Toward Accelerating Deep Learning at Scale Using Specialized Hardware in the Datacenter

Kalin Ovtcharov, Olatunji Ruwase, Joo-Young Kim, Jeremy Fowers, Karin Strauss, Eric S. Chung

Acknowledgments: Doug Burger and the Catapult Team, Trishul Chilimbi and the Digital Cortex Team, Altera Corporation
The Rise of Deep Learning

- Significant advances in
  - Computer vision
  - Speech recognition
  - Natural language processing
  - Recommendation systems
  - Intelligent agents
  - Etc.

- Examples
  - Convolutional Neural Networks (CNNs)
  - Deep Belief Networks (DBNs)
  - Recurrent Neural Networks (RNNs)
  - ...?
This Talk: Are FPGAs a Promising Target in the Datacenter for Deep Learning$^1$?

$^1$Case study: CNN-based Image Classification (inference)
Cloud Specialization Tradeoffs

- Excellent maintainability in datacenter
- Maximum flexibility for all workloads
- Performance of CNNs/DNNs vastly slower than specialized HW
Cloud Specialization Tradeoffs

Homogeneous ➔ Heterogeneous

- CPU-Based Servers

- + CNNs/DNNs that utilize GPUs or ASICs benefit significantly
- - CNNs/DNNs cannot scale beyond limited pools
- - Heterogeneity challenging for maintainability

CPU-Based Servers + separate pools of GPUs or ASICs
Cloud Specialization Tradeoffs

Homogeneous - Increased cost and power per server (particularly GPUs) - Not economical for all applications in the datacenter (GPUs and ASICs)

Heterogeneous

CPU-Based Servers

Servers with GPUs or ASICs

CPU-Based Servers + separate pools of GPUs or ASICs

+ Homogeneous
Cloud Specialization Tradeoffs

Homogeneous → Heterogeneous

CPU-Based Servers

Servers with GPUs or ASICs

CPU-Based Servers + separate pools of GPUs or ASICs

Servers with FPGAs

+ Homogeneous
+ Low overhead in power and cost per server
+ Flexibility benefits many workloads?
- Lower peak performance than GPUs or ASICs on some workloads

→ Overtake through scale?
MICROSOFT SUPERCHARGES BING SEARCH WITH PROGRAMMABLE CHIPS

http://www.wired.com/2014/06/microsoft-fpga/
MICROSOFT SUPERCHARGES BING SEARCH WITH FPGA

2x Increase in Throughput
29% Latency Reduction
SW Only
SW + FPGA
< 30% Cost
< 25 W Power
0 HW Failures

http://www.wired.com/2014/06/microsoft-fpga/
Catapult FPGA Accelerator Card

- Altera Stratix V D5
- 172,600 ALMs, 2,014 M20Ks, 1,590 DSPs
- PCIe Gen 3 x8
- 8GB DDR3-1333
- Powered by PCIe slot
- Torus Network
Microsoft Open Compute Server

- Two 8-core Xeon 2.1 GHz CPUs
- 64 GB DRAM
- 4 HDDs @ 2 TB, 2 SSDs @ 512 GB
- 10 Gb Ethernet
- No cable attachments to server

Air flow
200 LFM
68 °C Inlet
Azure SmartNIC

- Use Catapult FPGAs for reconfigurable functions
  - Already used in Bing
  - Roll out Hardware as we do software

- Programmed using Generic Flow Tables (GFT)
  - Language for programming SDN to hardware
  - Uses connections and structured actions as primitives

- SmartNIC also does Crypto, QoS, storage acceleration, and more...
Harnessing Catapult for Deep CNNs

- Leverage abundant FPGA resources in the datacenter for scaling up evaluation and training\(^1\) of deep CNNs
- Achieve order-of-magnitude performance gain relative to CPUs with low cost (<30%) and power (<10%) overheads
- Expose to practitioners as composable SW libraries

\(^1\)Under development
Deep Convolutional Neural Networks

**INPUT**

- 3-D Convolution and Max Pooling

**Dense Layers**

- Max pooling
- Dense

**OUTPUT**

- “Dog”

* Krizhevsky et al, NIPS’12
3-D Convolution and Max Pooling

- **N**, **k**, **H**, and **p** may vary across layers

**Input Feature Map**
- $N$ = input height and width
- $k$ = kernel height and width
- $D$ = input depth

**Convolution Output**
- Convolution between $k \times k \times D$ kernel and region of Input Feature Map

**Max Pooled Output (Optional)**
- Max value over $p \times p$ region
- $H$ = # feature maps
- $S$ = kernel stride

* $N$, $k$, $H$, and $p$ may vary across layers*
3-D Convolution
3-D Convolution

Input  Kernel Weights  Output
3-D Convolution

Input  Kernel Weights  Output
3-D Convolution

Input  Kernel Weights  Output
3-D Convolution

Input

Kernel Weights

Output
3-D Convolution

Input

Kernel Weights

Output
3-D Convolution

Input

Kernel Weights

Output
3-D Convolution

Input

Kernel Weights

Output
3-D Convolution

Input

Kernel Weights

Output
3-D Convolution

Input | Kernel Weights | Output
3-D Convolution

Input  Kernel Weights  Output
3-D Convolution

Input  Kernel Weights  Output
CNN Accelerator Building Block

- **Configurable**
  - Numerical precision (static)
  - Number of layers
  - Layer dimensions
  - Stride and pooling

- **Scalable**
  - Can compose multiple engines together over Catapult network

- **Efficient**
  - Minimize memory bandwidth via data re-distribution NoC
  - On-chip per-row broadcast
Systolic Array Microarchitecture
DEMO
# ImageNet-1K Classification Performance

<table>
<thead>
<tr>
<th>Platform</th>
<th>Library/OS</th>
<th>ImageNet 1K Inference Throughput</th>
<th>Peak TFLOPs</th>
<th>Effective TFLOPs</th>
<th>Estimated Peak Power with Server</th>
<th>Estimated GOPs/J (assuming peak power)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-core, 2-socket Xeon E5-2450, 2.1GHz</td>
<td>Caffe + Intel MKL Ubuntu 14.04.1*</td>
<td>53 images/s</td>
<td>0.27T</td>
<td>0.074T (27%)</td>
<td>~225W</td>
<td>~0.3</td>
</tr>
<tr>
<td>Arria 10 GX1150</td>
<td>Windows Server 2012</td>
<td>369 images/s¹</td>
<td>1.366T</td>
<td>0.51T (38%)</td>
<td>~265W</td>
<td>~1.9</td>
</tr>
</tbody>
</table>

¹Dense layer time estimated
²https://github.com/soumith/convnet-benchmarks
## ImageNet-1K Classification Performance

<table>
<thead>
<tr>
<th>Platform</th>
<th>Library/OS</th>
<th>ImageNet 1K Inference Throughput</th>
<th>Peak TFLOPs</th>
<th>Effective TFLOPs</th>
<th>Estimated Peak Power with Server</th>
<th>Estimated GOPs/J (assuming peak power)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-core, 2-socket Xeon E5-2450, 2.1GHz</td>
<td>Caffe + Intel MKL Ubuntu 14.04.1*</td>
<td>53 images/s</td>
<td>0.27T</td>
<td>0.074T (27%)</td>
<td>~225W</td>
<td>~0.3</td>
</tr>
<tr>
<td>Arria 10 GX1150</td>
<td>Windows Server 2012</td>
<td>369 images/s¹</td>
<td>1.366T</td>
<td>0.51T (38%)</td>
<td>~265W</td>
<td>~1.9</td>
</tr>
<tr>
<td>NervanaSys-32 on NVIDIA Titan X</td>
<td>NervanaSys-32 on Ubuntu 14.0.4</td>
<td>4129 images/s²</td>
<td>6.1T</td>
<td>5.75T (94%)</td>
<td>~475W</td>
<td>~12.1</td>
</tr>
</tbody>
</table>

1Dense layer time estimated
2https://github.com/soumith/convnet-benchmarks

Includes server power; however, CPUs available to other jobs in the datacenter
## ImageNet-1K Classification Performance

<table>
<thead>
<tr>
<th>Platform</th>
<th>Library/OS</th>
<th>ImageNet 1K Inference Throughput</th>
<th>Peak TFLOPs</th>
<th>Effective TFLOPs</th>
<th>Estimated Peak Power for CNN Computation</th>
<th>Estimated GOPs/J (assuming peak power)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-core, 2-socket Xeon E5-2450, 2.1GHz</td>
<td>Caffe + Intel MKL Ubuntu 14.04.1*</td>
<td>53 images/s</td>
<td>0.27T</td>
<td>0.074T (27%)</td>
<td>~225W</td>
<td>~0.3</td>
</tr>
<tr>
<td>Arria 10 GX1150</td>
<td>Windows Server 2012</td>
<td>369 images/s(^1)</td>
<td>1.366T</td>
<td>0.51T (38%)</td>
<td>~40W</td>
<td>~12.8</td>
</tr>
<tr>
<td>NervanaSys-32 on NVIDIA Titan X</td>
<td>NervanaSys-32 on Ubuntu 14.0.4</td>
<td>4129 images/s(^2)</td>
<td>6.1T</td>
<td>5.75T (94%)</td>
<td>~250W</td>
<td>~23.0</td>
</tr>
</tbody>
</table>

\(^1\) Dense layer time estimated  
\(^2\) https://github.com/soumith/convnet-benchmarks

**Under-utilized FPGA vs. highly tuned GPU-friendly workload**
# Projected Improvements with Tuning

<table>
<thead>
<tr>
<th>Platform</th>
<th>Library/OS</th>
<th>ImageNet 1K Inference Throughput</th>
<th>Peak TFLOPs</th>
<th>Effective TFLOPs</th>
<th>Estimated Peak Power for CNN Computation</th>
<th>Estimated GOPs/J (assuming peak power)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>16-core, 2-socket Xeon E5-2450, 2.1GHz</strong></td>
<td>Caffe + Intel MKL Ubuntu 14.04.1*</td>
<td>53 images/s</td>
<td>0.27T</td>
<td>0.074T (27%)</td>
<td>~225W</td>
<td>~0.3</td>
</tr>
<tr>
<td><strong>Arria 10 GX1150</strong></td>
<td>Windows Server 2012</td>
<td>369 images/s(^1) ~880 images/s</td>
<td>1.366T</td>
<td>0.51T (38%)</td>
<td>~40W</td>
<td>20.6 (\approx 30.6)</td>
</tr>
<tr>
<td><strong>NervanaSys-32 on NVIDIA Titan X</strong></td>
<td>NervanaSys-32 on Ubuntu 14.0.4</td>
<td>4129 images/s(^2)</td>
<td>6.1T</td>
<td>5.75T (94%)</td>
<td>~250W</td>
<td>~23.0</td>
</tr>
</tbody>
</table>

\(^1\) Dense layer time estimated
\(^2\) [https://github.com/soumith/convnet-benchmarks](https://github.com/soumith/convnet-benchmarks)

Projected Results Assuming Floorplanning and Scaling Up PEs
Are FPGAs a Promising Target in the Datacenter for Deep Learning? **Yes.**

- Best-case FPGA design is ~1/5th GPU throughput but can overtake at scale

- Although CNNs are ideal on GPUs, FPGAs with hardened FPUs can achieve GPU-like energy efficiency

- FPGA is 7X faster (~16X within reach) than multicore CPUs while flexible enough for diverse cloud scenarios (Bing Ranking, Azure SmartNIC)
Related Work

• ASICs
  • [Holler’90], [Chen’14], [Cavigelli’15], etc.

• FPGAs
  • [LeCun’09], [Farabet’10], [Aysegul’13], [Baidu’14], [Gokhale’15], [Zhang’15], etc.

• GPUs, Appliances
  • Nervana, Nvidia DIGITS, Ersatz, etc.
Thank you!
erchung@microsoft.com