Design Considerations for Latency and Throughput on KNL

Jed Brown jed@jedbrown.org (CU Boulder)
Collaborators: Karl Rupp, Satish Balay, Matthew Knepley, Richard Mills, Barry Smith

MultiCore6, 2016-09-14
Scaling goals

Aurora ESP: Evaluation of Proposals

- An existing or reasonably well-planned implementation to make use of thread concurrency on Aurora. ALCF expects to see strong-scaling up to at least 8 threads per MPI rank, with greater than 75% efficiency. – https://www.alcf.anl.gov/programs/aurora-esp

- How much memory bandwidth is achievable with 8 threads on KNL/7210?
  - MCDRAM: 110 GB/s (of 420 GB/s)
  - DRAM: 80 GB/s (of 88 GB/s)
Scaling goals

Aurora ESP: Evaluation of Proposals

- An existing or reasonably well-planned implementation to make use of thread concurrency on Aurora. ALCF expects to see strong-scaling up to at least 8 threads per MPI rank, with greater than 75% efficiency. – https://www.alcf.anl.gov/programs/aurora-esp

- How much memory bandwidth is achievable with 8 threads on KNL/7210?
  - MCDRAM: 110 GB/s (of 420 GB/s)
  - DRAM: 80 GB/s (of 88 GB/s)
But affinity is hard

- MPI in Flat:Quadrant worked with default settings – 110 GB/s for 8x1
- MPI in Flat:SNC-4 seems to require manual enumeration
  
  mpiexec -n 8 -env I_MPI_PIN_PROCESSOR_LIST
  0,2,18,20,36,38,52,54 numaclt -m 4,5,6,7
- OpenMP with all tested variants of affinity flags – 63 GB/s for 1x8
  
  - My Intel colleague was adamant this was the best possible and I must have
    a bug in my MPI test.

- KMP_AFFINITY='explicit,proclist=[0,8,16,24,32,40,48,56],granularity=core'
  OMP_NUM_THREADS=8 numaclt -m 1 ./stream – 110 GB/s
NUMA architecture: E5-2699v4
- Cores are not in a NUMA domain.
Automatic NUMA migration

- Linux feature for a few years now (Rik van Riel, Red Hat)

NUMA page migration

- NUMA page faults are relatively cheap
- Page migration is much more expensive
  - ... but so is having task memory on the “wrong node”
- Quadratic filter: only migrate if page is accessed twice
  - From same NUMA node, or
  - By the same task
  - CPU number & low bits of pid in page struct
- Page is migrated to where the task is running
Scaling regime: HPGMG-FE on Edison, SuperMUC, Titan

HPGMG-FE Performance

- edison np=131072
- supermuc np=140608
- titan np=262144

DOF/s

Climate SYPD goal

Titan >200ms

1.6B

12.9B

155B

309B
HPGMG-FE: Broadwell and KNL

HPGMG-FE Performance

- broadwell-opt np=40
- knl-opt np=128
- knl-opt np=64

MEquations/second vs. Solve time (s)

GFlop/s
### HPGMG-FE: Broadwell profile

<table>
<thead>
<tr>
<th>% cumulative</th>
<th>time</th>
<th>seconds</th>
<th>cumulative</th>
<th>self</th>
<th>seconds</th>
<th>calls</th>
<th>s/call</th>
<th>total</th>
<th>self</th>
<th>seconds</th>
<th>calls</th>
<th>s/call</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.87</td>
<td>3.71</td>
<td>3.71</td>
<td>1979816</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>DMFEExtractElements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.80</td>
<td>7.22</td>
<td>3.51</td>
<td>2330024</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>OpPointwiseElement_Poisson</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.78</td>
<td>9.98</td>
<td>2.76</td>
<td>14040600</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>TensorContract_FMA_8_1_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.82</td>
<td>12.56</td>
<td>2.58</td>
<td>676</td>
<td>0.00</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>OpApply_Poisson</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.89</td>
<td>14.22</td>
<td>1.66</td>
<td>3176520</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>TensorContract_FMA_8_3_3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.82</td>
<td>15.68</td>
<td>1.46</td>
<td>1023800</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>DMFESetElements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.32</td>
<td>16.30</td>
<td>0.62</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>DMGlobalToLocalEnd_FE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.27</td>
<td>16.91</td>
<td>0.61</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>DMLocalToGlobalEnd_FE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.62</td>
<td>17.40</td>
<td>0.49</td>
<td>168</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>DMFERestrict</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.71</td>
<td>17.72</td>
<td>0.32</td>
<td>168</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>DMFEInterpolate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### HPGMG-FE: KNL profile

<table>
<thead>
<tr>
<th>% cumulative</th>
<th>self time</th>
<th>self seconds</th>
<th>self calls</th>
<th>self s/call</th>
<th>total s/call</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>21.01</td>
<td>5.06</td>
<td>5.06</td>
<td>647544</td>
<td>0.00</td>
<td>0.00</td>
<td>DMFEExtractElements</td>
</tr>
<tr>
<td>15.95</td>
<td>8.90</td>
<td>3.84</td>
<td>396</td>
<td>0.01</td>
<td>0.04</td>
<td>OpApply_Poisson</td>
</tr>
<tr>
<td>13.64</td>
<td>12.19</td>
<td>3.29</td>
<td>5220144</td>
<td>0.00</td>
<td>0.00</td>
<td>TensorContract_AVX512_8_1_3_3</td>
</tr>
<tr>
<td>12.96</td>
<td>15.31</td>
<td>3.12</td>
<td>865620</td>
<td>0.00</td>
<td>0.00</td>
<td>OpPointwiseElement_Poisson</td>
</tr>
<tr>
<td>12.15</td>
<td>18.23</td>
<td>2.93</td>
<td>1049004</td>
<td>0.00</td>
<td>0.00</td>
<td>TensorContract_AVX512_8_1_3_3</td>
</tr>
<tr>
<td>8.10</td>
<td>20.18</td>
<td>1.95</td>
<td>336528</td>
<td>0.00</td>
<td>0.00</td>
<td>DMFESetElements</td>
</tr>
<tr>
<td>2.45</td>
<td>20.77</td>
<td>0.59</td>
<td>5497632</td>
<td>0.00</td>
<td>0.00</td>
<td>OpPointwiseForcing_Poisson</td>
</tr>
<tr>
<td>2.37</td>
<td>21.34</td>
<td>0.57</td>
<td>140</td>
<td>0.00</td>
<td>0.00</td>
<td>DMFERestrict</td>
</tr>
<tr>
<td>1.95</td>
<td>21.81</td>
<td>0.47</td>
<td>140</td>
<td>0.00</td>
<td>0.00</td>
<td>DMFEInterpolate</td>
</tr>
<tr>
<td>1.95</td>
<td>22.28</td>
<td>0.47</td>
<td></td>
<td></td>
<td></td>
<td>DMGlobalToLocalEnd_FE</td>
</tr>
<tr>
<td>1.87</td>
<td>22.73</td>
<td>0.45</td>
<td></td>
<td></td>
<td></td>
<td>DMLocalToGlobalEnd_FE</td>
</tr>
</tbody>
</table>
HPGMG-FE: KNL assembly

```
vmulpd zmm1,zmm3,ZMMWORD PTR [r12+r14*1+0x40]
vmulpd zmm2,zmm3,ZMMWORD PTR [r12+r14*1+0x80]
vmulpd zmm4,zmm3,ZMMWORD PTR [r12+r14*1+0xc0]
vbroadcastsd zmm5,QWORD PTR [r15+rsi*1+0x8]
vmovups ZMMWORD PTR [rax+0x40],zmm5
vfmadd231pd zmm0,zmm5,ZMMWORD PTR [r12+r14*1+0x100]
vmovups ZMMWORD PTR [rdi+r9*1],zmm0
vfmadd231pd zmm1,zmm5,ZMMWORD PTR [r12+r14*1+0x140]
vmovups ZMMWORD PTR [rdi+r9*1+0x40],zmm1
vfmadd231pd zmm2,zmm5,ZMMWORD PTR [r12+r14*1+0x180]
vmovups ZMMWORD PTR [rdi+r9*1+0x80],zmm2
vfmadd132pd zmm5,zmm4,ZMMWORD PTR [r12+r14*1+0x1c0]
vmovups ZMMWORD PTR [rdi+r9*1+0xc0],zmm5
```
Outlook

- MCDRAM is likely all we need for strong scaling
- SNC-4 requires manual affinity for MCDRAM (due to NUMA mapping)
- If the kernel understood the actual memory architecture, perhaps it could automate placement
- TLB effect? 400 GB/s with MPI versus 420 GB/s with threads
- Irregular access/packing is becoming more expensive