Towards Efficient Two-Level Preconditioned Conjugate Gradient on the GPU

International Conference on Preconditioning Techniques for Scientific and Industrial Applications
May 16-18, 2011, Bordeaux, France
Kees Vuik, Rohit Gupta
16-05-2011
Content

• Introduction
• GPU Computing
• Solver - Overview
• Preconditioning
• Deflation
• Results
• Conclusions
Bird’s Eye View

Comparison of CPU optimized code with respect to GPU code

Two Level Preconditioned CG
Single Precision

Run Time in Seconds

GPU

CPU
Bird’s Eye View

Comparison of CPU optimized code with respect to GPU code

23x
Problem Description

Mass-Conserving Level Set Method to Solve the Navier Stokes Equation.

Air bubbles rising in Water.
Computational Model

Computational Hotspot - Solution of the Pressure Correction equation in 2D

\[-\nabla \cdot \left( \frac{1}{\rho(x)} \nabla p(x) \right) = f(x), \quad x \in \Omega \]  

\[\frac{\partial}{\partial n} p(x) = g(x), \quad x \in \partial \Omega \]
Nature of the Coefficient Matrix

\[ Ax = b, \text{ where } x^T A x > 0 \quad (0) \]

\( A \) is a sparse matrix with a 5-point stencil. It has a very large condition number due to a huge jump in the density.

- Condition Number \( \rightarrow \kappa(A) := \frac{\lambda_n}{\lambda_1} \)
- Stopping criterion \( \rightarrow \frac{\| b - Ax_k \|_2}{\| r_0 \|} \leq 10^{-6} \)
Nature of the Coefficient Matrix

Huge jump at the interface due to contrast in densities.
A Brief Introduction to the GPU

- SIMD Architecture
- Large Memory Bandwidth
- User Managed Caches
A brief Intro.... Contd.

- Basic Unit of Execution - Thread
- Each Thread Executes a Kernel
- Aggregates of Threads = Blocks
- Shared Memory within a block
A brief Intro.... Contd.

```
  0 1 2 3
0  S S S S
1  S S S S
2  S S S S
3  S S S S
```

```
0  S S
1  S S
2  S S
3  S S
```
A brief Intro.... Contd.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>TeslaC1060</th>
<th>TeslaC2070 (Fermi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Compute Cores</td>
<td>240 cores</td>
<td>448 cores</td>
</tr>
<tr>
<td>Memory Bandwidth</td>
<td>102 Gb/s</td>
<td>144 Gb/s</td>
</tr>
<tr>
<td>Double Precision Throughput (Peak)</td>
<td>78 Gflops/s</td>
<td>515 Gflops/s</td>
</tr>
<tr>
<td>Memory</td>
<td>4GB</td>
<td>6GB</td>
</tr>
<tr>
<td>Shared Memory / L1 cache configurability</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
A brief Intro.... Contd.

Most Important Optimizations for GPU Code

- Reduce Host-GPU Transfers
- Maximize use of Memory Bandwidth
- Minimize Thread Divergence
- Utilize Shared Memory/ L1 cache based on kernel
Control Flow in the Algorithm

Conjugate Gradient with Two Level Preconditioning
Preconditioning

- Diagonal Preconditioning
- Block Incomplete Cholesky
- Incomplete Poisson (IP)
- Modifications based on IP
Preconditioning

Block Incomplete Cholesky Preconditioning

Within blocks the computation is sequential


16-05-2011
Preconditioning

Incomplete Poisson

\[ M = K \ast K^T, \text{ where } K = (I - L \ast D). \] (0)

\[ \text{Stencil for } A = (-1, -1, 4, -1, -1). \] (0)

\[ \text{Corresponding Stencil for } M^{-1} = \left( \frac{1}{4}, \frac{1}{16}, \frac{1}{4}, \frac{9}{8}, \frac{1}{4}, \frac{1}{16}, \frac{1}{4} \right) \] (0)

Drop the lowest terms (i.e. \( \frac{1}{16} \)).

\( M^{-1} \) has the same sparsity pattern as \( A \).

Degree of Parallelism for \( M^{-1} \ast r \) is N.

A Parallel Preconditioned Conjugate Gradient Solver for the Poisson Problem on a Multi-GPU Platform, M. Ament. PDP 2010
Preconditioning

Incomplete Poisson Variants $^a$

Scaling of $A$ matrix. $\hat{A} = D^{-\frac{1}{2}} \ast A \ast D^{-\frac{1}{2}}$

As parallel as IP and as effective as Block-IC
Slightly more computations compared to IP.

Deflation

Operations involved in deflation

\begin{align*}
\text{• } b &= Z^T \ast x \\
\text{• } m &= E^{-1}b \\
\text{• } w &= A \ast Z \ast m \\
\text{• } w &= x - w
\end{align*}

where, $E$ is the Galerkin Matrix and $Z$ is the matrix of deflation vectors.

\begin{flushleft}
\end{flushleft}
## Deflation

### Stripe-Wise Domains

<table>
<thead>
<tr>
<th></th>
<th>57</th>
<th>58</th>
<th>59</th>
<th>60</th>
<th>61</th>
<th>62</th>
<th>63</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>56</td>
</tr>
<tr>
<td>41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>
Deflation

Efficient Data Structures
Deflation

Efficient Data Structures

This data Structure has the advantages of the DIA Storage format\textsuperscript{a}.

\textsuperscript{a}Efficient Sparse Matrix-Vector Multiplication on CUDA. N. Bell and M. Garland, 2008, NVIDIA Corporation, NVR-2008-04
Deflation

- Breaking Up of Operations
- Stripe-Wise Domains
- Efficient Data Structures
Results

Single Precision Experiments

SpeedUp and Convergence across Grid Sizes - Poisson Type Problem

Deflation Vectors are of size $8n$. Precision Criteria is $10^{-5}$ till $260k$ and for $1m$ it is $10^{-4}$
Results

Single Precision Experiments
Wall Clock Times across Grid Sizes - Poisson Type Problem

Wall Clock Times across Grid Sizes

---

Deflation Vectors are of size $8n$. Precision Criteria is $10^{-5}$ till 260K and for 1m it is $10^{-4}$
Results

Single Precision Experiments
Accuracy across Grid Sizes - Poisson Type Problem

Deflation Vectors are of size $8n$. Precision Criteria is $10^{-5}$ till $260k$; and for $1m$ it is $10^{-4}$.
Results

IP Variants For two Phase Problem - Double Precision

Deflation Vectors are of size $2n$. Precision Criteria is $10^{-6}$.

Block size is $2n$ for $n \times n$ grid where $N = n \times n$ and $N$ is the number of unknowns.
Results

IP Variants For two Phase Problem - Double Precision

Deflation Vectors are of size $2n$. Precision Criteria is $10^{-6}$.

Block size is $2n$ for $n \times n$ grid where $N = n \times n$ and $N$ is the number of unknowns.
Results

Double Precision Experiments

SpeedUp and Convergence across GridSizes - Two-Phase Problem

Convergence Rate and SpeedUp across Grid Sizes

Deflation Vectors are $8n$. Density Contrast (1000 : 1). Precision Criteria is $10^{-6}$. 
Results

Double Precision Experiments
Wall Clock Times across Grid Sizes - Two-Phase Problem

Wall Clock Times across Grid Sizes

Deflated Preconditioned CG.

<table>
<thead>
<tr>
<th>Number of Unknowns</th>
<th>Time in Seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>~16k</td>
<td>0.01</td>
</tr>
<tr>
<td>~65k</td>
<td>1</td>
</tr>
<tr>
<td>~260k</td>
<td>10</td>
</tr>
<tr>
<td>~1m</td>
<td>100</td>
</tr>
</tbody>
</table>

Deflation Vectors are $8n$. Density Contrast (1000 : 1). Precision Criteria is $10^{-6}$. 

---

16-05-2011
Results

Double Precision Experiments
Accuracy across Grid Sizes - Two-Phase Problem

![Result Accuracy across Grid Sizes](image)

Deflation Vectors are $8n$. Density Contrast $(1000 : 1)$. Precision Criteria is $10^{-6}$. 
Conclusions

• Deflation is highly parallelizable.

• Suitable Preconditioning for GPU must be used.

• More optimizations in order for Double Precision results.
Further Information

• Masters Thesis of Rohit Gupta
  [http://ta.twi.tudelft.nl/users/vuik/numanal/gupta_eng.html](http://ta.twi.tudelft.nl/users/vuik/numanal/gupta_eng.html)

• GPU page
  [http://ta.twi.tudelft.nl/users/vuik/gpu.html](http://ta.twi.tudelft.nl/users/vuik/gpu.html)

• Open Source GPU software
  [http://ta.twi.tudelft.nl/users/vuik/gpu.html#software](http://ta.twi.tudelft.nl/users/vuik/gpu.html#software)