GPU Acceleration of a Non-Hydrostatic Ocean Model with Lagrangian Particle Tracking

Takateru Yamagishi(1) Yoshimasa Matsumura(2)

1: Research Organization for Information Science and Technology, yamagishi@rist.jp 2: Institute of Low Temperature Science, Hokkaido University, ymatsu@iis.hokudai.ac.jp

Background and Objective

Multiphase flow: significant process in the ocean
- Formation of grease ice
- Ocean ecosystem, etc.

Matsumura and Ohshima (2015)
- Explicit expression of the particles via the Lagrangian particle tracking to a non-hydrostatic ocean model on a GPU

GPU can accelerate particle tracking ocean model, but no previous studies have tackled.

Our objectives:
- Accelerate the tracking particle ocean model using NVIDIA GPU
- Demonstrates some experimental cases and techniques that would be applicable to other studies

Non-hydrostatic ocean model “kinaco”

kinaco [1] can resolve vertical convection, eddy mixing with non-hydrostatic approximation on ~1 km scale

Particle tracking on Lagrangian form

velocity: Eulerian form
particle: Lagrangian form

Advect via a fourth-order Runge-Kutta method
\[ \mathbf{x}_n(t + \Delta t) = \mathbf{x}_n(t) + \Delta t \sum b_i f_i (\mathbf{v}_n(t)) \]

\( \mathbf{x}_n \): location of the particle (n: index)
\( \mathbf{v}_n \): subgrid velocity at \( \mathbf{x}_n \)
\( b_i, f_i \): coefficient and procedure of RK4 method
\( \Delta t \): time step

Implementation of particle tracking

Coalesced access to Velocity array \( V \).

Sorting by i-j order

We assigned the arrays of the ocean current velocity field to the texture cache to reduce redundant accesses to the GPU global memory, especially in cases where the accesses are irregular and noncontiguous amongst the threads.

Ocean dynamics kernel optimization

The pseudo CPU code indicating the essence of the ocean dynamics kernel.

Experiments and Results

Basic performance metrics for the CPU/GPU executions (dp: double precision, sp: single precision)

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>GPU</th>
<th>speed up</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>138.8</td>
<td>45.8</td>
<td>3.0</td>
</tr>
<tr>
<td>MGCG solver</td>
<td>44.0</td>
<td>16.0</td>
<td>2.8</td>
</tr>
<tr>
<td>Dynamics</td>
<td>47.9</td>
<td>14.0</td>
<td>3.4</td>
</tr>
<tr>
<td>Particle tracking</td>
<td>46.9</td>
<td>15.0</td>
<td>3.1</td>
</tr>
</tbody>
</table>

The cost of sorting the (256, 256, 64) grids was approximately 7 ms per sorting, which is approximately 5% of a particle-tracking cycle.

Discussion and Conclusions

- CPUs are useful not only for ocean dynamics but also for particle tracking in the ocean, which has not been represented in previous studies.
- The change in the algorithm and the use of a texture cache can accelerate particle tracking in the ocean.
- Exploiting thread-level parallelism with the help of an additional calculation and the use of registers is effective for the ocean dynamics kernels.
- The output from the GPU showed a good reproduction of the ocean particles’ distribution.
- Further performance analyses and validations with various experimental settings are required.
- Basic metric profiles show that the values of the metrics are less than the theoretical peak performances and that there is still room for improvement on both the CPU and GPU.

References