Communication and Vectorization within HPGMG

Vladimir Marjanovic
HPGMG in EU Project

- Mont Blanc 3 – hardware / software co-design

- HPGMG representative app used in app WP of MB3

- Profiling : Communication and Vectorization
  - Bandwidth (B/s)
  - Latency (s)
  - Message Injection Rate(1/s)
HPGMG Code

- Well written balanced codes MPI+OpenMP
- HPCG, HPGMG complexity:
  - Computational $O(n)$ communication $O(n^{2/3})$
- MG many levels

- mpirun ./hpgmg-fv SIZE_OF_BLOCK BLOCKs_PER_MPI

Perfect MPI data distribution:
$(\text{NUM\_OF\_MPIs} \times \text{BLOCKs\_PER\_MPI})^{1/3} = \text{INTEGER\_NUMBER}$
Communication Profiling: Tracing

- Trace 4MPI + 6OpenMP
- 25 OpenMP section
- MG – V cycle

- MPI:
  - MPI_Allreduce
  - ExchangeHalos
MPI_Allreduce 12288 MPIs

- XC40 - Intel Haswell E5-2680v3
- 24 cores per node
- HPGMG input 6 9
- $24 \times 512 \times 9 = 48 \times 48 \times 48$ perfect cube

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average</strong></td>
<td>2453728ns</td>
</tr>
<tr>
<td><strong>Max</strong></td>
<td>3907503 ns</td>
</tr>
<tr>
<td><strong>Min</strong></td>
<td>34976 ns</td>
</tr>
<tr>
<td><strong>StDev</strong></td>
<td>508181 ns</td>
</tr>
<tr>
<td><strong>Avg/Max</strong></td>
<td>0.63</td>
</tr>
</tbody>
</table>

::V.Marjanovic BoF@SC16 HPGMG:: 16.11.2016 ::
Comm Pattern: ExchangeHalos 12288 MPIs

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max(bytes)</td>
<td>624640</td>
</tr>
<tr>
<td>Min(bytes)</td>
<td>2048</td>
</tr>
<tr>
<td>MaxNumOfNeighbor</td>
<td>24</td>
</tr>
</tbody>
</table>
Unbalancing: ExchangeHalos 12288 MPIs

<table>
<thead>
<tr>
<th></th>
<th>2⁶</th>
<th>12</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2⁵</td>
<td>24</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>2⁴</td>
<td>26</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td>2³</td>
<td>108-360</td>
<td>35-all</td>
</tr>
<tr>
<td></td>
<td>2²</td>
<td>108</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>120</td>
<td>3</td>
</tr>
</tbody>
</table>

2⁶ unbalancing issue ~ 25%

---

Average: 15514163ns
Max: 20811766ns
Min: 11503473ns
StDev: 847420ns
Avg/Max: 0.75

:: V. Marjanovic BoF@SC16 HPGMG :: 16.11.2016 ::
Exchange Halos: Histogram of time duration

$2^6$

$2^5$

$2^4$

$2^3$
Conclusion: HPGMG Communication

- Large number of message.
- Stresses latency, routing
- Data distribution: perfect cube with input data
- Synchronization Issue
Vectorization of HPGMG : SX-ACE NEC

• SX – ACE NEC vector processor
• 4 cores for 256GFlops peak performance
• 1 GHz
• 16 vectore pipes per core
• 32 memory channels for 256GB/s bandwidth, one core or sharable
• Assignable Data Buffer (ADB) – Vector Cache 1MB
HPGMG-FV NEC-ACE versus CRAY XC30 no opt

Ivy Bridge 5576
NEC SX-ACE
• Reference code
• no optimization
• Data distribution 8 2
• 4MPI vs 4MPI+6OpenM
NEC-ACE Profiler

<table>
<thead>
<tr>
<th>PROC.NAME</th>
<th>FREQUENCY</th>
<th>EXCLUSIVE</th>
<th>AVER.TIME</th>
<th>MOPS</th>
<th>MFLOPS</th>
<th>V.OF</th>
<th>AVER.</th>
<th>VECTOR</th>
<th>I-CACHE</th>
<th>O-CACHE</th>
<th>BANK</th>
<th>CONFLICT</th>
<th>ADB HIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>smooth</td>
<td>25940</td>
<td>111.814(54.3)</td>
<td>4.311</td>
<td>34115.5</td>
<td>15630.4</td>
<td>97.67</td>
<td>58.6</td>
<td>108.565</td>
<td>0.173</td>
<td>0.928</td>
<td>0.002</td>
<td>31.061</td>
<td>83.93</td>
</tr>
<tr>
<td>residual</td>
<td>17588</td>
<td>29.260(44.2)</td>
<td>1.564</td>
<td>36589.9</td>
<td>14159.3</td>
<td>99.12</td>
<td>131.5</td>
<td>25.883</td>
<td>0.018</td>
<td>0.136</td>
<td>0.000</td>
<td>14.702</td>
<td>0.00</td>
</tr>
</tbody>
</table>

::V.Marjanovic BoF@SC16 HPGMG:: 16.11.2016 ::
SX-ACE Code Optimization

```c
#pragma cdir on_adb(x)
    for(k=klo;k<khi;k++){
        #pragma cdir on_adb(x)
        for(j=jlo;j<jhi;j++){
            #pragma cdir on_adb(x)
            for(i=ilo;i<ihi;i++){
                int ijk = i + j*jStride + k*kStride;
                double Ax = apply_op_ijk(x);
                res[ijk] = rhs[ijk]-Ax;
            }
        }
    }
```

- Simple code modification
- Pragma ADB hint
- Keep data in vector cache
HPGMG-FV NEC-ACE versus CRAY XC30 with opt

- Copt
- use ADB
- 7 days
- NEC support
- 17% improvement
Node Comparison

- Memory BW
- Code Optimization
- Number of cores does not help
- Different performance for different routines
Conclusion: Vectorization

- Reference code very good well written for vectorization
- HPGMG stresses vector unit and compiler
- HPGMG shows performance issue on vectorization for small problem sizes