INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

A GENTLE INTRODUCTION TO HARDWARE ARCHITECTURES FOR DNN ACCELERATION

Vaader Reading Group, 06/05/21

Mickaël Dardaillon - mdardail@insa-rennes.fr



Menu du jour

DNN layers Architecture support

GPUs SIMT Machines

Systolic Arrays Domain-Specific Architectures

DNN Optimizations

for hardware acceleration

Chapter 4: Data-Level Parallelism Chapter 7: Domain-Specific Architectures

Sixth Edition

John L. Hennessy | David A. Patterson

COMPUTER Architecture







DNN LAYERS

Architecture support





Arithmetic Intensity

Arithmetic intensity is the ratio of floating-point operations per byte of memory accessed.

It is computed by taking the total number of operations for a program and dividing it by the number of data bytes transferred during program execution.



J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 2018.



Roofline model

A visual performance model to find bound and bottlenecks.

The horizontal line shows the peak computational performance, the diagonal is the peak memory performance. Figure 3. The rooflines of TPUs, CPUs, and GPUs combined into a single log-log graph. Stars are for the TPU, triangles for the K80, and circles for Haswell; all TPU stars are at or above the other two rooflines.





Operational Intensity: MAC Ops/weight byte (log scale)

An operational intensity is a vertical line, which cross the roofline at its peak performance.

N. P. Jouppi, C. Young, N. Patil, and D. Patterson, "A domain-specific architecture for deep neural networks," Commun. ACM, vol. 61, no. 9, pp. 50–59, 2018.









Equivalent to Matrix Multiplication

Virtual memory duplication



S. Chetlur et al., "cuDNN: Efficient Primitives for Deep Learning"



Convolutional Layer







Forget DNN, let's implement Matrix Multiplication!

"Nvidia Deep Learning Performance Documentation"



GPUS

SIMT Machines





Single Instruction Multiple Threads (SIMT)

```
SIMT C = A + B
kernel(){
   tid = blkDim * blkId + thId
   C[tid] = A[tid] + B[tid]
}
```



J. Gómez Luna and O. Mutlu, "Computer Architecture: GPU Programming," ETH Zurich, Fall 2020.



Latency hiding

2 active Warps



4 active Warps



J. Gómez Luna and O. Mutlu, "Computer Architecture: GPU Programming," ETH Zurich, Fall 2020.



Data layout

Data layout has an effect on performance Shared memory is interleaved (banked) Typically 32 banks on Nvidia GPUs



"Nvidia Deep Learning Performance Documentation"



Naïve implementation





Tiled implementation





Tiled² implementation





Nvidia GPUs

Jetson Nano (Maxwell, 2014)

- 4 Streaming Multi-processors
- 32 CUDA cores / SM
- ~235 GFLOPS FP32
- 15 W

RTX 8000 (Turing, 2018)

- 72 Streaming Multi-processors
- 64 CUDA cores / SM
- 16,3 TFLOPS FP32
- 295 W

RTX A6000 (Ampere, 2020)

- 84 Streaming Multi-processors
- 128 CUDA cores / SM
- 38,7 TFLOPS FP32
- 300 W

_									_								
		L0 Ir	nstruct	ion C	ache							L0 In	struct	ion C	ache		
Warp Scheduler (32 thread/clk)								Warp Scheduler (32 thread/clk)									
	DI	spatci	runit	32 th	read/cl	к)					וט	spatch	ronit (32 th	read/d	:тк)	
	Reç	jister	File (1	6,384	4 x 32-	bit)					Reg	ister	File (1	6,38	4 x 32	2-bit)	
INT32 INT32	FP32	FP32	FP	64					INT32	INT32	FP32	FP32	FP	64			
INT32 INT32	FP32	FP32	FP	64					INT32	INT32	FP32	FP32	FP	64			
INT32 INT32	FP32	FP32	FP	64					INT32	INT32	FP32	FP32	FP	64			
INT32 INT32	FP32	FP32	FP	64	TEI	NSO	RCORE		INT32	INT32	FP32	FP32	FP	64	TF		R COR
INT32 INT32	FP32	FP32	FP	64					INT32	INT32	FP32	FP32	FP	64			
INT32 INT32	FP32	FP32	FP	64					INT32	INT32	FP32	FP32	FP	64			
INT32 INT32	FP32	FP32	FP	64					INT32	INT32	FP32	FP32	FP	64			
INT32 INT32	FP32	FP32	FP	64					INT32	INT32	FP32	FP32	FP	64			
LD/ LD/ ST ST	LD/ ST	LD/	LD/	LD/	LD/	LD/	CELL		LD/	LD/	LD/	LD/	LD/	LD/ ST	LD/	LD/	SEI
			51	ST	SI	ST	3F0		si							ST	SPC
			nstru <u>ct</u>	ST	ache_	ST	3F0	ון זר	SI	31		L0 In	struct	ion <u>C</u>	ache	ST	SPC
	Wa	L0 Ir	si nstruct iedul <u>e</u> i	sт ion С <mark>(32<u>t</u></mark>	ache hread <u>/c</u>	ST	3F0			31	Wa	L0 In	struct eduler	ion C (32 <u>t</u>	ache hread	ST /clk)_	SPC
	Wa	L0 Ir rp Sch spatch	si nstruct nedulei n Unit i	sт ion С (32 tl (32 th	ache hread/c read/cl	ST cik) k)	3F0			31	War	L0 In rp Sch spatch	struct eduler	ion C (32 t (32 th	ache hread	ST /clk) :lk)	SPC
	Wa Di Reg	L0 Ir rp Sch spatch jister	si Iedulei In Unit I File (1	sT ion C (32 th (32 th (32 sh	ache hread/c read/cl 4 x 32-	st ik) k) bit)				31	Wa Di Reg	L0 In rp Sch spatch jister I	eduler Unit (ion C (32 t (32 th 6,38	ache hread read/c 4 x 32	st /clk) clk) 2-bit)	
INT32 INT32	Wa Di Reg	L0 Ir rp Sch spatch jister FP32	si nstruct n Unit File (1	sT ion C (32 th (32 th (32 th (32 th) (32 th) (32 th) (32 th) (32 th) (32 th)	ache hread/c read/cl 4 x 32-	st clk) k) bit)			INT32	INT32	War Di Reg	L0 In rp Sch spatch jister I	struct eduler Unit (File (1	ion C (32 t (32 th 6,38	ache hread read/d 4 x 32	st /clk) clk) 2-bit)	
INT32 INT32 INT32 INT32	Wa Di Reg FP32 FP32	L0 Ir rp Sch spatch jister FP32 FP32	si nstruct n Unit File (1 FP	sT ion C (32 th (32 th (6,384 64 64	ache hread/cl 4 x 32-	st :lk) k) bit)			INT32	INT32	War Di Reg FP32 FP32	L0 In rp Sch spatch jister I FP32 FP32	eduler Unit (File (1	ion C (32 t (32 th 6,38 64 64	ache hread read/d	st /clk) clk) 2-bit)	
INT32 INT32 INT32 INT32 INT32 INT32	Wa Di Reg FP32 FP32 FP32	L0 Ir rp Sch spatch jister FP32 FP32 FP32	SI Istruct Dunit File (1	sT (32 th (32 th (32 th (32 th (32 th (32 th (32 th) (32 th) (ache hread/c read/cl 4 x 32-	st :lk) k) bit)			INT32 INT32 INT32	INT32 INT32 INT32	Wan Di Reg FP32 FP32 FP32	L0 In rp Sch spatch jister I FP32 FP32 FP32	eduler Unit (File (1 FP	ion C (32 th 32 th 6,38 64 64 64	ache hread/c 4 x 32	ST /clk) clk) 2-bit)	
INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	Wa Di Reg FP32 FP32 FP32 FP32	L0 Ir rp Sch spatch jister FP32 FP32 FP32 FP32	si istruct iedule i Unit File (1 FP FP FP	sT (32 th (32 th))))))))))))))))))))))))))))))))))))	ache hread/cl 4 x 32-	st lk) k) bit)	R CORE_		INT32 INT32 INT32	INT32 INT32 INT32	Wan Di Reg FP32 FP32 FP32 FP32	L0 In rp Sch spatch jister I FP32 FP32 FP32 FP32	struct eduler Unit (File (1 FPI FPI	ion C (32 th (32 th 6,38 64 54 54 54	ache hread read/o 4 x 32	ST /clk) clk) 2-bit)	R COR
INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	Wa Di Reg FP32 FP32 FP32 FP32 FP32	L0 Ir rp Sch spatcl gister FP32 FP32 FP32 FP32 FP32	si nstruct n Unit File (1 FP FP FP	st ion C (32 th (32 th 64 64 64 64 64 64	ache hread/cl 4 x 32- TEI	st cik) k) bit)	R CORE		INT32 INT32 INT32 INT32 INT32	INT32 INT32 INT32 INT32 INT32	War Di Reg FP32 FP32 FP32 FP32 FP32	L0 In rp Sch spatch jister I FP32 FP32 FP32 FP32 FP32	STruct eduler Unit (File (1 FPI FPI FPI	ion C (32 t (32 t 6,38 64 64 64 64 64 64	ache hread/o 4 x 32	st /clk) :lk) 2-bit)	R COR
INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	Wa Di Reg FP32 FP32 FP32 FP32 FP32 FP32	L0 Ir rp Sch spatcl jister FP32 FP32 FP32 FP32 FP32 FP32	si istruct n Unit File (1 FP FP FP FP	st ion C (32 th (32 th) 6,384 64 64 64 64 64 64	ache hread/cl 4 x 32- TEI	st cik) k) bit)	R CORE		INT32 INT32 INT32 INT32 INT32	INT32 INT32 INT32 INT32 INT32	Wan Di Regg FP32 FP32 FP32 FP32 FP32 FP32	L0 In rp Sch spatch jister I FP32 FP32 FP32 FP32 FP32 FP32	struct eduler Unit (File (1 FPI FPI FPI	ion C (32 th 32 th 6,38 64 64 64 64 64 64	ache hread/ read/c 4 x 32	st /clk) :lk) 2-bit)	R COR
INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	Wa Di Reg FP32 FP32 FP32 FP32 FP32 FP32	LO In rp Sch spatcl jister FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	si instruct eduler i Unit I FPIe (1 FP FP FP FP FP	st ion C (32 th 332 th 66,384 64 64 64 64 64 64 64	ache hread/cl 4 x 32- TEI	st cik) k) bit)	R CORE		INT32 INT32 INT32 INT32 INT32 INT32 INT32	INT32 INT32 INT32 INT32 INT32 INT32 INT32	Van Di Regg FP32 FP32 FP32 FP32 FP32 FP32 FP32	L0 In rp Sch spatcF ister I FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	struct eduler b Unit (FPIe (1 FPI FPI FPI FPI FPI	ion C (32 th 6,38 64 64 64 64 64 64 64 64 64	iache hread read/d 4 x 32	st /clk) ::lk) ENSO	R COR
INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	Wa Di Rec FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	LO In rp Sch spatch jister FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	si reduler n Unit File (1 FP FP FP FP FP FP	st ion C (32 th (32 th (32 th (32 th (32 th (32 th (32 th (32 th) (32	ache hread/ci 4 x 32-	st k) bit)	R CORE		INT32 INT32 INT32 INT32 INT32 INT32 INT32	INT32 INT32 INT32 INT32 INT32 INT32 INT32	Waa Di Reg FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	L0 inn rp Sch spatch ister i FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	File (1 File (1 FPI FPI FPI FPI FPI FPI	ion C (32 t 32 th 6,38 64 64 64 64 64 64 64	iache hread read/o 4 x 32	st /clk) 2-bit) ENSO	R COR
INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	Wa Di FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	LO IIT rp Sch spatcl jister FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	si neduler Dunit File (1 FP FP FP FP FP FP FP FP FP	st ion C (32 th 32 th 6,384 64 64 64 64 64 64 64 64 64 64 64 64 64	ache hread/cl 4 x 32- TEI	st ik) k) bit) NSOI	R CORE SFU		INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	Vaa Di Regg FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	L0 Im rp Sch spatch FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	struct eduler I Unit (FPIe (1 FPI FPI FPI FPI FPI FPI FPI FPI FPI FPI	ion C (32 t 32 th 6,38 64 64 64 64 64 64 64 64 64 64 64 64 1 27 57	ache hread/ 4 x 32	st /clk) :lk) 2-bit) ENSO	R CORI
INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	Wa Di Reg FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	LO IIT rp Sch spatcl jister FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	si neduleri 1 Unit i File (1 FP FP FP FP FP FP FP FP	sT ion C (32 th 6,384 64 64 64 64 64 64 64 64 64 64 64 64 64	ache hread/d read/cl 4 x 32- TEI	ST EIk) k) bit) bit) NSOI	R CORE SFU		NT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32 INT32	Waa Di Regg FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	L0 Im rp Sch spatch spatch FP32 FP32 FP32 FP32 FP32 FP32 FP32 FP32	struct eduler b Unit (File (1 FPI FPI FPI FPI FPI FPI FPI FPI FPI	ion C (32 th 32 th 6,38 64 64 64 64 64 64 64 64 64 5T	ache hread/ 4 x 32 LD/ ST	sT /clk) :lk) 2-bit) ENSO	R CORI

"NVIDIA A100 Tensor Core GPU Architecture"



SYSTOLIC ARRAYS

Domain-Specific Architectures





Guidelines for DSA

- Use **dedicated memories** to minimize the distance over which data is moved.
- Invest the resources saved from dropping advanced microarchitectural optimizations into **more arithmetic units or bigger memories**.
- Use the easiest form of parallelism that matches the domain.
- **Reduce data size** and type to the simplest needed for the domain.
- Use a domain-specific programming language to port code to the DSA.





J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 2018. Kung, "Why systolic architectures?" Computer, vol. 15, pp. 37–46, 1982



X3 X2 X1 W13 W11 W12 W23 W21 W22 (A) X3 X2 X1 W12 W13 W11 W21 W22 W23 (B) X3 X2 W11 W12 W13 X1 W22 W23 W21

(C)

Matrix Multiplication: systolic array way



J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 2018.



Tensor Processing Unit (TPU v3, 2018)

- 256 x 256 grid
- 90 TOPS INT8
- 75 W

RTX 8000 (Turing, 2018)

- 4 x 4 x 4 grid / tensor core
- 8 TC x 72 SM
- 130 TFLOPS FP16
- 295 W

RTX A6000 (Ampere, 2020)

- 4 x 4 x 4 grid / tensor core
- 4 TC x 84 SM
- 154 TFLOPS FP16
- 300 W

J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 2018.



"NVIDIA A100 Tensor Core GPU Architecture"





DNN OPTIMIZATIONS

for hardware acceleration





Big impact on performance, but platform specific

RTX A6000 (Ampere, 2020)

- 77 TFLOPS TF32
- 154 TFLOPS BF16/FP16
- 309 TOPS INT8
- 619 TOPS INT4



"NVIDIA A100 Tensor Core GPU Architecture"

Quantization

FP32

FP32



Sparsity

Coarse grain sparsity

Reduce workload and improve throughput

Fine grain sparsity Irregular memory access reduce throughput



Different number of data fetches and computations per output causes throughput inefficiency

"NVIDIA A100 Tensor Core GPU Architecture"



Sparse matrix packing



"NVIDIA A100 Tensor Core GPU Architecture"

H. T. Kung, et al., "Packing Sparse Convolutional Neural Networks for Efficient Systolic Array Implementations: Column Combining Under Joint Optimization," ASPLOS 2019.



Nvidia documentation

- Nvidia, "<u>Nvidia Deep Learning Performance Documentation</u>," Jul. 2020.
- Nvidia, "NVIDIA A100 Tensor Core GPU Architecture," 2020.
- S. Chetlur *et al.*, "<u>cuDNN: Efficient Primitives for Deep Learning</u>," 2014.
 Bibliography
- J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, Sixth Edition. Elsevier, 2018.
- J. Gómez Luna and O. Mutlu, "<u>Computer Architecture: GPU</u> <u>Programming</u>," ETH Zurich, Fall 2020.
- H. T. Kung, B. McDanel, and S. Q. Zhang, "Packing Sparse Convolutional Neural Networks for Efficient Systolic Array Implementations: Column Combining Under Joint Optimization," ASPLOS, pp. 821–834, 2019.
- N. P. Jouppi, C. Young, N. Patil, and D. Patterson, "A domain-specific architecture for deep neural networks," *Commun. ACM*, Aug. 2018.
- Kung, "Why systolic architectures?," *Computer*, vol. 15, Jan. 1982.