A Semi-Automated Tool Flow for Roofline Analysis of OpenCL Kernels on Accelerators

Serveshe Muralidharan, ICHEC
Kenneth O’Brien, Xilinx
Christian Lalanne, ICHEC

Presented By
Gilles Civario, ICHEC
Motivation

• Comparing a diverse set of OpenCL supported platforms on a common set of metrics is a non-trivial problem
• Optimizations performed on one platform may or may not lead to optimal performance on another
• Lack of a tool that compares device capabilities and OpenCL kernel performance
Semi-Automated Tool Flow Design

- Complete automation is difficult to impossible due to the variety of tools and platforms
- Staged approach to eliminate redundant steps
- Device analysis performed once on each platform
- OpenCL kernel analysis repeated for each application version
OpenCL Accelerators Compared

<table>
<thead>
<tr>
<th>Device</th>
<th>( \overline{F}_f )</th>
<th>( \overline{F}_i )</th>
<th>( B )</th>
<th>( \overline{P} )</th>
<th>( \overline{F}'_f )</th>
<th>( \overline{F}'_i )</th>
<th>( B )</th>
<th>( \overline{P}' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tesla K20</td>
<td>3524</td>
<td>587</td>
<td>208</td>
<td>225</td>
<td>2903</td>
<td>585</td>
<td>143</td>
<td>225</td>
</tr>
<tr>
<td>Phi 5110P</td>
<td>1988</td>
<td>1988</td>
<td>320</td>
<td>245</td>
<td>1189</td>
<td>946</td>
<td>119</td>
<td>245</td>
</tr>
<tr>
<td>ADM 7V3</td>
<td>738</td>
<td>8880</td>
<td>21</td>
<td>25</td>
<td>200(\frac{3}{2})</td>
<td>3032(\frac{3}{2})</td>
<td>8.5(\frac{1}{2})</td>
<td>25</td>
</tr>
</tbody>
</table>

- Measured Peak is better for comparisons but in some cases estimations are necessary
- Xeon Phi has the best measured peak integer based performance
- Tesla K20 has the best measured peak floating point performance
- ADM 7V3 has the lowest peak power consumption and estimated non-floating point performance

ADM 7V3 peak integer performance is estimated using 70% of \( \#\text{LUTS}/20 \) *200Mhz (operating frequency of the FPGA), which is 0.7*(433200/20)*200 = 3032.4 OPS/s.

Remaining LUTs comprise infrastructure surrounding kernel.
Device Rooflines

Performance Roofline

- \( F \times 10^9 \text{ OPS/Second} \)

Operational Intensity \( I \) (OPS/Byte)

- \( \times 10^9 \text{ OPS/Second/Watt} \)

Operational Intensity \( I \) (OPS/Byte)

- Nvidia Tesla K20
- Intel Xeon Phi 5110P
- AlphaData FPGA 7V3

Performance Per Watt Roofline

- Represents non floating point performance
- Represents floating point performance
Tool Flow

- Iterative approach
- Analysis feedbacks to optimizations
Evaluation

**ALGORITHM 1:** Bob Jenkins `lookup3` hash function

```plaintext
key ← Input string to hash
length ← Length of input string
init ← Initialization value of the hash
hash ← Returns the hash value

begin
    a, b, c ← Initialize based on length and init
    index ← Index of the key
    while length > 12 do
        // mixing
        a += key[ index + 0 ]
        b += key[ index + 1 ]
        c += key[ index + 2 ]
        Mix ( a, b, c )
        length -= 12
        index -= 3
    end

    /* Mix the remainder in a, b, c */
    /* and return the hash */
    return MixRemainder ( a, b, c, key, length )
end
```

Instructions executed for kernel 'hash':
- 242,802,730 - add
- 175,687,220 - phi
- 166,657,332 - xor
- 158,268,724 - sub
- 158,268,724 - shl
- 158,268,724 - lshr
- 158,268,724 - or
- 100,506,735 - getelementptr
- 83,568,763 - load global (334,275,052 bytes)
- 66,951,596 - br
- 33,556,176 - icmp
- 25,165,824 - mul
- 16,777,216 - zext
- 14,559,207 - and
- 8,388,608 - udiv
- 8,388,608 - trunc
- 8,388,608 - switch
- 8,388,608 - select
- 8,388,608 - ret
- 8,388,608 - store global (33,554,432 bytes)
- 8,388,608 - call get_global_id()

**\( W = 1224 \text{ Million OPS} \)**

**\( Q = 367 \text{ Million bytes} \)**

**\( I = 3.33 \text{ OPS/Byte} \)**
Results – Intel Xeon Phi 5110P

- Optimal implementation of the function is memory bound on the Xeon Phi
- $66.70 \times 10^9$ OPS/second
- $0.38 \times 10^9$ OPS/second/Watt
- Performance limitation due to the inability to use vector processing units of the Phi due to the inherent feedback loop and branch divergence
Results – Nvidia Tesla K20

• Optimal implementation of the function is not as badly memory bound in comparison to Xeon Phi
• $126.42 \times 10^9$ OPS/second
• $1.18 \times 10^9$ OPS/second/Watt
• Possible performance limitation due to branch divergence
Results – Alpha Data ADM-PCIE-7V3

• Optimal implementation is heavily memory bound much worse than the Xeon Phi
• $18.11 \times 10^9$ OPS/second
• $1.02 \times 10^9$ OPS/second/Watt
• Improvements to memory controller efficiency and number of memory channels on the platform can increase performance
Conclusion

• Semi-automated tool that can benchmark, measure and evaluate implementations of an algorithm across different OpenCL accelerators.

• Performance per Watt extension to roofline models presents insight into the peak energy efficiency

• Methodology to present experimental results on otherwise theoretical roofline models

• Currently investigating a diverse range of OpenCL applications that reflect a wide range of operational intensities.