Benchmarking AI-methods on Heterogeneous Hardware Resources

Christopher Hesse, Holger Eichelberger

{eichelberger}@sse.uni-hildesheim.de,
hessech@uni-hildesheim.de

Software Systems Engineering
University of Hildesheim

www.sse.uni-hildesheim.de
Motivation

• Artificial Intelligence (AI) is “everywhere”
• HAISEM-Lab (http://haisem-lab.de/)
  – BMBF founded AI lab
  – Hardware-optimized Artificial Intelligence Applications using modern Software Engineering Methods
  – Qualification and training for industry personnel
  – AI/Hardware/SE research
• Partners
  – University of Hannover (L3S, IMS)
  – University of Hildesheim (SSE)
Problem

- Hardware for AI
  - GPU server 8 NVIDIA Tesla
  - FPGA server with 2 Maxeler Maia cards
  - TPU/GPU developer boards, GPU laptops

*How to compare AI performance (speed, energy) across all heterogeneous hardware resources?*

- Existing approaches: At least one hardware type missing
Approach

• Focus: Convolutional Neural Networks
  – "What's in this image?"

• Bottom-Up
  – Microbenchmarks
    • Convolution
    • Pooling
  – Macrobenchmarks
    • Training
    • Inference

• Methodology
  – Varying input/filter size, e.g., 100x100, 1000x1000, ...
  – Run each benchmark for \( n \) seconds
  – Measure per iteration / benchmark
    • CPU/GPU time
    • Energy
### Preliminary results

#### 2D Convolution

<table>
<thead>
<tr>
<th></th>
<th>Laptop (CPU)</th>
<th>Laptop (GPU)</th>
<th>Server (CPU)</th>
<th>Server (GPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speedup</td>
<td>1.0x</td>
<td>~ 10-20x</td>
<td>1.0x</td>
<td>~ 10-75x</td>
</tr>
<tr>
<td>GPU Power</td>
<td></td>
<td></td>
<td>45 W</td>
<td>65-295 W</td>
</tr>
<tr>
<td>Active Power</td>
<td></td>
<td></td>
<td>760-850 W</td>
<td>760-1150 W</td>
</tr>
</tbody>
</table>

#### CNN Training 32x32

<table>
<thead>
<tr>
<th></th>
<th>Laptop (CPU)</th>
<th>Laptop (GPU)</th>
<th>Server (CPU)</th>
<th>Server (GPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speedup</td>
<td>1.0x</td>
<td>~ 1.2x</td>
<td>1.0x</td>
<td>~ 1.6x</td>
</tr>
<tr>
<td>Memory</td>
<td>~ 4.5 GB</td>
<td>~ 5.6 GB</td>
<td>~ 5.0 GB</td>
<td>~ 8.3 GB</td>
</tr>
</tbody>
</table>

Input size too small? 224x224 scales much better.
Preliminary results

2D Convolution (DGX-1 Server, GPU)
• Width x Height x Channels

Discussion: 1000x1000-3x3x3

$t = 60$ seconds
$n = 320,000$
$\sigma = 1.8 \times 10^{-5}$

Outliers: ~2.4% (random)
Conclusions & Future Work

- Microbenchmarks: good scaling with hardware capability
- Macrobenchmarks: it depends ;)

- Compare with more/less GPUs
- Realize micro-benchmarks on FPGA
- Compare with existing benchmarks where possible
- Derive “best practice” tradeoffs

[Link to HASEM-Lab](http://haisem-lab.de/)