Hardware for Deep Learning

Hot Chips
August 29, 2023

Bill Dally
Chief Scientist and SVP of Research, NVIDIA Corporation
Adjunct Professor of CS and EE, Stanford
Q: What would Bill Dally say about directions in deep learning hardware

Chat GPT: Bill Dally has discussed several directions in deep learning hardware that he believes are important for the future of the field. Some of these directions include:

1. Specialized hardware for specific deep learning tasks: Dally has argued that as deep learning becomes more complex and diverse, specialized hardware tailored to specific tasks, such as natural language processing or computer vision, will become increasingly important.

2. Energy efficiency: As deep learning systems grow in size and complexity, their energy requirements can become prohibitively high. Dally has emphasized the need for energy-efficient hardware designs that can handle the demands of deep learning workloads.

3. Hardware-software co-design: Dally believes that hardware and software must be designed together to achieve optimal performance. This includes optimizing the hardware architecture for specific deep learning algorithms and developing software tools that can take advantage of the hardware’s unique features.

4. Low-precision arithmetic: Dally has proposed using low-precision arithmetic, such as 8-bit or 4-bit arithmetic, in deep learning hardware to reduce energy consumption while still maintaining high accuracy.

5. Memory hierarchy optimization: Dally has also stressed the importance of optimizing the memory hierarchy in deep learning hardware to reduce energy consumption and improve performance.

In summary, Bill Dally believes that deep learning hardware must be tailored to the specific needs of different tasks, be energy-efficient, use low-precision arithmetic, optimize the memory hierarchy, and be co-designed with software to achieve optimal performance.
10^{12} Tokens  
10^9 Images

Data

Model

Training

$10M-50M$ GPU Time

Trained Model

1K-1M

Fine Tuning

Special Data

Specialized Model

Application Dependent

Coding Medical Advice Education Writing Creative

Query

Inference

<1s GPU Time

Answer

~$3 x 10^{-4}$ per Word

1K-1M Annotated Queries

Human Feedback

100M-2T Parameters

$10M-50M$ GPU Time

$100K-1M$ GPU Time

©
Motivation
Deep Learning was Enabled by Hardware
Deep Learning is Gated by Hardware

Petaflop/s – Days Training


1.E+04

1.E+03

1.E+02

1.E+01

1.E+00

1.E-01

1.E-02

1.E-03

1.E-04

AlexNet

ResNet

Megatron-GPT2

Megatron-BERT

Turing NLG

GPT-2

GPT-3

GPT-4

est
Some History
Single-Chip Inference Performance - 1000X in 10 years

Int 8 TOPS

FP32 FMA
K20X 3.94

FP16 HDP4
M40 6.84
P100 21.20

FP16 HMMA
V100 125.00

A100 1248.00

Sparsity

H100 4000.00

INT8 IMMA
Q8000 261.00

HDP4
V100 125.00

FP16

FP32 FMA

H100 4000.00

V100
Gains from

- **Number Representation**
  - FP32, FP16, Int8
  - (TF32, BF16)
  - ~16x

- **Complex Instructions**
  - DP4, HMMA, IMMA
  - ~12.5x

- **Process**
  - 28nm, 16nm, 7nm, 5nm
  - ~2.5x

- **Sparsity**
  - ~2x

- Model efficiency has also improved – overall gain > 1000x
Specialized Instructions Amortize Overhead

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy**</th>
<th>Overhead*</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFMA</td>
<td>1.5pJ</td>
<td>2000%</td>
</tr>
<tr>
<td>HDP4A</td>
<td>6.0pJ</td>
<td>500%</td>
</tr>
<tr>
<td>HMMA</td>
<td>110pJ</td>
<td>22%</td>
</tr>
<tr>
<td>IMMA</td>
<td>160pJ</td>
<td>16%</td>
</tr>
</tbody>
</table>

*Overhead is instruction fetch, decode, and operand fetch – 30pJ
**Energy numbers from 45nm process
Hopper H100

1 PFLOPS (TF32)
1 / 2 PLFLOPS (FP16 or BF16) (dense/sparse)
2 / 4 PLFOPS (FP8 or Int8) (dense/sparse)

3.4TB/s (HBM3) 94GB
18 NVLINK ports
400Gb/s each 900GB/s total
700W

Transformer Engine
Dynamic Programming Instructions

9 TOPS/W (Int8/FP8)

4PF Sparse FP8, 900GB/s, 700W
3D Parallelism

It takes 20 GPUs to hold one copy of GPT4 model parameters.
DGX H100 Server

8-H100 4-NVSwitch Server
- 32 PFLOPS of AI Performance
- 640 GB aggregate GPU memory
- 18 NVLink Network OSFPs
- 3.6 TBps of full-duplex NVLink Network bandwidth (72 NVLinks)
- 8x 400 Gb/s ConnectX-7 InfiniBand/Ethernet ports
- 2 dual-port Bluefield-3 DPUs
- Dual Sapphire Rapids CPUs
- PCIe Gen5

32PF Sparse FP8, 11.3kW, 900GB/s
One Big GPU
DGX H100 Superpod: NVLink Switch

**NVLink Switch**
- Standard 1RU 19-inch formfactor highly leveraged from InfiniBand switch design
- Dual NVLink4 NVSwitch chips
- 128 NVLink4 ports
- 32 OSFP cages
- 6.4 TB/s full-duplex BW
- Managed switch with out-of-band management communication
- Support for passive-copper, active-copper and optical OSFP cables (custom FW)
Scale-up – NVLink and NVSwitch – to 256 GPUs
Scale-out – IB to 10,000s of GPUs
Collectives Double Effective Network Bandwidth (AllReduce)
Software
NVIDIA AI and H100 Deliver 6.7X in 2.5 Years

Full-stack innovation fuels continuous performance gains

MLPerf™ Training v2.1 Performance

Up to 6.7X

Higher performance with new H100 GPUs

Up to 2.5X

Speedup on existing A100 GPUs with software
NVIDIA H100 GPUs Set Standard for Generative AI in Debut MLPerf Benchmark

June 28, 2023

June 28, 2023 — Leading users and industry-standard benchmarks agree: NVIDIA H100 Tensor Core GPUs deliver the best AI performance, especially on the large language models (LLMs) powering generative AI.

H100 GPUs set new records on all eight tests in the latest MLPerf training benchmarks released this week, excelling on a new MLPerf test for generative AI. That excellence is delivered both per-accelerator and at-scale in massive servers.

Nvidia Hopper, Ampere GPUs Sweep MLPerf Benchmarks in AI Training

November 9, 2022

Nov. 9, 2022 — Two months after their debut sweeping MLPerf inference benchmarks, NVIDIA H100 Tensor Core GPUs set world records across enterprise AI workloads in the industry group’s latest tests of AI training.

Together, the results show H100 is the best choice for users who demand utmost performance when creating and deploying advanced AI models.

MLPerf Inference 3.0 Highlights — Nvidia, Intel, Qualcomm and…ChatGPT

By John Russell

April 5, 2023

MLCommons today released the latest MLPerf Inferencing (v3.0) results for the datacenter and edge. While Nvidia continues to dominate the results — topping all performance categories — other companies are joining the MLPerf

Nvidia Dominates MLPerf Inference, Qualcomm also Shines, Where’s Everybody Else?

By John Russell

April 6, 2022
Future Directions
Future Directions

Number representation
- Log numbers
- Vector scaling (VS-Quant)
- Optimal Clipping
- Much cheaper math
- Smaller numbers

Sparsity
- Activations
- Lower density (vs 2:4 in A100/H100)

Better tiling
- Lower memory energy

Circuits
- Memory
- Communication
- 3D memory

Process
- Capacitance scaling
Number Representation
- Attributes:
  - Cost
  - Operation energy
  - Movement energy
  - Accuracy
  - Dynamic range
  - Precision (error)

---

**int8**

- S
- M

**fp16**

- S
- E
- M

**log8**

- S
- E

**sym8**

- X

**spike**

- Waveform

**analog**

- Waveform

---

**Weight Buffer**

**Multiply Accumulate**

**Activation Buffer**

**Operation**

**Storage**

**Transport**
Symbol Representation (Codebook)

- **linear quantization**
- **nonlinear quantization by clustering and finetuning**

Graph showing the density of weight values with symbols indicating different quantization methods.
Log Representation
Dynamic Range $10^5$
WC Accuracy 4%

Vs Int8 – DR $10^2$
WC Accuracy 33%

$v = -1^s 2^{ei.ef}$

Log4.3  S  EI  EF
4-bit Log Representation (L2.2)

Max Error 9%

4-bit Integer Representation (Int4)

Max Error 33%
4-bit Log Representation (L2.2)

Closest Representable Value

Actual Value

Max Error 9%

FP2.2

Closest Representable Value

Actual Value

Max Error 13%
• Log Numbers

• Multiplies are cheap – just an add

• Adds are hard – convert to integer, add, convert back
  • Fractional part of log is a lookup
  • Integer part of log is a shift

• Can factor the lookup outside the summation
  • Only convert back after summation (and NLF)
Optimum Clipping
Whatever number representation you use
Pick the range optimally
large q. noise

low density data

Tensor PDF

Value

-0.8  -0.6  -0.4  -0.2  0.0  0.2  0.4  0.6  0.8

Tensor PDF

Value

-0.8  -0.6  -0.4  -0.2  0.0  0.2  0.4  0.6  0.8
\[ J = \frac{4-B}{3} s^2 \int_0^s f_X(x)dx + \int_s^{\infty} (s-x)^2 f_X(x)dx \]

\[ S_{n+1} = \frac{E[|X| \cdot 1_{\{|X| > s_n\}}]}{4-B} \frac{3}{3} E[1_{\{|X| < s_n\}}] + E[1_{\{|X| > s_n\}}] \]
Vector Scaling
VS-Quant
Per-vector scaled quantization for low-precision inference

\[ y_q(j) = \left( \sum_{i=0}^{\text{vecsize}-1} w_q(i) a_q(i) \right) s_w(j) s_a(j) \]

Works with either post-training quantization or quantization-aware retraining!

[Dai et al., MLSYS 2021]

**Fine-grained scale factors per vector**

**Modified vector MAC unit for VS-Quant**
INT4 Quantization

Traditional Quantization

Min value in matrix
Max value in matrix
Scaling

More scaling

FP32 data distribution

Noise

More scaling

VSQ Scale Factors

One scale factor for each 64-element input vector

Second scale factor for each input matrix

Traditional Quantization

<table>
<thead>
<tr>
<th>vsq</th>
<th>traditional quantization</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSQ</td>
<td>Two scale factors: one per vector, one per matrix</td>
</tr>
<tr>
<td>N</td>
<td>Reduced quantization noise</td>
</tr>
<tr>
<td>M</td>
<td>High quantization noise</td>
</tr>
<tr>
<td>K</td>
<td>One scale factor per matrix</td>
</tr>
</tbody>
</table>
Sparsity
Pruning

before pruning

after pruning

pruning synapses

pruning neurons
Structured Sparsity


NVIDIA A100 Tensor Core GPU Architecture whitepaper
Accelerators
Accelerators Employ:

• Special **Data Types** and **Operations**
  • Do in 1 cycle what normally takes 10s or 100s – **10-1000x efficiency gain**

• Massive **Parallelism** – >1,000x, not 16x – with **Locality**
  • This gives performance, not efficiency

• Optimized **Memory**
  • High bandwidth (**and low energy**) for specific data structures and operations

• Reduced or Amortized **Overhead**
  • **10,000x efficiency gain** for simple operations

• Algorithm-Architecture **Co-Design**
Fast Accelerators since 1985


- **Darwin**: Turakhia, Bejerano, and Dally, “Darwin: A Genomics Co-processor provides up to 15,000× acceleration on long read assembly”, ASPLOS 2018.

- **SATiN**: Zhuo, Rucker, Wang, and Dally, “Hardware for Boolean Satisfiability Inference,”
Eliminating Instruction Overhead

Area is proportional to energy – all 28nm

16b Int Add, 32fJ


### Cost of Operations

**Energy (pJ) Numbers** are from Mark Horowitz “Computing's Energy Problem (and what we can do about it)”, ISSCC 2014.

**Area (µm²) Numbers** are from synthesized result using Design Compiler under TSMC 45nm tech node. FP units used DesignWare Library.

<table>
<thead>
<tr>
<th>Operation:</th>
<th>Energy (pJ)</th>
<th>Area (µm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8b Add</td>
<td>0.03</td>
<td>36</td>
</tr>
<tr>
<td>16b Add</td>
<td>0.05</td>
<td>67</td>
</tr>
<tr>
<td>32b Add</td>
<td>0.1</td>
<td>137</td>
</tr>
<tr>
<td>16b FP Add</td>
<td>0.4</td>
<td>1360</td>
</tr>
<tr>
<td>32b FP Add</td>
<td>0.9</td>
<td>4184</td>
</tr>
<tr>
<td>8b Mult</td>
<td>0.2</td>
<td>282</td>
</tr>
<tr>
<td>32b Mult</td>
<td>3.1</td>
<td>3495</td>
</tr>
<tr>
<td>16b FP Mult</td>
<td>1.1</td>
<td>1640</td>
</tr>
<tr>
<td>32b FP Mult</td>
<td>3.7</td>
<td>7700</td>
</tr>
<tr>
<td>32b SRAM Read (8KB)</td>
<td>5</td>
<td>N/A</td>
</tr>
<tr>
<td>32b DRAM Read</td>
<td>640</td>
<td>N/A</td>
</tr>
</tbody>
</table>

---

Energy numbers are from Mark Horowitz “Computing’s Energy Problem (and what we can do about it)”, ISSCC 2014.
Area numbers are from synthesized result using Design Compiler under TSMC 45nm tech node. FP units used DesignWare Library.
The Importance of Staying Local

LPDDR DRAM
GB
640pJ/word

On-Chip SRAM
MB
50pJ/word

Local SRAM
KB
5pJ/word
Magnet
Configurable using synthesizable SystemC, HW generated using HLS tools

[Venkatesan et al., ICCAD 2019]
Energy-efficient DL Inference accelerator
Transformers, VS-Quant INT4, TSMC 5nm

- Efficient architecture
  - Used MAGNet [Venkatesan et al., ICCAD 2019] to design a low-precision DL inference accelerator for Transformers
  - Multi-level dataflow to improve data reuse and energy efficiency

- Low-precision data format: VS-Quant INT4
  - Hardware-software techniques to tolerate quantization error
  - Enable low cost multiply-accumulate (MAC) operations
  - Reduce storage and data movement

- Special function units

- TSMC 5nm
- 1024 4-bit MACs/cycle (512 8-bit)
- 0.153 mm² chip
- Voltage range: 0.46V – 1.05V
- Frequency range: 152 MHz – 1760 MHz

- 95.6 TOPS/W with 50%-dense 4-bit input matrices with VSQ enabled at 0.46V
- 0.8% energy overhead from VSQ support with 50%-dense inputs at 0.67V

[Keller, Venkatesan, et al., “A 95.6-TOPS/W Deep Learning Inference Accelerator with Per-Vector Scaled 4-bit Quantization in 5nm”, JSSC 2023]
Conclusion
Conclusion

- Deep Learning was enabled by hardware and its progress is limited by hardware
  - 1000x in last 10 years
    - Number representation, complex ops, sparsity
- Logarithmic numbers
  - Lowest worst-case error for a given number of bits
  - Can ‘factor out’ hard parts of an add
- Optimum clipping
  - Minimize MSE by trading quantization noise for clipping noise
- VS-Quant
  - Separate scale factor for each small vector – 16 to 64 scalars
- Accelerators – Testbeds for GPU ‘cores’
  - Test chip validates concepts and measures efficiency
  - 95.6 TOPS/W on BERT with negligible accuracy loss