

Future Work

- GPU implementation for the high-frequency regime (Partially done)
- Kernel independent “universal” algorithm, applicable to general kernel types (e.g. Helmholtz, Coulomb, London, Lennard-Jones, H-bonds)
- Multiple GPUs parallelization

Reference


2009 nVidia Research Summit, San Jose, CA