

Hello!

# CS 744: TPU

Shivaram Venkataraman

Spring 2025

# ADMINISTRIVIA

Midterm 2, April 24<sup>th</sup>

- Papers from FI Query to Luminix
- Similar format as first midterm
- Bring anything printed/written
- Details on Piazza

Poster session: May 1st

- More details soon

Midterm 2 Grading

Project check in  
feedback

~ 50-60%

→ End of this  
week

# MOTIVATION

→ Hardware that is being used

Capacity demands on datacenters → more

New workloads → ML workloads × Compute capacity

Metrics

Power/operation → one ML inference call  
one MM etc.

Performance/operation → well understood

Total cost of ownership →

→ Cost efficient

↳ Disks

CPUs

....

↳ K80s etc.

New Hardware

revolution

Goal: Improve cost-performance by 10x over GPUs

minimize

→

biggest constraint in scaling capacity!

# WORKLOAD

how much compute  
can you do per  
byte of data

operation  
intensity

Name	LOC	Layers					Nonlinear function	Weights	TPU Ops / Weight Byte	TPU Batch Size	% of Deployed TPUs in July 2016
		FC	Conv	Vector	Pool	Total					
MLP0	100	5				5	ReLU	20M	200	200	61%
MLP1	1000	4				4	ReLU	5M	168	168	
LSTM0	1000	24		34		58	sigmoid, tanh	52M	64	64	29%
LSTM1	1500	37		19		56	sigmoid, tanh	34M	96	96	
CNN0	1000		16			16	ReLU	8M	2888	8	5%
CNN1	1000	4	72		13	89	ReLU	100M	1750	32	

~ 2015 era  
of ML  
at Google

DNN: RankBrain, LSTM: subset of GNM Translate  
CNNs: Inception, DeepMind AlphaGo

"big models"  
only 5% of  
TPUs used for it

# WORKLOAD: ML INFERENCE

Quantization → Lower precision, energy use

8 bit integers

large % of cycles go towards inference

8-bit integer multiplies (unlike training), 6X less energy and 6X less area

on the chip

→ Need for "predictable latency" and not throughput  
e.g., 7ms at 99th percentile

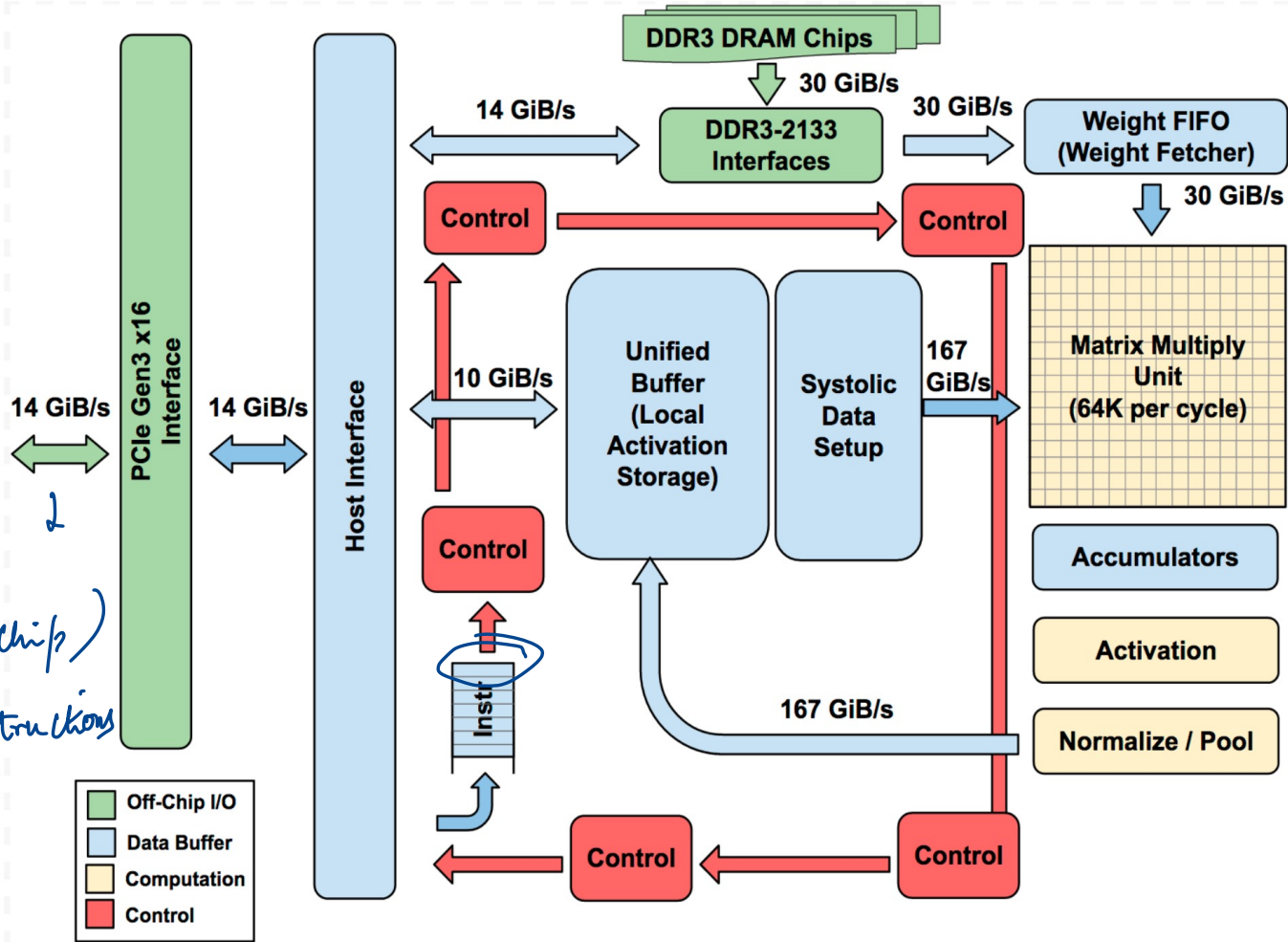
trade-off that for predictable latency

↳ SLO formulated

hardware design → SLO requirements

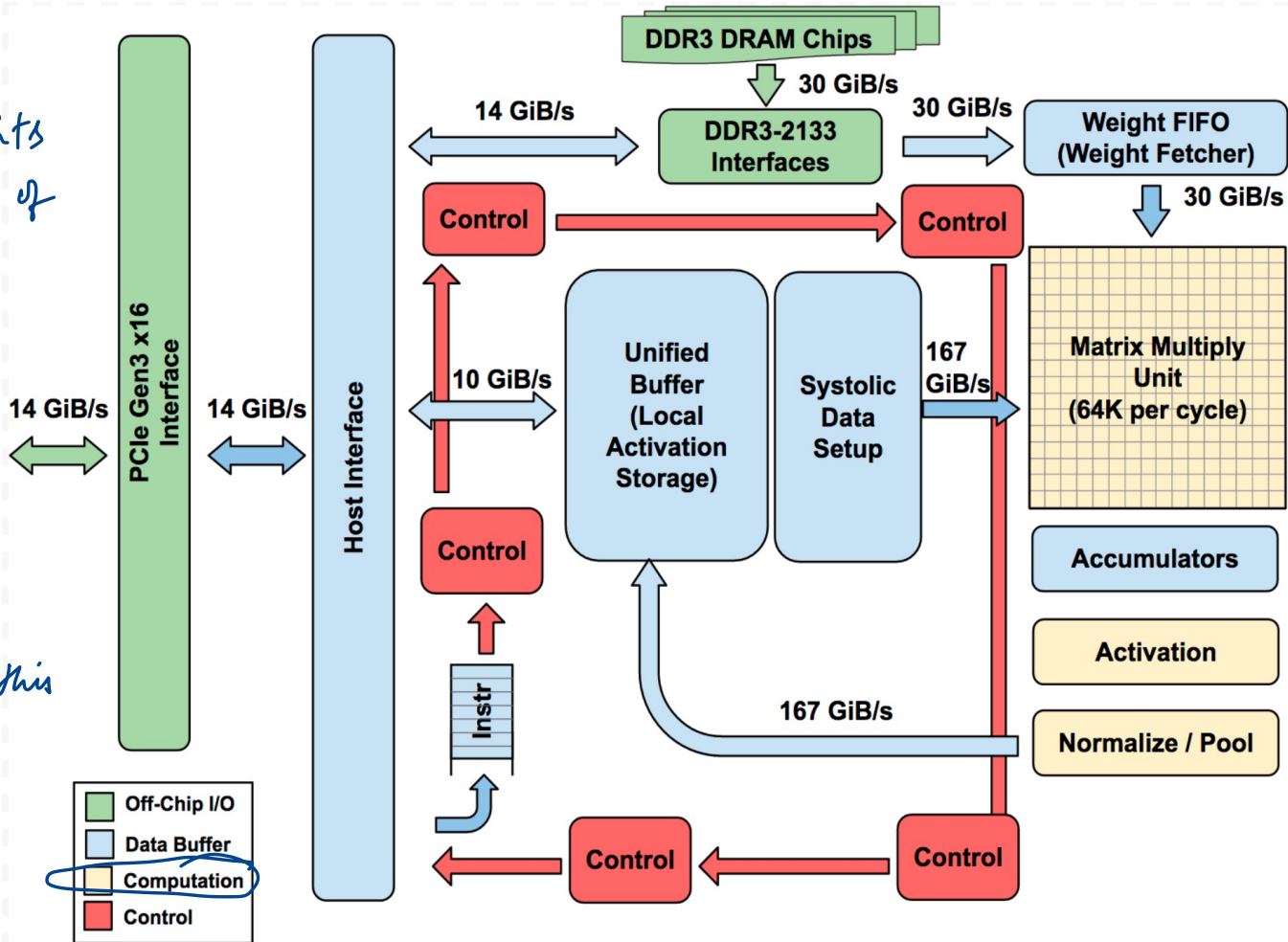
# TPU DESIGN CONTROL

- PCIe attachment
- Compatibility
- lower bandwidth  
(when compared memory bus / on chip)
- Coarse grained instructions  
↳ eg. Conv  
Matrix  
Multiply



# COMPUTE

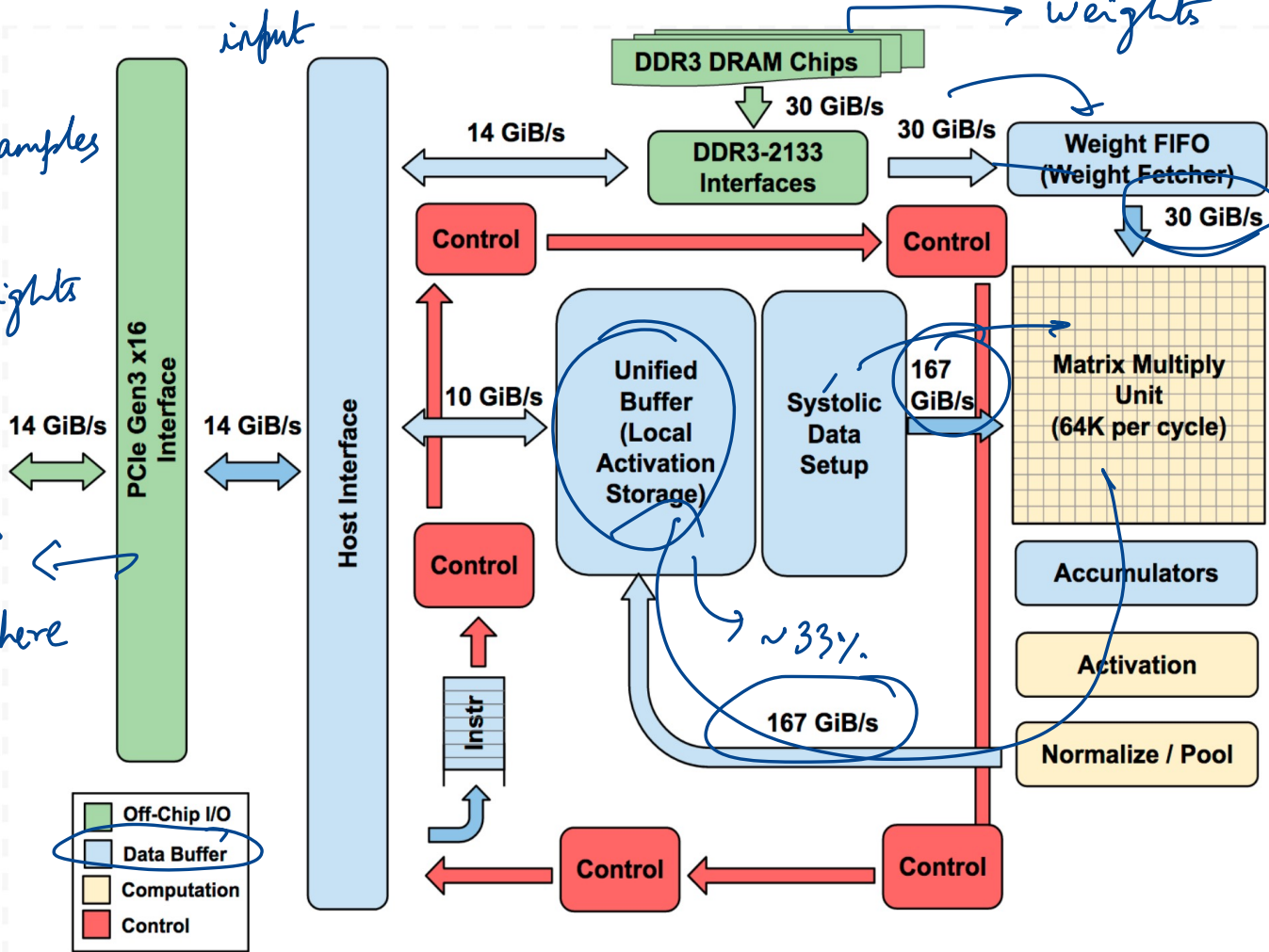
- Few compute units  
→ 25% size of die
- Specialized  
↳ \* ML models
- instruction  
does matrix  
multiply  
→ handled by this  
unit



# DATA


→ inference → examples  
out = model (data)  
↳ model weights

→ Activations & over intermediate examples go into unified buffer  
come from here



# INSTRUCTIONS

CISC format (why ?)  *Complex Instruction Set*

1. Read\_Host\_Memory
2. Read\_Weights
3. MatrixMultiply/Convolve
4. Activate  *Tanh / ReLU etc.*
5. Write\_Host\_Memory

# SYSTOLIC EXECUTION

Problem: Reading a large SRAM uses much more power than arithmetic!

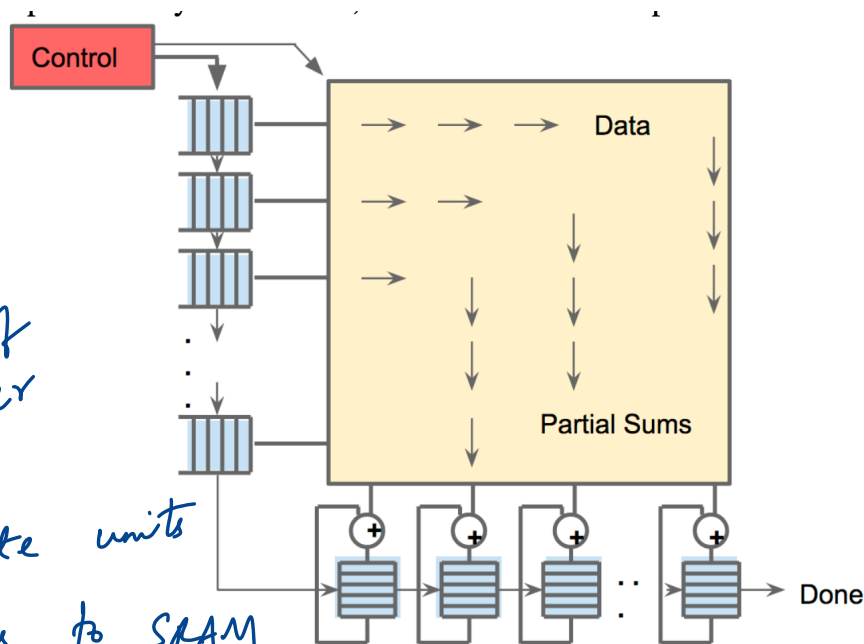
classic idea

$$c[k] = a[i] * b[j]$$

↳ read  
multiply  
write

lot of  
power

Data streamed through compute units  
minimize load & stores to SRAM



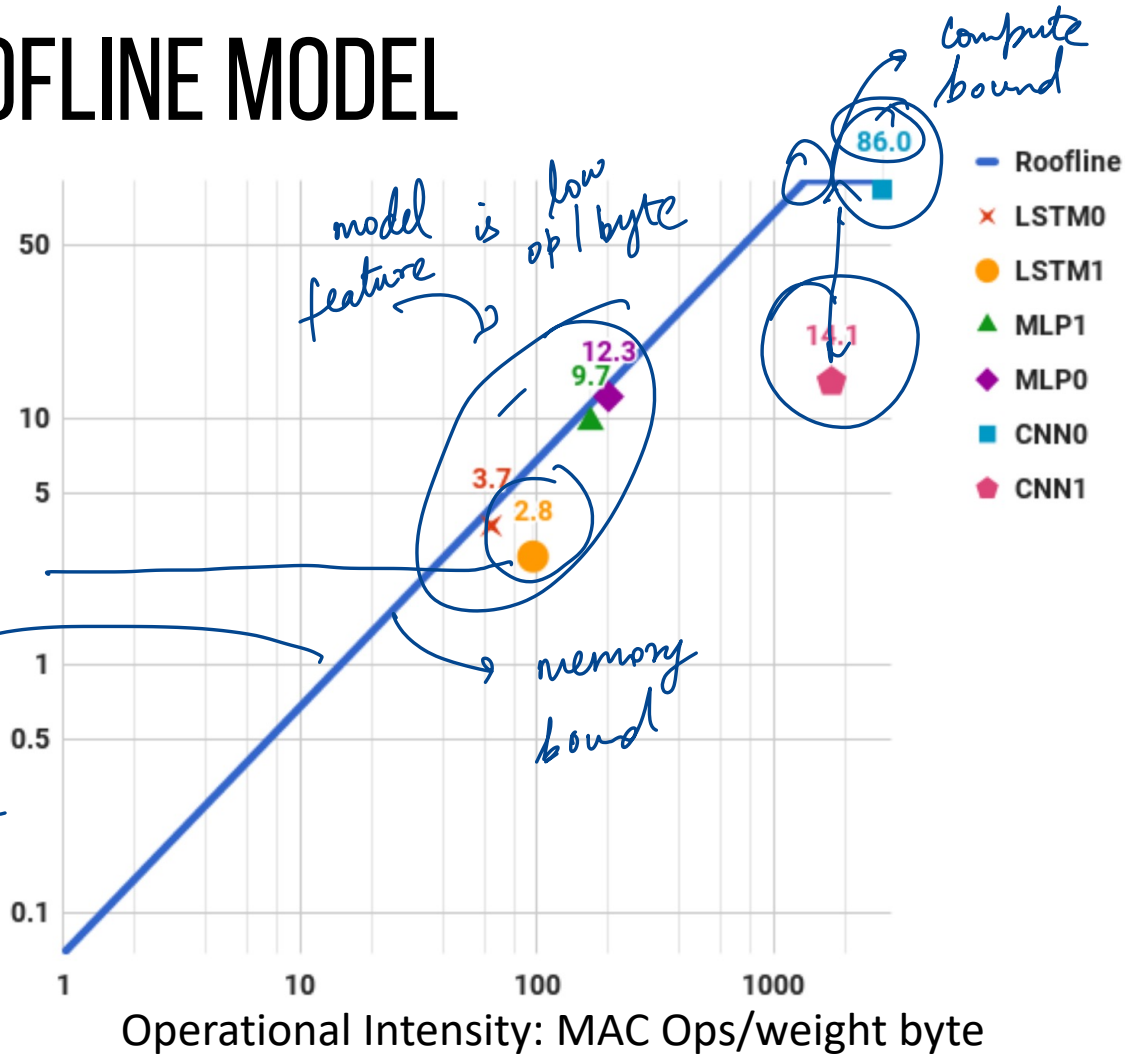
# ROOFLINE MODEL

X-axis → Operational intensity

Y-axis → Ops that you can get from hardware

maximum performance provided by this hardware

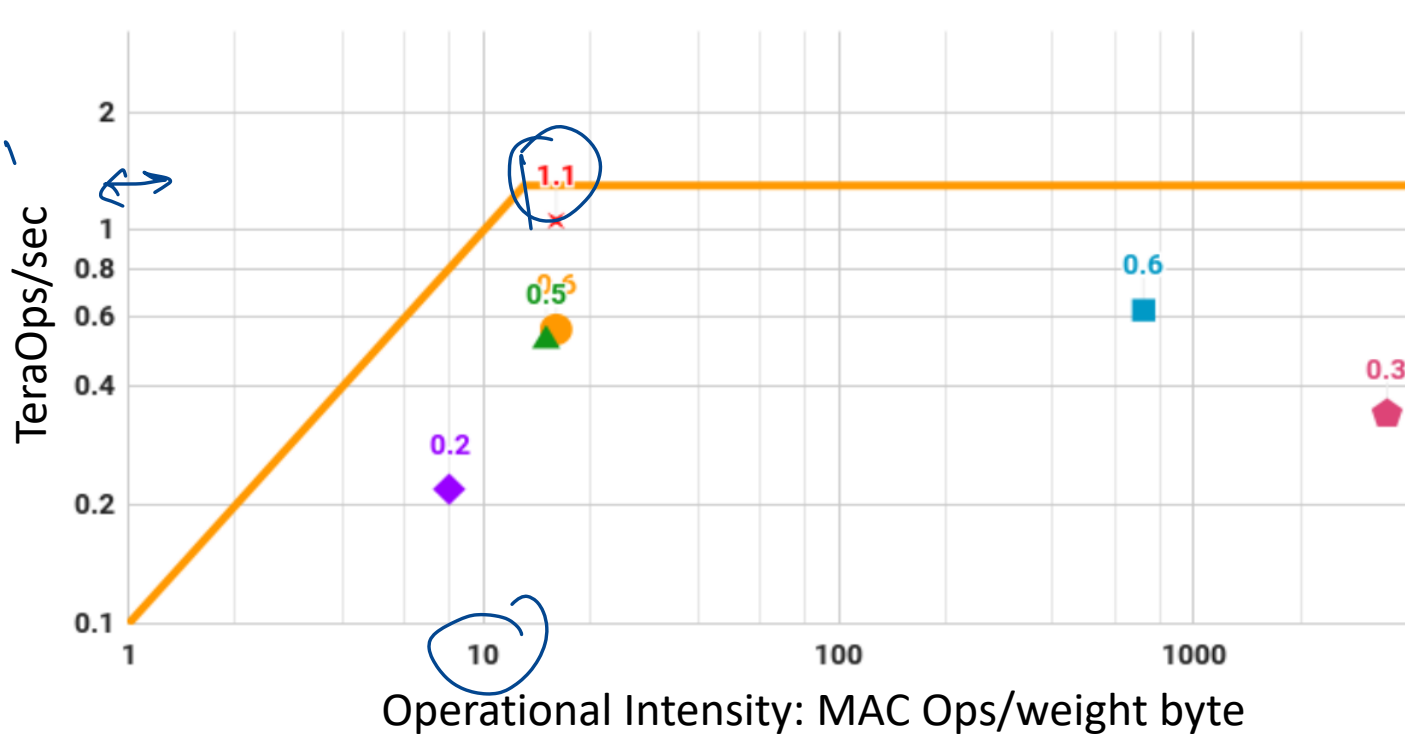
TeraOps/sec



# HASWELL ROOFLINE

CPU

TPU  
86 vs. 1.1



- Roofline
- LSTM0
- LSTM1
- MLP1
- MLP0
- CNN0
- CNN1

# COMPARISON WITH CPU, GPU

<i>Model</i>	<i>Die</i>									
	<i>mm<sup>2</sup></i>	<i>nm</i>	<i>MHz</i>	<i>TDP</i>	<i>Measured</i>		<i>TOPS/s</i>		<i>GB/s</i>	<i>On-Chip Memory</i>
					<i>Idle</i>	<i>Busy</i>	8b	FP		
Haswell E5-2699 v3	662	22	2300	145W	41W	145W	2.6	1.3	51	51 MiB
NVIDIA K80 (2 dies/card)	561	28	560	150W	25W	98W	--	2.8	160	8 MiB
TPU	<331*	28	700	75W	28W	40W	92	--	34	28 MiB

# WHAT HAPPENED NEXT?

*Ironwood*  
 ↳ new accelerator  
 inference ↗ transformer

DNN Model	TPU v1 7/2016 (Inference)	TPU v3 4/2019 (Training & Inference)	TPU v4 Lite 2/2020 (Inference)	TPU v4 10/2022 (Training)
MLP/DLRM	61%	27%	25%	24%
RNN	29%	21%	29%	2%
CNN	5%	24%	18%	12%
Transformer	--	21%	28%	57%
(BERT)	--	--	(28%)	(26%)
(LLM)	--	--	--	(31%)

*encoder + decoder*  
 ↳ decoder only

Feature	TPUv1	TPUv2	TPUv3
Peak TeraFLOPS/ Chip	92 (8b int)	46 (16b) 3 (32b)	123 (16b) 4 (32b)
Network links x Gbits/s/Chip	--	4 x 496	4 x 656
Max chips/supercomputer	--	256	1024
Peak PetaFLOPS/supercomputer	--	11.8	126
Bisection Terabits/supercomputer	--	15.9	42.0
Clock Rate (MHz)	700	700	940
TDP (Watts)/Chip	75	280	450
TDP (Kwatts)/supercomputer	--	124	594
Die Size (mm <sup>2</sup> )	<331	<611	<648
Chip Technology	28nm	>12nm	>12nm
Memory size (on-/off-chip)	28MiB/8GiB	32MiB/16GiB	32MiB/32GiB
Memory GB/s/Chip	34	700	900
MXUs/Core, MXU Size	1 256x256	1 128x128	2 128x128
Cores/Chip	1	2	2
Chips/CPU Host	4	4	8

*more  
bits*

# SUMMARY

New workloads → new hardware requirements

Domain specific design (understand workloads!)

- No features to improve the average case

- No caches, branch prediction, out-of-order execution etc.

- Simple design with MACs, Unified Buffer gives efficiency

Drawbacks

- No sparse support, training support (TPU v2, v3)

- Vendor specific ?



# DISCUSSION

<https://forms.gle/6t8EE4DPCKvszjHu7>

4x batch size  $\Rightarrow$  2.2x overall tput

same 99% latency

Type	Batch	99th% Response	Inf/s (IPS)	% Max IPS
CPU	16	7.2 ms	5,482	42%
CPU	64	21.3 ms	13,194	100%
GPU	16	6.7 ms	13,461	37%
GPU	64	8.3 ms	36,465	100%
TPU	200	7.0 ms	225,000	80%
TPU	250	10.0 ms	280,000	100%

4x batch size  $\Rightarrow$  2.2x overall tput

→

→

→

batch size ↑  
tput ↑  
tail latency get worse

get worse

higher tput  
given a fix  
99% ile latency

higher fraction of the device

How would TPUs impact serving frameworks like Nanoflow? What specific effects could it have on LLM serving architecture

Nanoflow  
batching → more  
compute  
intensive → better for TPU?

# NEXT STEPS

Next week schedule

*Thu* ~~Thu~~: Sinan (ML for Systems)

*Tue* ~~Thu~~: Luminix (Inference scheduler)