The WRF Single-Moment 6-Class Microphysics’ (WSM6) Performance Optimization
Authors: Timbwaoga Aime Judcael Ouermi (TAJO), Aaron Knoll, Robert M Kirby, Martin Berzins

Introduction:
WSM6 = Weather Research Forecasting Single-Moment 6-class Microphysics.

The “microphysics” scheme is a physical parametrization that simulates processes in the atmosphere that cause precipitation of rain, snow, graupel, water vapor, cloud water, cloud ice.

Methodology:

1. Basic Loop Analysis
   - Understand loop behavior
   - Loop sizes
   - Data dependency

2. Simple Optimization
   - Add compiler flags: qopenmp, qopenmp-simd and architecture dependent FLAGS
   - Basic timing
   - Align array to 64-byte (compiler flag)
   - Add OpenMP (OMP) directives

3. Profiling & Vectorization
   - Advisor, Vtune, compiler optrpt
   - Identify non-vectorized loop
   - Re-write code for auto vectorization

4. Threading & Directives Analysis
   - Understand directives overhead
   - Apply appropriate directives to WSM6
   - Code modification

Vectorization Efforts:
Initial addition of directives:
- $OMP DO SIMD
- Alignment of arrays in code
- !DIR$ ASSUME_ALIGNED
- !DIR$ ATTRIBUTES ALIGN
- Other code changes (via vtune):
  - removed initialization loop
  - removed outer timestep loop

SIMD results (VTune)
- Main wsm6 loop (70% of cost) vectorizes with OMP SIMD (AVX2) -- 28% efficiency
- rain_plm6 (15%), rain_plm(15%) did not vectorize with OMP SIMD
- Unaligned accesses remain (optrpt)
- Overall 21% gain in performance over autovectorized code

Results:
Directives’ Overhead in µs

<table>
<thead>
<tr>
<th># Threads</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>-15.6</td>
<td>6.75</td>
<td>14.79</td>
<td>12.98</td>
<td>12.98</td>
<td>8.98</td>
<td>10.65</td>
</tr>
<tr>
<td>Parallel Do</td>
<td>340.41</td>
<td>160.51</td>
<td>100.57</td>
<td>46.45</td>
<td>32.1</td>
<td>20.11</td>
<td>26.49</td>
</tr>
<tr>
<td>Do</td>
<td>-19.56</td>
<td>16.69</td>
<td>1.21</td>
<td>2.31</td>
<td>4.43</td>
<td>3.68</td>
<td>6.41</td>
</tr>
</tbody>
</table>

Compiler flags optimization
WSM6 measurement in s

| NO FLAGS | 1.96 |
| FLAGS (AUTO-VECTORIZATION) | 0.46 |

Compiler Flags + Threading

Future Work:
- Re-write loops to get vectorization
- Thread scalability and task parallelism
- Examine KNL performance

Acknowledgements:
We want to acknowledge and thank the parties below for their support in this effort.
- Department Of Defense PETTT program
- Rajiv Bendale and Hugh Thorbunberg at Engility Corporation,
- Alex Reinecke and Kevin Viner at NRL