Evaluation of Intel Xeon Phi "Knights Landing": Initial impressions and benchmarking results

J. Eitzinger

PRACE PATC, 28.6.2017
History of Intel hardware developments

[Graph showing the progression of Intel processors from 1994 to 2016, highlighting the trade-off between peak bandwidth and peak flops with a note on multicore versus single-core design.]
The real picture
Finding the right compromise

Core complexity

SIMD

Frequency

# cores

Intel Skylake-EP

Intel KNL

Nvidia GP100
Maximum DP floating point (FP) performance

\[ P_{\text{core}} = n_{\text{super}}^{FP} \cdot n_{\text{FMA}} \cdot n_{\text{SIMD}} \cdot f \]

<table>
<thead>
<tr>
<th>uArch</th>
<th>( n_{\text{super}}^{FP} )</th>
<th>( n_{\text{FMA}} )</th>
<th>( n_{\text{SIMD}} )</th>
<th>( n_{\text{cores}} )</th>
<th>Release</th>
<th>Model</th>
<th>( P_{\text{core}} ) [GF/s]</th>
<th>( P_{\text{chip}} ) [GF/s]</th>
<th>( P_{\text{serial}} ) [GF/s]</th>
<th>TDP</th>
<th>GF/Watt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sandy Bridge</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>8</td>
<td>Q1/2012</td>
<td>E5-2680</td>
<td>11.7</td>
<td>173</td>
<td>7</td>
<td>130</td>
<td>1,33</td>
</tr>
<tr>
<td>Ivy Bridge</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>10</td>
<td>Q3/2013</td>
<td>E5-2690-v2</td>
<td>24</td>
<td>240</td>
<td>7.2</td>
<td>130</td>
<td>1,85</td>
</tr>
<tr>
<td>KNC</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>61</td>
<td>Q2/2014</td>
<td>7120A</td>
<td>10.6</td>
<td>1210</td>
<td>1,33</td>
<td>300</td>
<td>4,03</td>
</tr>
<tr>
<td>Haswell</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>14</td>
<td>Q3/2014</td>
<td>E5-2695-v3</td>
<td>21.6</td>
<td>425</td>
<td>6.6</td>
<td>120</td>
<td>3,54</td>
</tr>
<tr>
<td>Broadwell</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>22</td>
<td>Q1/2016</td>
<td>E5-2699-v4</td>
<td>17.6</td>
<td>704</td>
<td>7.2</td>
<td>145</td>
<td>4,85</td>
</tr>
<tr>
<td>Pascal</td>
<td>1</td>
<td>2</td>
<td>32</td>
<td>56</td>
<td>Q2/2016</td>
<td>GP100</td>
<td>36.8</td>
<td>4700</td>
<td>1,5</td>
<td>300</td>
<td>15,67</td>
</tr>
<tr>
<td>KNL</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>72</td>
<td>Q4/2016</td>
<td>7290F</td>
<td>35.2</td>
<td>2995</td>
<td>3.4</td>
<td>260</td>
<td>11,52</td>
</tr>
<tr>
<td>Skylake</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>26</td>
<td>Q3/2017</td>
<td>8170</td>
<td>23.4</td>
<td>1581</td>
<td>7.6</td>
<td>165</td>
<td>9,58</td>
</tr>
</tbody>
</table>
Chebyshev Filter Diagonalization on KNL and P100

DFG SPPEXA Essex 2 project
Basic ChebFD scheme

1. Filter $n_s$ search vectors
2. Orthogonalize $n_s$ search vectors
3. Go to 1 if not converged

- $n$: matrix/vector dimension
- $n_p$: polynomial degree (defined by application)
- $n_s$: number of search vectors (defined by application)

Test systems
- **Piz Daint** (Switzerland): 5320 Node Cray XC50
- **Oakforst-PACS** (Japan): 8208 Node Fujitsu PRIMERGY
Node-level Performance

**KNL**

- MKL
- GHOST
- GHOST

**P100**

- cuBLAS/Sparse
- GHOST
- GHOST

ChebFD+KPM performance (GFLOP/s)

Block vector width $n_b$

Number of vectors $n_b$

$2\times$ HSW
Scaling Results

Fujitsu PRIMERGY

Cray XC50
System configuration challenge
Configuration complexity

- **Cluster modes**: lower the latency and increase the bandwidth
  - All-to-all
  - Quadrant mode (default)
  - Sub-numa-clustering (SNC), best performance but explicit

- **Memory modes**:
  - Cache mode (default)
  - Flat mode (explicit)
  - Hybrid

- **Mapping** of application on hardware:
  - Use SMT or not. How many SMT threads?
  - Use all cores?
  - MPI+X. How exactly?

- **Memory configuration**: Alignment and page size choices
Impact of QPI snoop mode and CoD on latency

- Starting with HSW, QPI snoop mode can be set via BIOS
  - Early Snoop
  - Home Snoop
  - Home Snoop + Opportunistic Snoop Broadcast (BDW only)
  - Directory (CoD)

<table>
<thead>
<tr>
<th></th>
<th>SNB</th>
<th>IVB</th>
<th>HSW</th>
<th>BDW</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>L2</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>L3 (COD)</td>
<td>40</td>
<td>40</td>
<td>37</td>
<td>41 (CoD)</td>
</tr>
<tr>
<td>Mem (COD)</td>
<td>230</td>
<td>208</td>
<td>168</td>
<td>280 (HS), 248 (ES), 190 (HS+OSB), 176 (DIR)</td>
</tr>
</tbody>
</table>

Graph 500 (v2.1.4), full chip w/ SMT, Turbo
Xeon E5-2697v4 (BDW) μarch comparison

Cache/Memory Latency [cycles]
Uncore Frequency: Bandwidth and Energy Consumption

Intel HPCG (16.0.3), n=256, full chip (no SMT), Xeon E5-2697 v4 (BDW)

Core frequency: 2.3 GHz

Performance [GFlop/s]

Performance/Watt [MFlop/s/W]

Performance/Watt [GFlop/s/W]

Core Frequency [GHz]

Uncore Frequency [GHz]

Performance per Watt [MFlop/s/W]

Varying core frequency

Core Frequency [GFlop/s]

Core Frequency [GFlop/s/W]

Core Frequency [GFlop/s/W]

Varying core frequency

Core Frequency [GFlop/s]

Core Frequency [GFlop/s/W]

Core Frequency [GFlop/s/W]

Varying core frequency

Core Frequency [GFlop/s]

Core Frequency [GFlop/s/W]

Core Frequency [GFlop/s/W]

Varying core frequency

Core Frequency [GFlop/s]

Core Frequency [GFlop/s/W]

Core Frequency [GFlop/s/W]

Varying core frequency

Core Frequency [GFlop/s]

Core Frequency [GFlop/s/W]

Core Frequency [GFlop/s/W]

Varying core frequency

Core Frequency [GFlop/s]

Core Frequency [GFlop/s/W]

Core Frequency [GFlop/s/W]

Varying core frequency

Core Frequency [GFlop/s]

Core Frequency [GFlop/s/W]

Core Frequency [GFlop/s/W]
Uncore Frequency: LINPACK Performance

Intel HPL (16.0.3), N=60,000, full chip (no SMT), Xeon E5-2697 v4 (BDW)

Varying core frequency

Core frequency: Turbo mode

Performance [GFlop/s]

Frequency [GHz]
Specific issues with Xeon Phi

- **MCDRAM** adds additional complexity

- **Configuration** of system and **mapping** of application on hardware gets more critical

- The compromise and made with KNL will soon be outdated

- KNL as a hosted cluster system is probably too specialized for a general purpose academic cluster
But

- Xeon Phi implements features which are not available anywhere else:
  - High degree of chip level parallelism
  - Multiple memory types and explicit memory control
  - Mesh type on-die topology

- It allowed a glimpse in the future on real hardware