Panel – The Impact of AI Workloads on Datacenter Compute and Memory

Marc Tremblay
Microsoft
Distinguished Engineer, Azure H/W
Responsible for silicon/systems roadmap for AI
Full stack expert from application requirement to silicon
Previously CTO of Microelectronics @ Sun

Sumit Gupta
IBM
Vice President AI, ML and HPC business
Responsible for strategy and HW/SW products for Watson ML accelerator and Spectrum compute
Previously GM AI & GPU data center business, Nvidia

Cliff Young
Google
Software Engineer Google Brain team
HW/SW Codesign and Machine Learning
One of the TPU designers
Previously at D.E. Shaw and Bell Labs

Maxim Naumov
Facebook
Research Scientist Facebook AI Research
Deep Learning, Parallel Algorithms and Systems
Co-developed many of Nvidia GPU-accelerated libraries
Previously at Nvidia Research and Intel

Greg Diamos
Baidu
AI Research Lead Silicon Valley AI Labs
Co-developed Deep Speech and Deep Voice
Contributor to Volta SIMT scheduler, Compiler and Microarchitecture
Previously at Nvidia

Rob Ober
Nvidia
Chief Platform Architect Datacenter products
GPU deployments for AI and DL @ Hyperscalers
CPU architecture, storage systems, SSD, N/W, wireless and pwr mgmt.
Previously at Sandisk, LSI, Apple etc.

Anand Iyer
Samsung
Dir, Planning/Technology, Semiconductor products
Data center and AI HBM/Accelerators/CPU
CPU, Network infra, near memory compute
Previously at Broadcom, Cavium and Digital

---

Samsung @ The Heart of Your Data
Unparalleled Product Breadth & Technology Leadership
Data is being created faster than our ability to make sense of it.
ML/DL Systems – Market Demand and Key Application Domains

DL inference in data centers [1]

Vision

Recommenders

NMT

[1] “Deep Learning Inference in Data Centers: Characterization, Performance Optimizations and Hardware Implications”, ArXiv, 2018
https://code.fb.com/ml-applications/expanding-automatic-machine-translation-to-more-languages
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RecSys</td>
<td>FCs</td>
<td>1-10M</td>
<td>&gt; 10K</td>
<td>20-200</td>
<td>20-200</td>
</tr>
<tr>
<td></td>
<td>Embeddings</td>
<td>&gt;10 Billion</td>
<td>&gt; 10K</td>
<td>1-2</td>
<td>1-2</td>
</tr>
<tr>
<td>CV</td>
<td>ResNeXt101-32x4-48</td>
<td>43-829M</td>
<td>2-29M</td>
<td>Avg. 380/Min. 100</td>
<td>Avg. 188/Min. 28</td>
</tr>
<tr>
<td></td>
<td>Faster-RCNN-ShuffleNet</td>
<td>6M</td>
<td>13M</td>
<td>Avg.3.5K/Min. 2.5K</td>
<td>Avg. 145/Min. 4</td>
</tr>
<tr>
<td></td>
<td>ResNeXt3D-101</td>
<td>21M</td>
<td>58M</td>
<td>Avg. 22K/Min. 2K</td>
<td>Avg. 172/ Min. 6</td>
</tr>
<tr>
<td>NLP</td>
<td>seq2seq</td>
<td>100M-1B</td>
<td>&gt;100K</td>
<td>2-20</td>
<td>2-20</td>
</tr>
</tbody>
</table>
For the foreseeable future, off-chip memory bandwidth will often be the constraining resource in system performance.

System balance:
- Memory < Compute < Communication

Memory Access for:
- Network / program config and control flow
- Training data mini-batch compute flow

Compute consumes:
- Mini-batch data

Communication for:
- All reduce
- Embedding table insertion

Research + Development
- Time to train and accuracy
- Multiple runs for exploration, sometimes overnight

Production
- Optimal work/$
- Optimal work/watt
- Time to train
Hardware Trends

Time spent in caffe2 operators in data centers [1]

- High memory bandwidth and capacity for embeddings
- Support for powerful matrix and vector engines
- Large on-chip memory for inference with small batches
- Support for half-precision floating-point computation

Common activation and weight matrix shapes $(x_{MxK}W_{KxN})^T$ [1]

[1] “Deep Learning Inference in Data Centers: Characterization, Performance Optimizations and Hardware Implications”, ArXiv, 2018
Sample Workload Characterization

Embedding table hit rates and access histograms [2]

## Closed Division Speedups

<table>
<thead>
<tr>
<th>#</th>
<th>Submitter</th>
<th>Hardware</th>
<th>Chip count and type</th>
<th>Software</th>
<th>Benchmark results (speedup relative to reference implementation)</th>
<th>Cloud Scale</th>
<th>Power (unofficial, submitter-provided)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reference</td>
<td>Pascal P100</td>
<td>1 a Unoptimized reference</td>
<td>ImageNet COCO</td>
<td>1.0 1.0 1.0 1.0 1.0 1.0 1.0</td>
<td>1.0/a</td>
<td>none</td>
</tr>
<tr>
<td>2</td>
<td>Google</td>
<td>TPUv2.8</td>
<td>4 a TF 1.12</td>
<td>COCO COCO WMT E-G WMT E-G MovieLens-20M</td>
<td>29.3 8.5 28.1 781.5 171.6</td>
<td>2.6/a</td>
<td>none</td>
</tr>
<tr>
<td>3</td>
<td>TPUv2.512 + TPUv2.8</td>
<td>200 a TF 1.12</td>
<td>781.5 171.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Google</td>
<td>TPUv3.8</td>
<td>4 a TF 1.12</td>
<td>COCO COCO WMT E-G WMT E-G MovieLens-20M</td>
<td>48.2 11.1 43.1 4.2</td>
<td>4.2/a</td>
<td>none</td>
</tr>
<tr>
<td>5</td>
<td>8x Volta V100</td>
<td>8 a TF 1.12, cuDNN 7.4</td>
<td>64.1 11.4</td>
<td>NMT Transformer NCF Mini Go</td>
<td>64.1 11.4</td>
<td>11.4/a</td>
<td>none</td>
</tr>
</tbody>
</table>

### Available in cloud

<table>
<thead>
<tr>
<th>#</th>
<th>Submitter</th>
<th>Hardware</th>
<th>Chip count and type</th>
<th>Software</th>
<th>Benchmark results (speedup relative to reference implementation)</th>
<th>Cloud Scale</th>
<th>Power (unofficial, submitter-provided)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Intel</td>
<td>1x 2S SKX8180</td>
<td>2 c Intel Caffe 1.12a</td>
<td>ImageNet COCO COCO</td>
<td>0.85 0.5</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Intel</td>
<td>8x 2S SKX8180</td>
<td>16 c Intel Caffe 1.12a</td>
<td>COCO COCO</td>
<td>6.7</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Intel</td>
<td>4x 4S SKX8180</td>
<td>16 c Intel Caffe 1.12a</td>
<td>COCO COCO</td>
<td>6.6</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Intel</td>
<td>1x 2S SKX8180</td>
<td>2 c BigDL 0.7.0</td>
<td>COCO COCO</td>
<td>1.6</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Intel</td>
<td>1x 2S SKX8180</td>
<td>2 c TensorFlow 1.10.1</td>
<td>COCO COCO</td>
<td>6.3</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Intel</td>
<td>1x 4S SKX8180</td>
<td>4 c TensorFlow 1.10.1</td>
<td>COCO COCO</td>
<td>9.9</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>NVIDIA</td>
<td>DGX-1</td>
<td>8 a ngc18.11_MXNet, cuDNN 7.4</td>
<td>ImageNet COCO COCO</td>
<td>65.6</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>NVIDIA</td>
<td>DGX-1</td>
<td>8 a ngc18.11_pyTorch, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>30.8 15.5 62.0 57.2 93.4</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>NVIDIA</td>
<td>DGX-1</td>
<td>8 a ngc18.11_pyTorch, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>127.3 61.7</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>NVIDIA</td>
<td>DGX-1</td>
<td>192 a ngc18.11_pyTorch, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>301.6</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>NVIDIA</td>
<td>DGX-1</td>
<td>192 a ngc18.11_pyTorch, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>405.2</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>NVIDIA</td>
<td>DGX-1</td>
<td>640 a ngc18.11_MXNet, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>1,424.4</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>NVIDIA</td>
<td>DGX-2</td>
<td>16 a ngc18.11_MXNet, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>119.5</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>NVIDIA</td>
<td>DGX-2</td>
<td>16 a ngc18.11_pyTorch, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>52.1 28.4 108.0 86.2 116.8</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>NVIDIA</td>
<td>DGX-2</td>
<td>16 a ngc18.11_MXNet, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>126.2</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>NVIDIA</td>
<td>DGX-2</td>
<td>16 a ngc18.11_pyTorch, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>58.7 30.0 115.6 97.4 116.8</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>NVIDIA</td>
<td>DGX-2</td>
<td>16 a ngc18.11_MXNet, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>49.3</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>NVIDIA</td>
<td>DGX-2</td>
<td>8x DGX-2 with 8 V100s</td>
<td>COCO COCO</td>
<td>147.8</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>NVIDIA</td>
<td>DGX-2</td>
<td>256 a ngc18.11_pyTorch, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>420.2</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>NVIDIA</td>
<td>DGX-2</td>
<td>512 a ngc18.11_MXNet, cuDNN 7.4</td>
<td>COCO COCO</td>
<td>1,193.4</td>
<td>n/a none</td>
<td></td>
</tr>
</tbody>
</table>

### Research

<table>
<thead>
<tr>
<th>#</th>
<th>Submitter</th>
<th>Hardware</th>
<th>Chip count and type</th>
<th>Software</th>
<th>Benchmark results (speedup relative to reference implementation)</th>
<th>Cloud Scale</th>
<th>Power (unofficial, submitter-provided)</th>
</tr>
</thead>
<tbody>
<tr>
<td>26</td>
<td>Google</td>
<td>TPUv3.32 + TPUv2.8</td>
<td>20 a TF 1.12</td>
<td>COCO COCO</td>
<td>147.4 46.5 117.0</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Google</td>
<td>TPUv3.32 + TPUv2.8</td>
<td>200 a TF 1.12</td>
<td>COCO COCO</td>
<td>1,243.8</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Intel</td>
<td>1x 2S SKX8180</td>
<td>2 c custom TensorFlow 1.10.1</td>
<td>COCO COCO</td>
<td>6.9</td>
<td>n/a none</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Intel</td>
<td>1x 4S SKX8180</td>
<td>4 c custom TensorFlow 1.10.1</td>
<td>COCO COCO</td>
<td>12.9</td>
<td>n/a none</td>
<td></td>
</tr>
</tbody>
</table>
The Move to The Edge

By 2022, 7 out of every 10 bytes of data created will never see a data center.

Considerations
- Compute closer to Data
- Smarter Data Movement
- Faster Time to Insight

Let Data Speak for Itself!
Cloud AI
- 6x faster training time
- 8x training cost effectiveness

Edge computing AI
- 2x faster data access
- 2x hot data feeding

On-device AI
- 1.5x bandwidth
- privacy & fast response

Samsung @ The Heart of Your Data
Visit us at Booth #726