ParaDnn

github.com/Emma926/paradnn

A Systematic Methodology for Analysis of Deep Learning Hardware and Software Platforms

Yu (Emma) Wang, Gu-Yeon Wei, David Brooks
Harvard University

Contact: ywang03@g.harvard.edu
3/3/2020

HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

Acknowledgement

Frank Chen, Glenn Holloway, Dan Janni, Peter Mattson, Lifeng Nai,
David Patterson, Francesco Pontiggia, Parthasarathy Ranganathan,
Vijay Reddi, Brennan Saeta, Zak Stone, Anitha Vijayakumar, Shibo Wang,
Qiumin Xu, Doe Hyun Yoon, Cliff Young

HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

Challenges with ML Benchmarking

- Diversity in deep learning models used
  - Problem Domains, Models, Datasets
- Pace of field
  - State-of-the-art models evolve every few months
- Varying evaluation metrics
  - Accuracy, Time to train, Latency of inference
- Multi-disciplinary field
  - Algorithms, Systems, Hardware, ML Software Stacks

State of the art: MLPerf 0.6

Area	Benchmark	Dataset	Model	Reference Implementation
Vision	Image classification	ImageNet	ResNet-50	TensorFlow
	Object detection	COCO 2017	Mask R-CNN	Pytorch
	Object detection	COCO 2017	SSD-ResNet34	Pytorch
Language/ Audio	Translation	WMT Eng-Germ	Transformer	TensorFlow
	Speech recognition	WMT Eng-Germ	GNMT	PyTorch
Commerce	Recommendation	MovieLens-20M	NCF	PyTorch
Action	Reinforcement Learning	Go	Mini-go	TensorFlow

State of the art: MLPerf 0.6

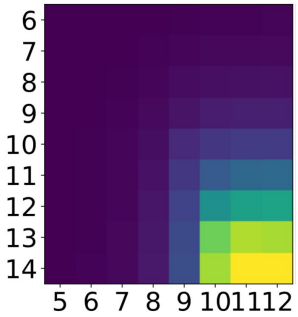
Area	Benchmark	Dataset	Model	Reference Implementation
Vision	Image classification	ImageNet	ResNet-50	TensorFlow
	Object detection	COCO 2017	Mask R-CNN	Pytorch
	Object detection	COCO 2017	SSD-ResNet34	Pytorch
Language/ Audio	Translation	WMT Eng-Germ	Transformer	TensorFlow
	Speech recognition	WMT Eng-Germ	GNMT	PyTorch
Commerce	Recommendation	MovieLens-20M	NCF	PyTorch
Action	Reinforcement Learning	Go	Mini-go	TensorFlow

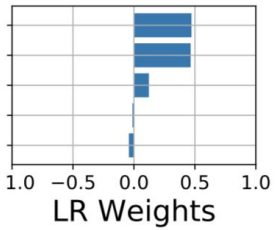
Our Methodology

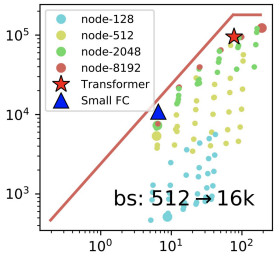
ParaDnn

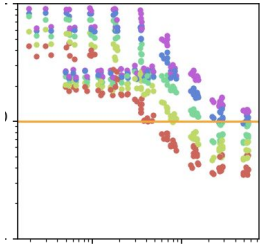
Our Methodology

ParaDnn +









ParaDnn vs MLPerf

ParaDnn

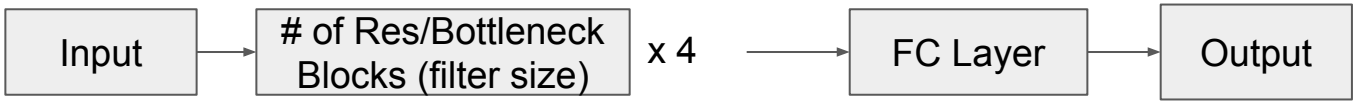
- Avoid drawing conclusions based on several arbitrary models
- Generate **thousands** of **parameterized**, **end-to-end** models
- Prepare hardware designs for future models
- Complement the use of existing real-world models, i.e. MLPerf

- Good for studying accuracy or convergence with real datasets
- Represent the specific models some people care about

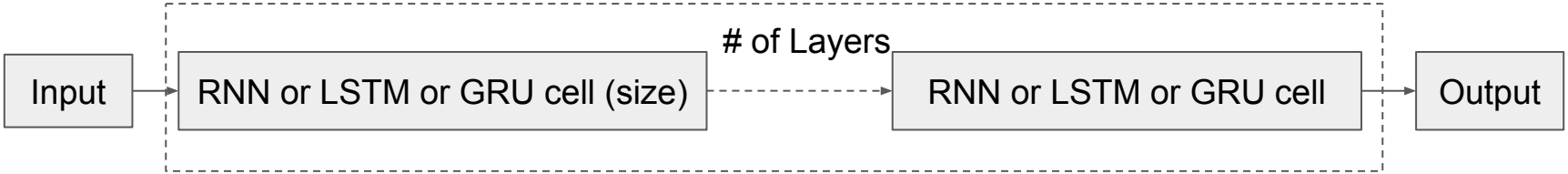
ParaDnn Canonical Models

Fully Connected (FC)

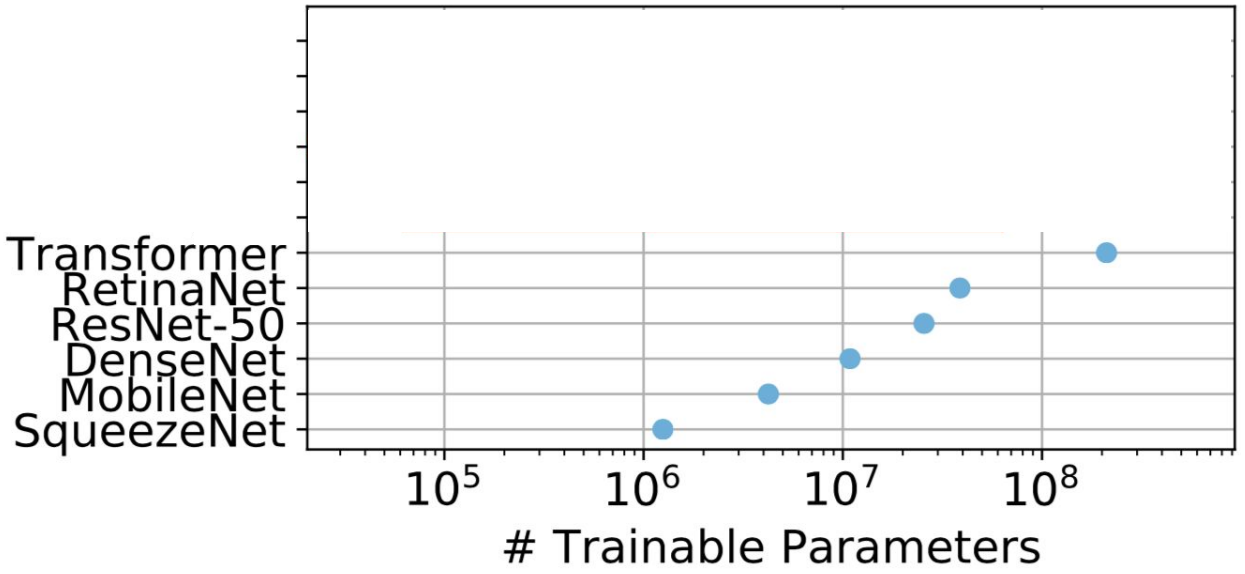
CNNs: Residual, Bottleneck



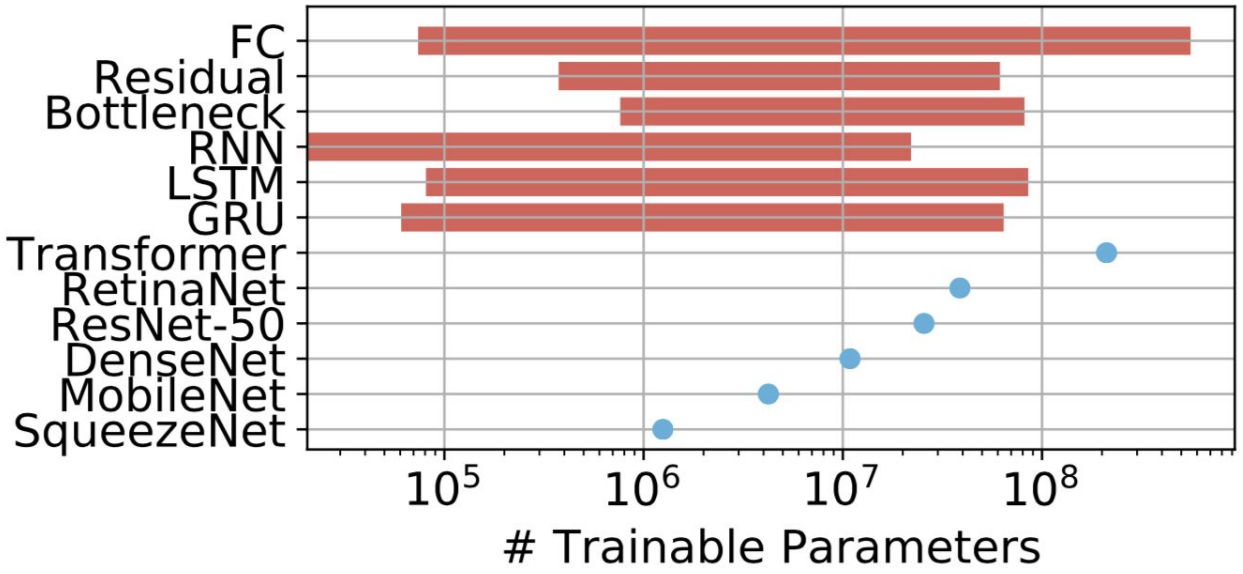
RNNs: RNN, LSTM, GRU



Models


Transformer
RetinaNet
ResNet-50
DenseNet
MobileNet
SqueezeNet
10^5
10^6
10^7
10^8
Trainable Parameters

Models

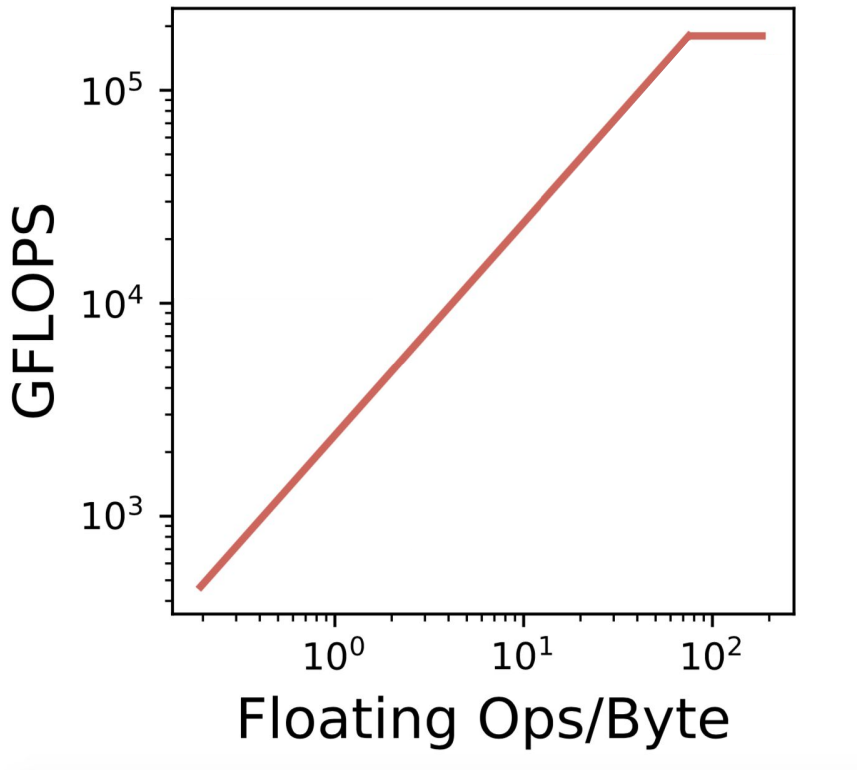


- ParaDnn covers a larger range than the real models
  - from 10k to ~1 billion parameters

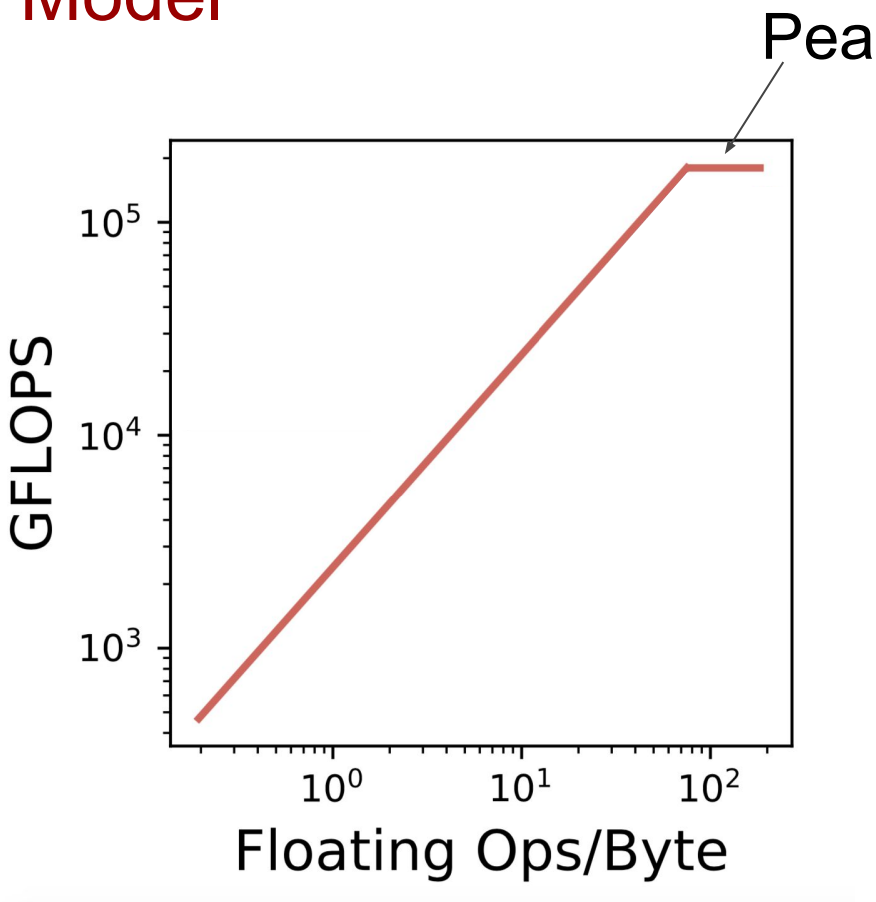
Analysis Enabled by ParaDnn

- Roofline analysis of TPU v2
- Homogenous Platform Comparison: TPU v2 vs v3
- Heterogeneous Platform Comparison: TPU vs GPU

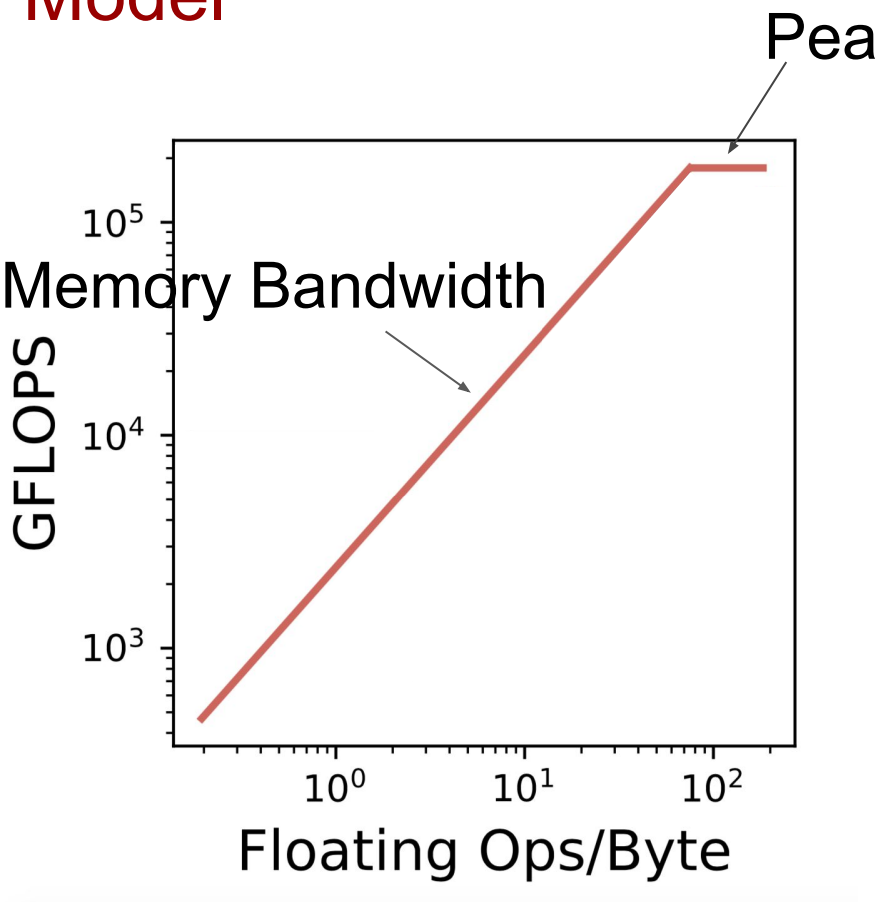
The Roofline Model



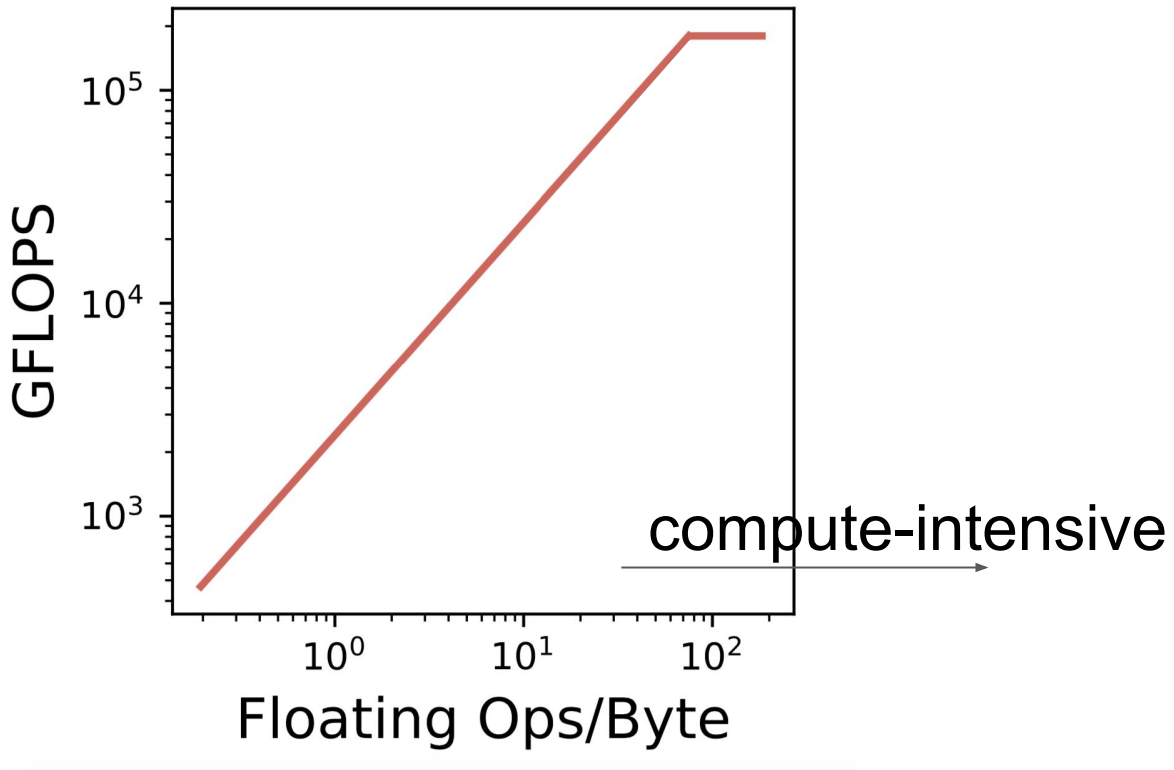
The Roofline Model



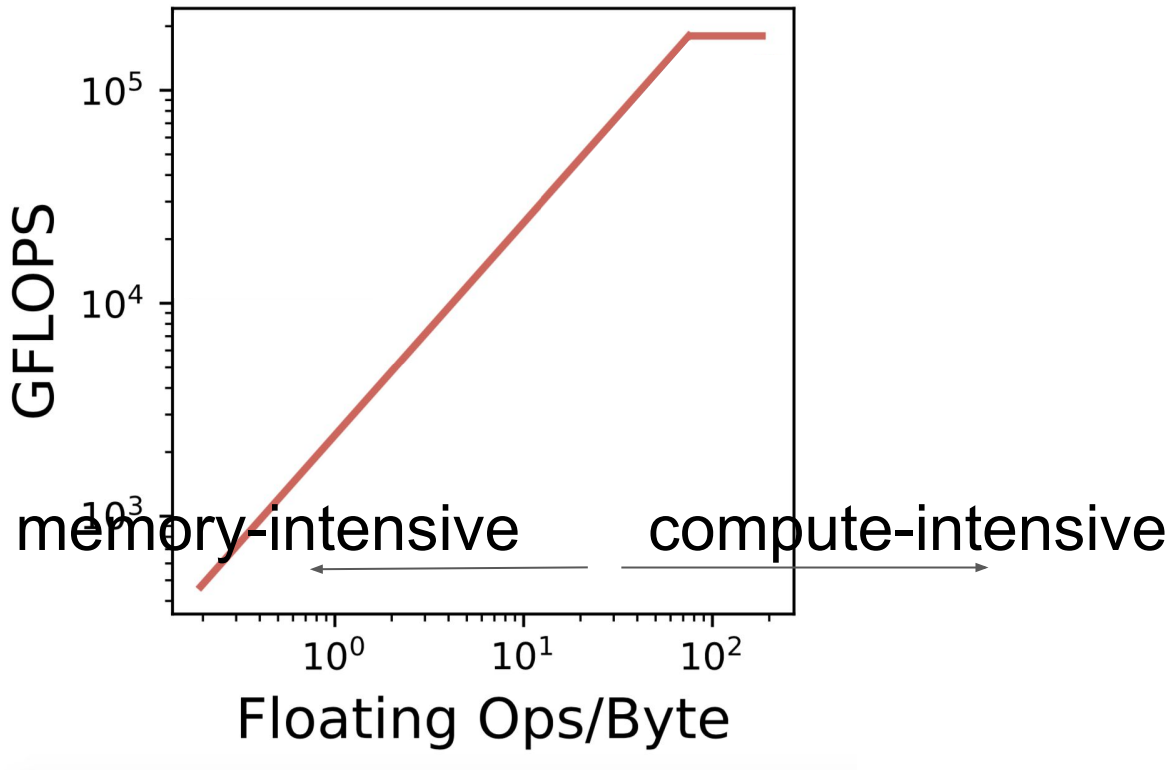
The Roofline Model



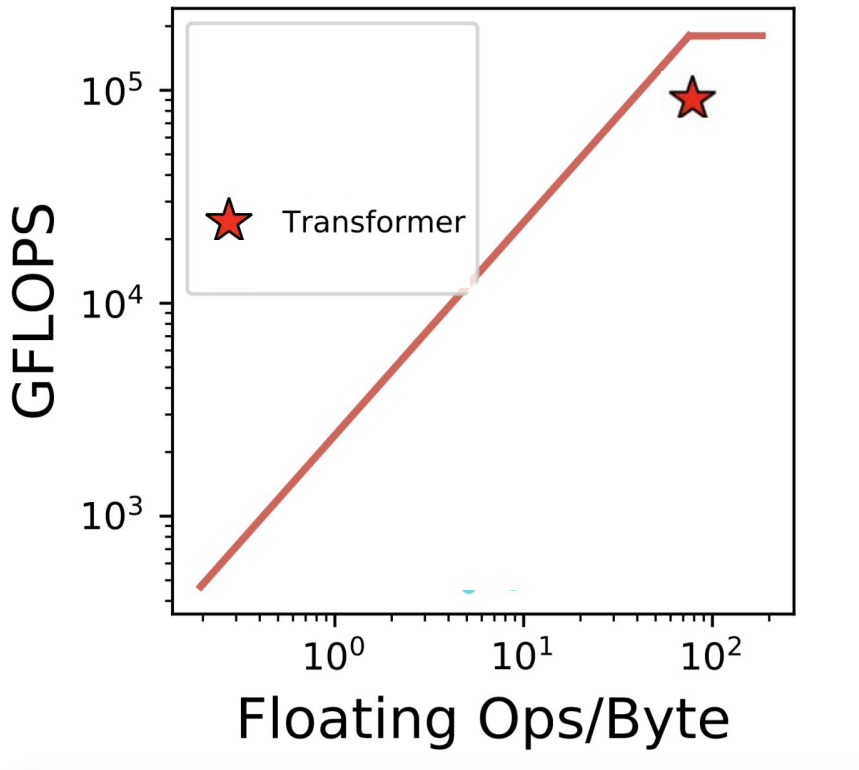
The Roofline Model



The Roofline Model

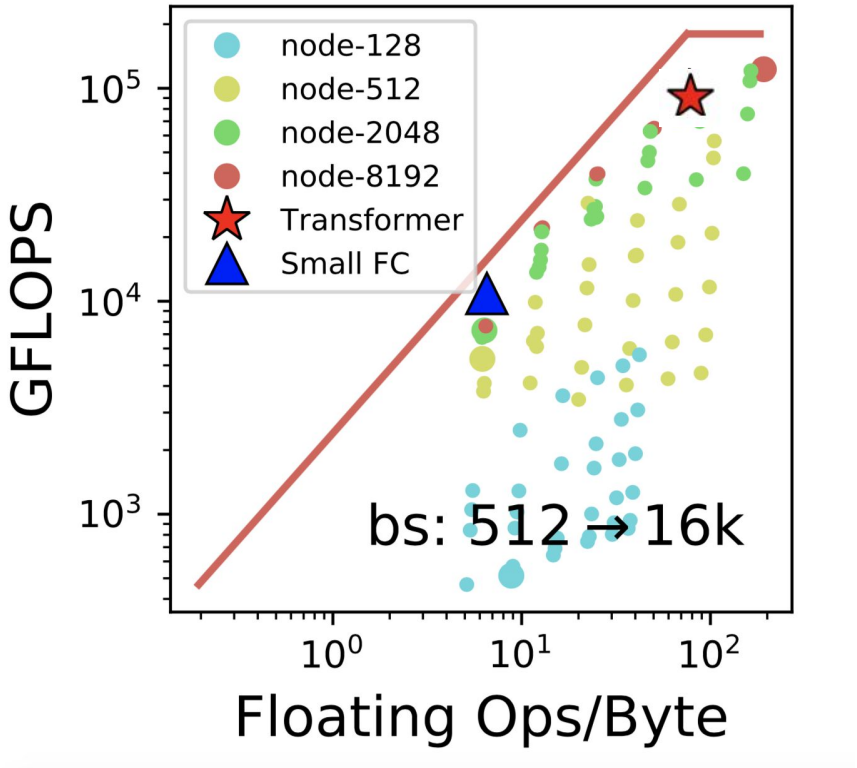


Transformer

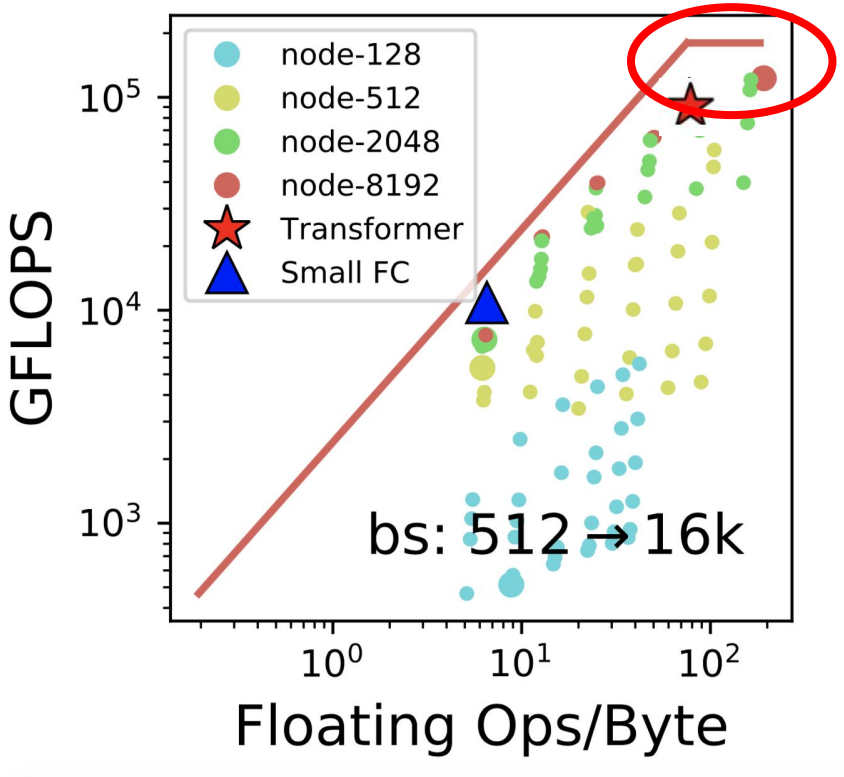


FC Models

ParaDnn sweeps a large range of models, from memory-bound to compute-bound.

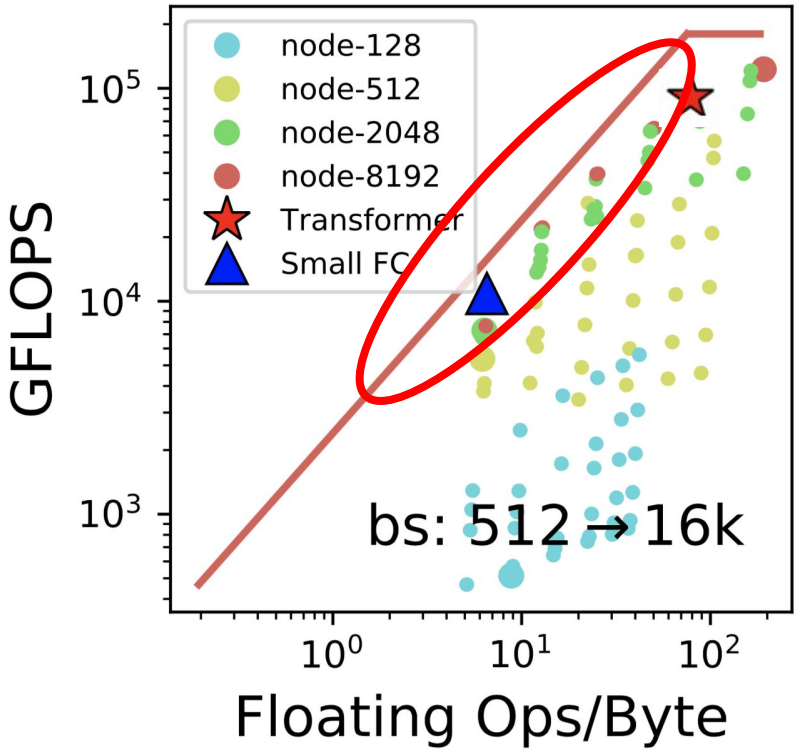


FC Models



Compute-bound

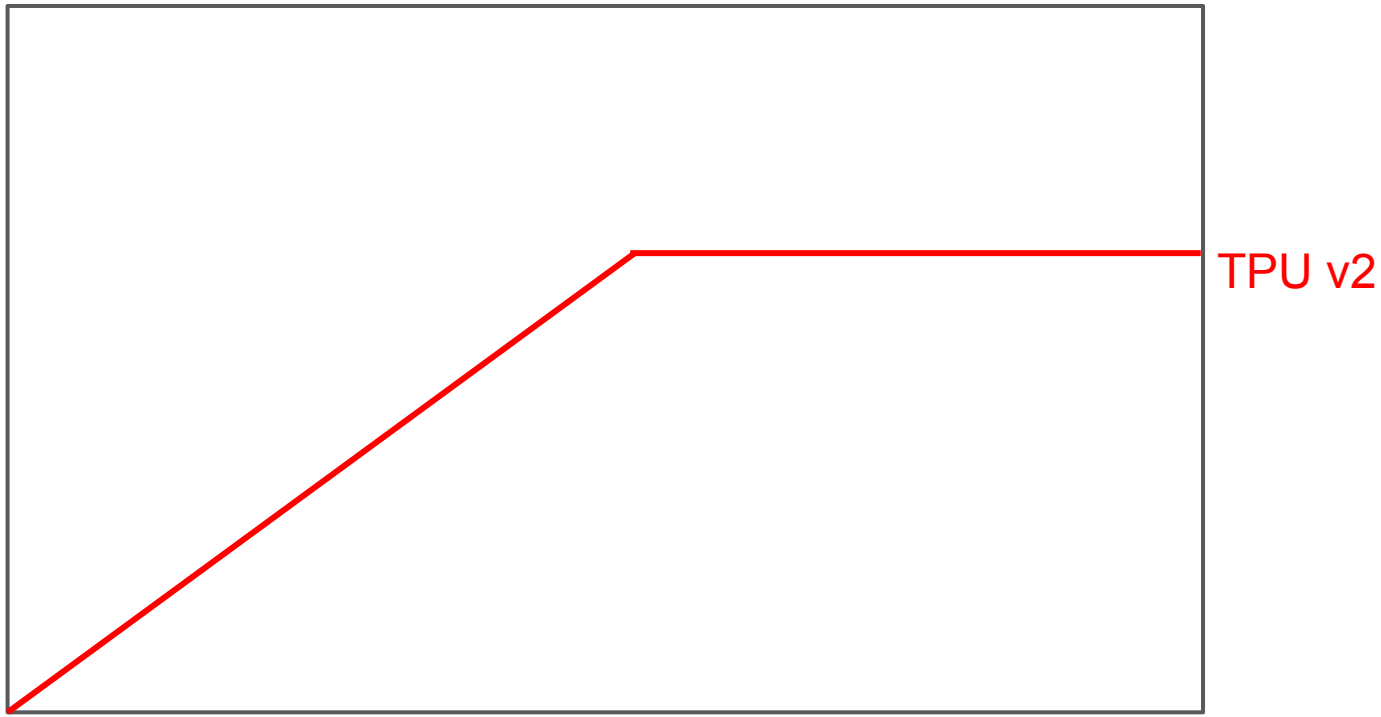
FC Models



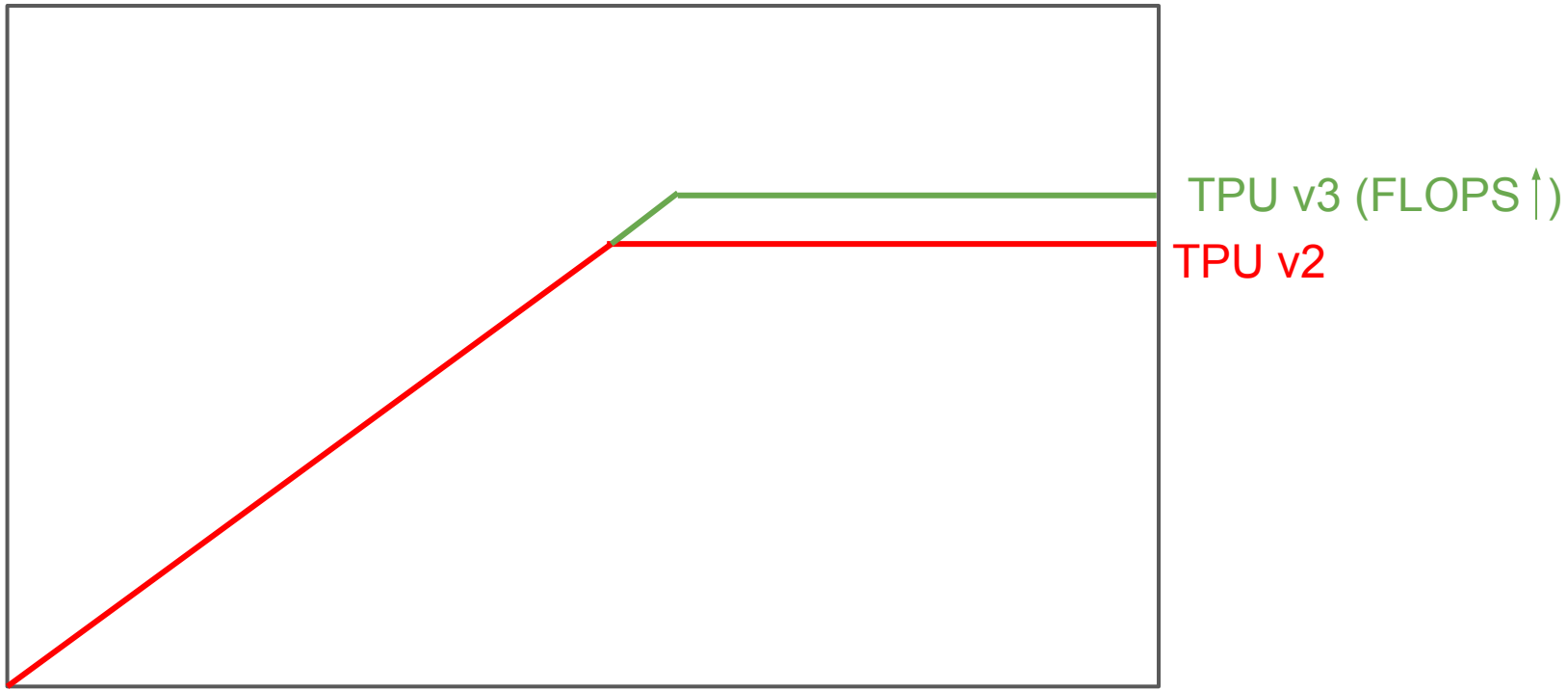
Memory-bound

TPU v2 vs v3?

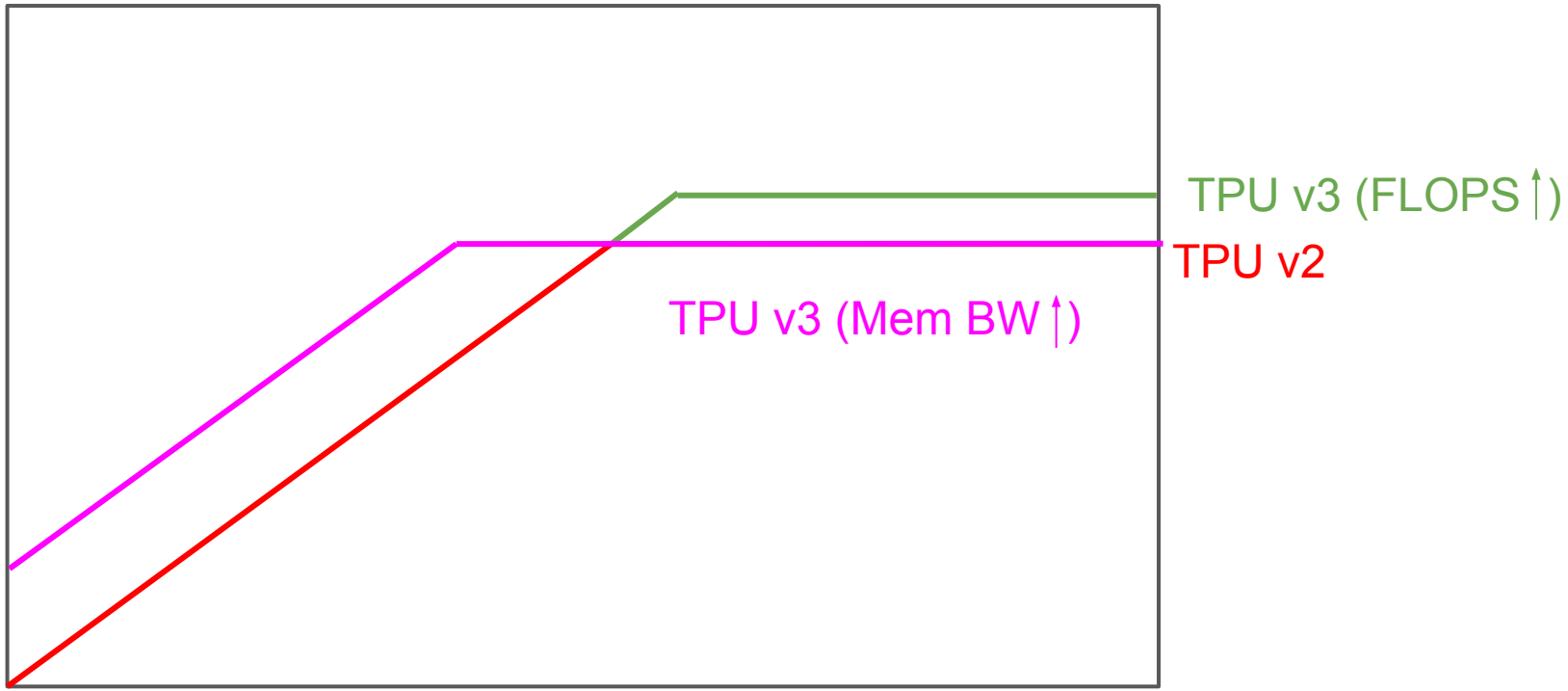
How to upgrade to TPU v3?



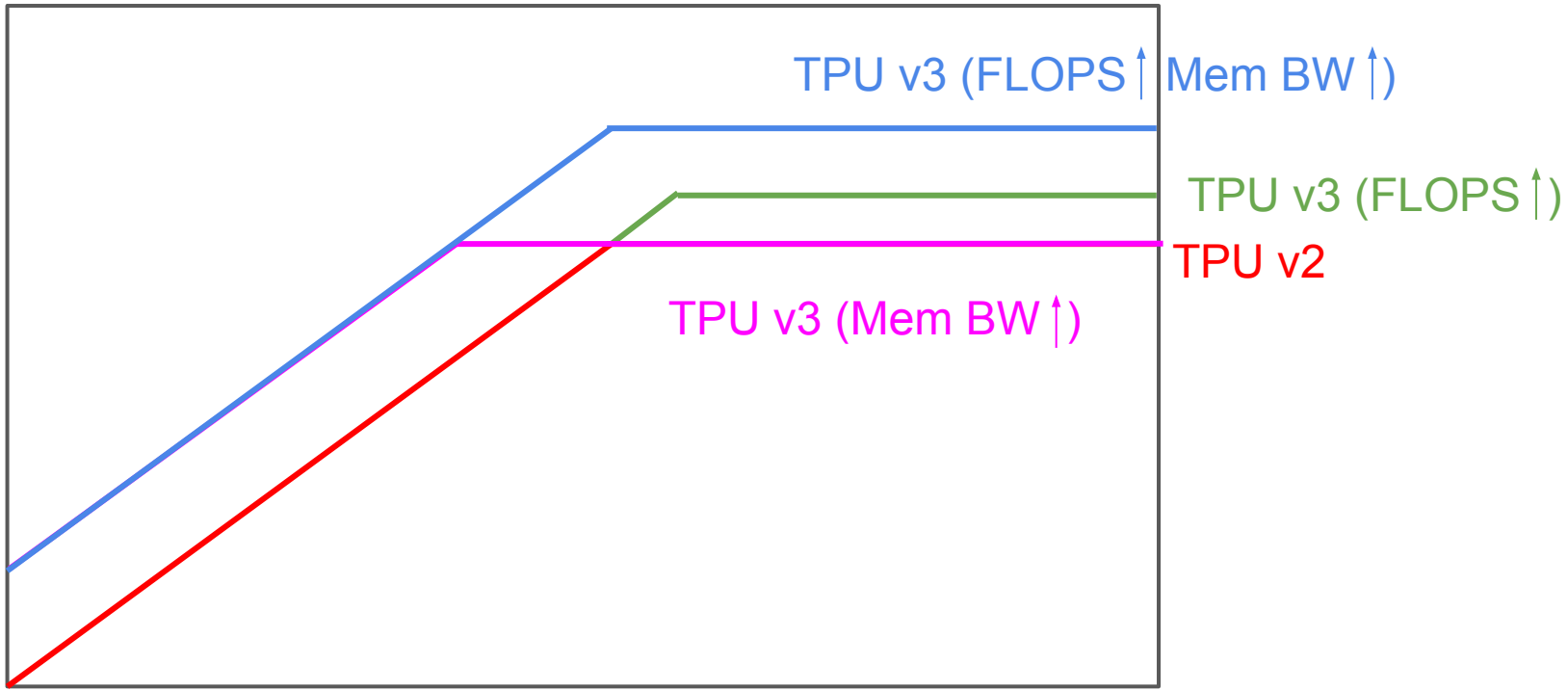
How to upgrade to TPU v3?



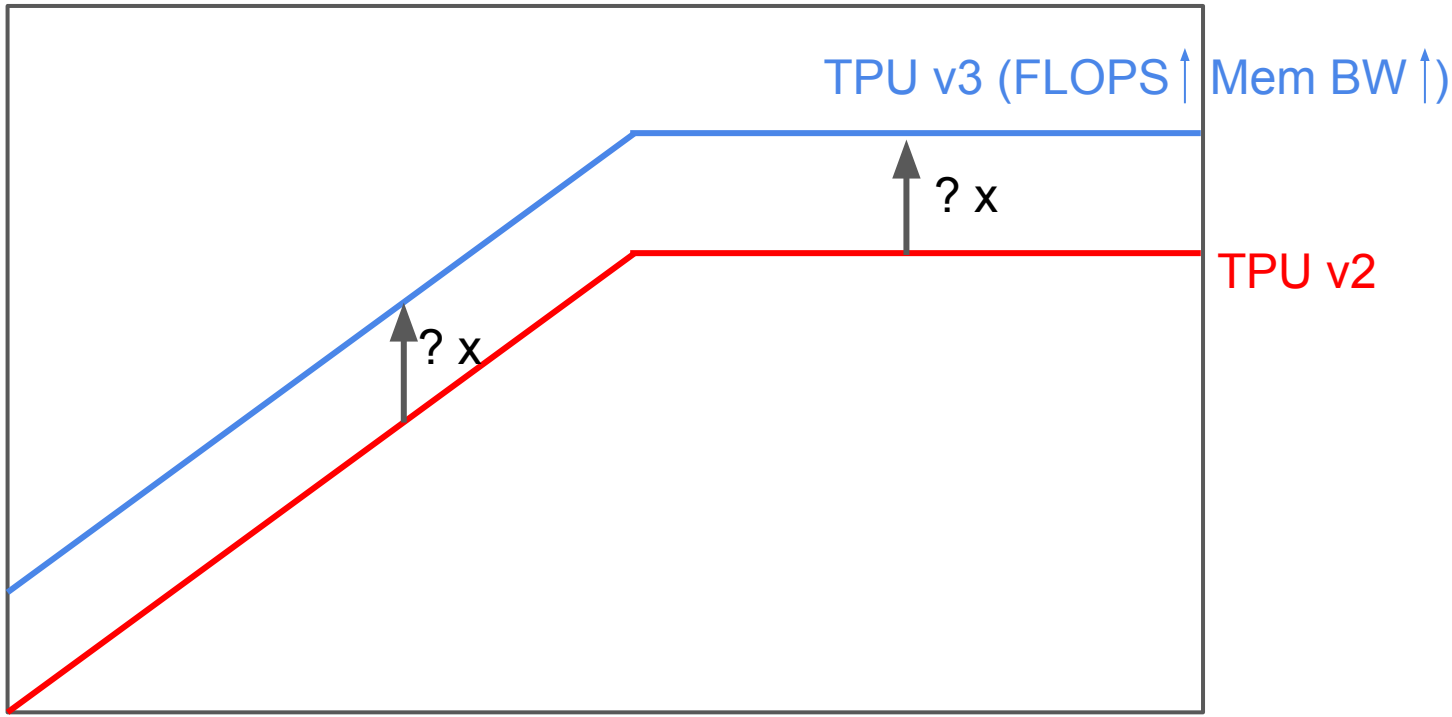
How to upgrade to TPU v3?



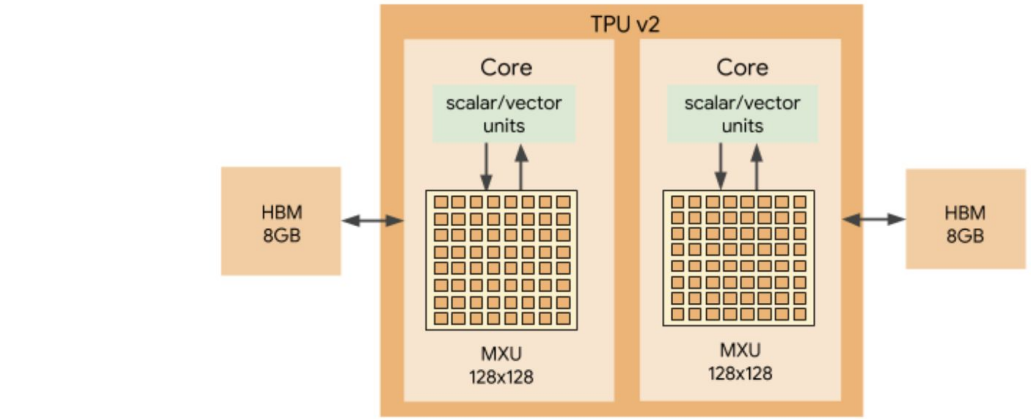
How to upgrade to TPU v3?



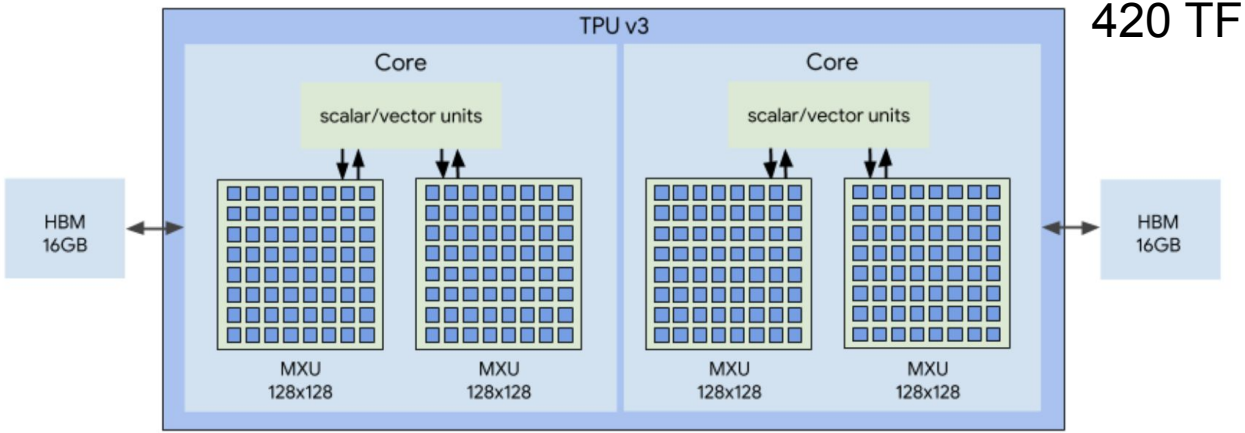
How to upgrade to TPU v3?



Architecture of TPU v2 vs v3



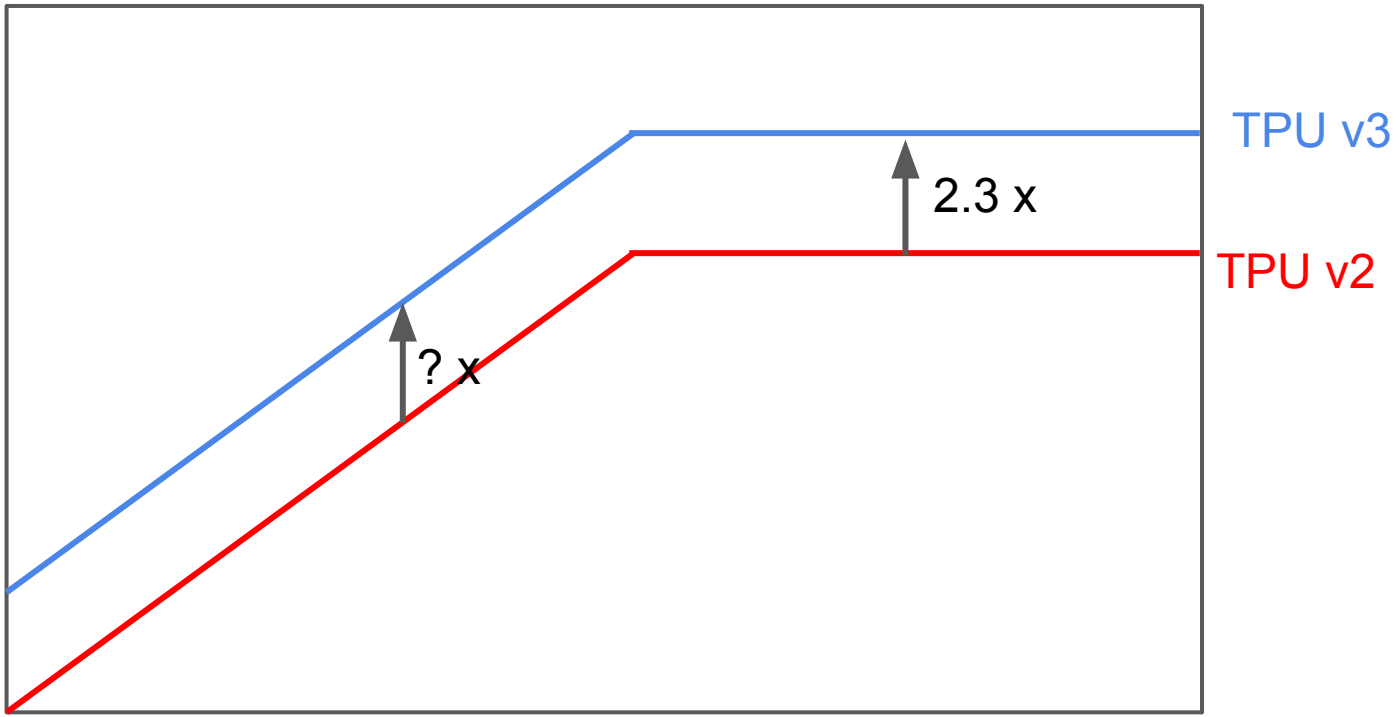
180 TFLOPS / Board



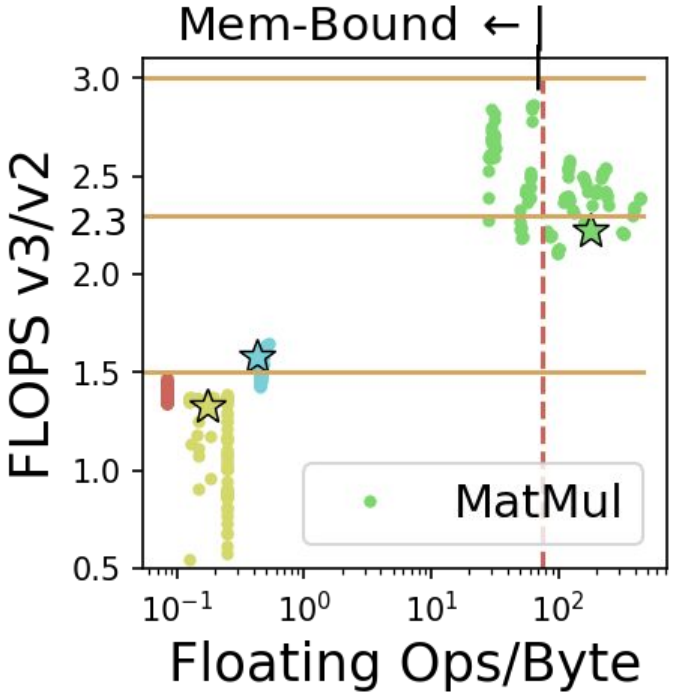
420 TFLOPS / Board

Figure is from https://cloud.google.com/tpu/docs/system-architecture

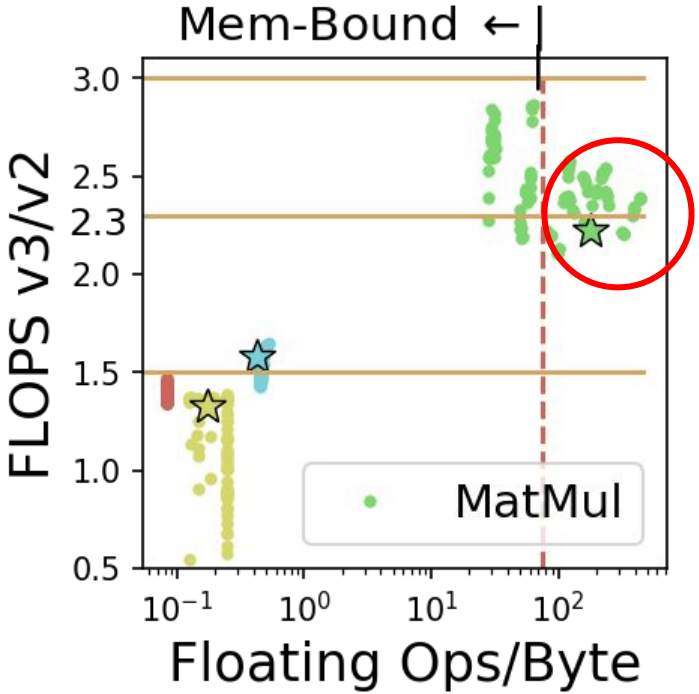
Google's Choice of TPU v3



TPU v3 vs v2: FC Operation Breakdown

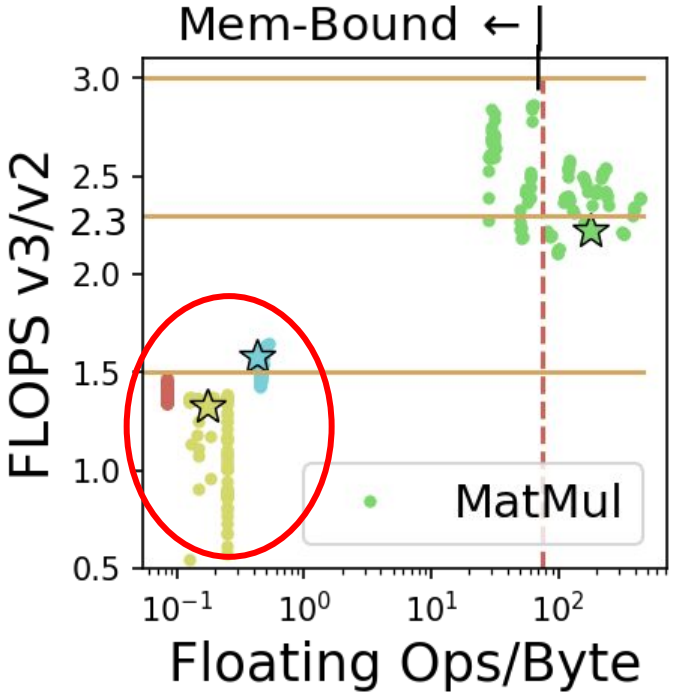


TPU v3 vs v2: FC Operation Breakdown



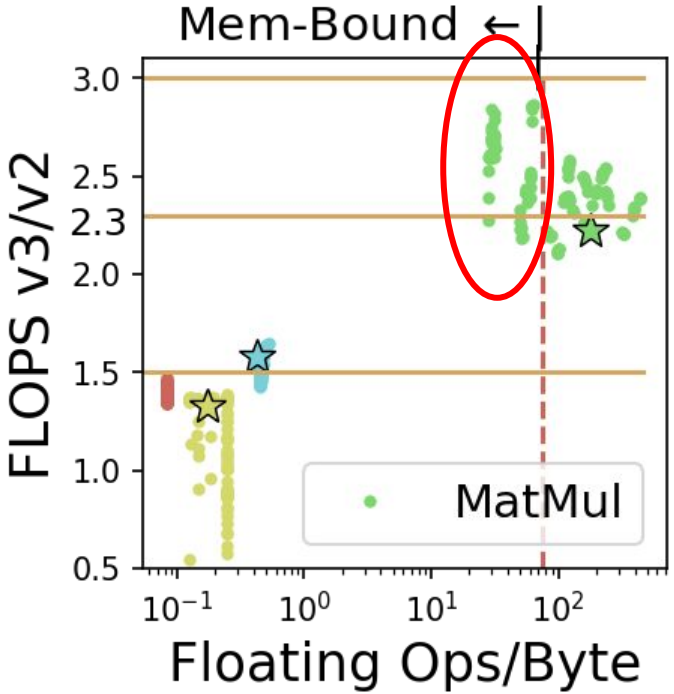
Compute-bound:
2.3x speedup

TPU v3 vs v2: FC Operation Breakdown



Memory-bound: 1.5x speedup

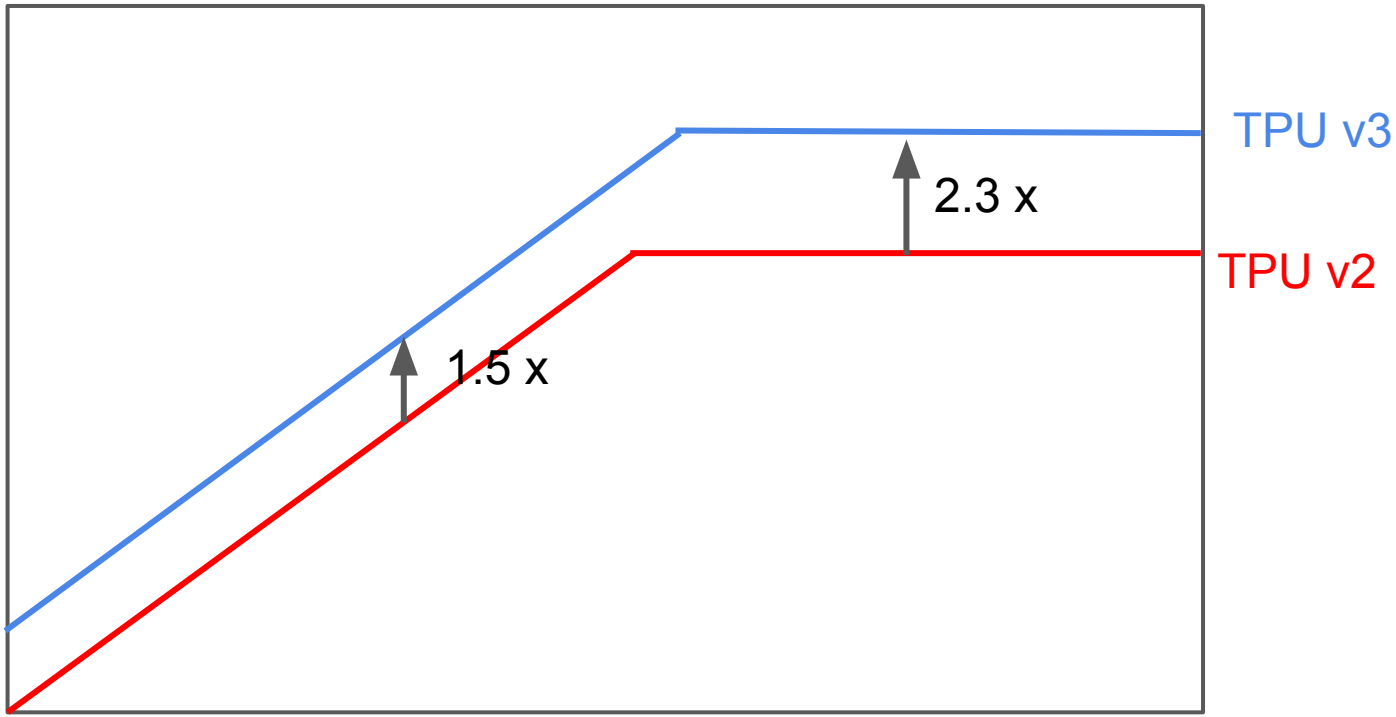
TPU v3 vs v2: FC Operation Breakdown



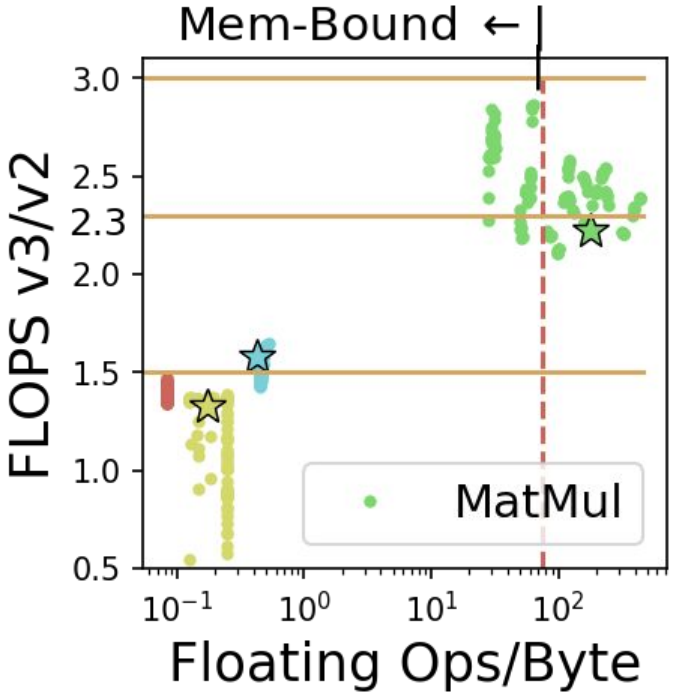
Memory-bound, but benefit from 2x memory capacity:

3x speedup

Google's Choice of TPU v3



TPU v3 vs v2: FC Operation Breakdown



ParaDnn provides diverse set of operations, and shows different operations are sensitive to different system component upgrades.

TPU vs GPU?

Hardware Platforms

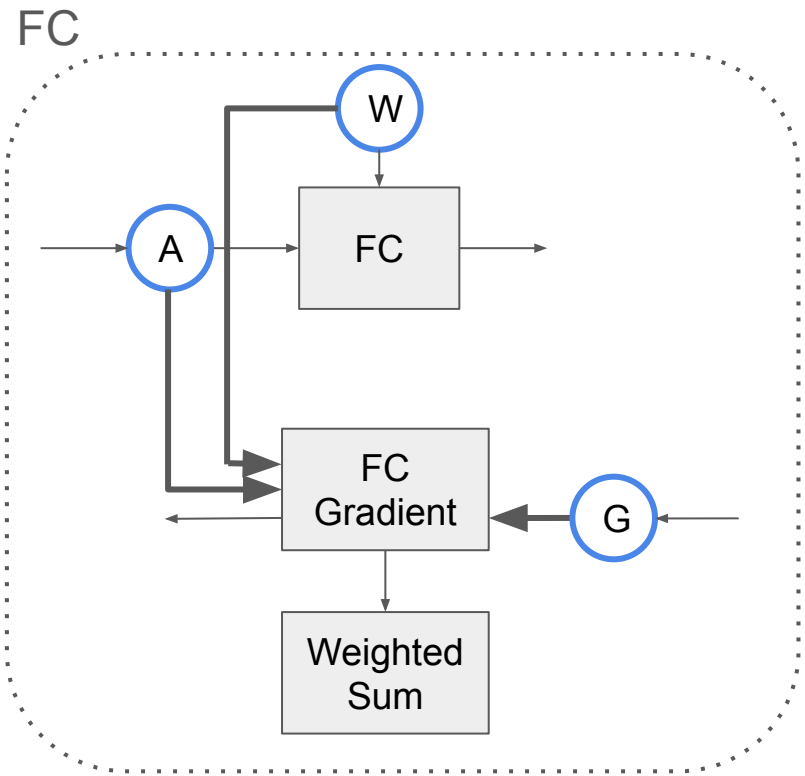
Platform	*Unit*	*Version*	*Mem Type*	*Mem (GB)*	*Mem Bdw (GB/s)*	Peak *FLOPS*
GPU (DGX-1)	1 Pkg	V100 (SXM2)	HBM2	16	900	125T
TPU	1 Board (8 cores)	v2	HBM	8	2400	180T

Hardware Platforms

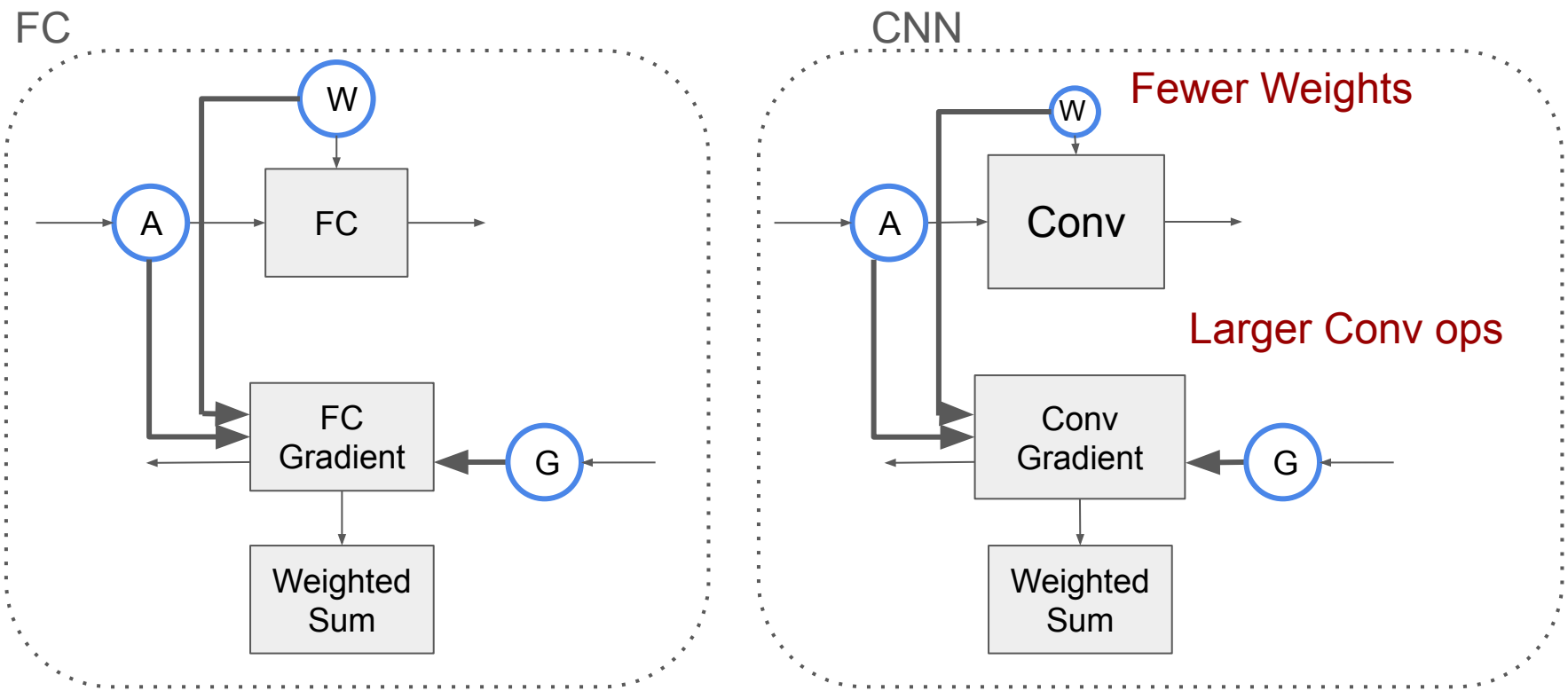
Platform	*Unit*	*Version*	*Mem Type*	*Mem (GB)*	*Mem Bdw (GB/s)*	Peak *FLOPS*
GPU (DGX-1)	1 Pkg	V100 (SXM2)	HBM2	16	900	125T
TPU	1 Board (8 cores)	v2	HBM	8	2400	180T

300 GB/s per core

FC and CNN



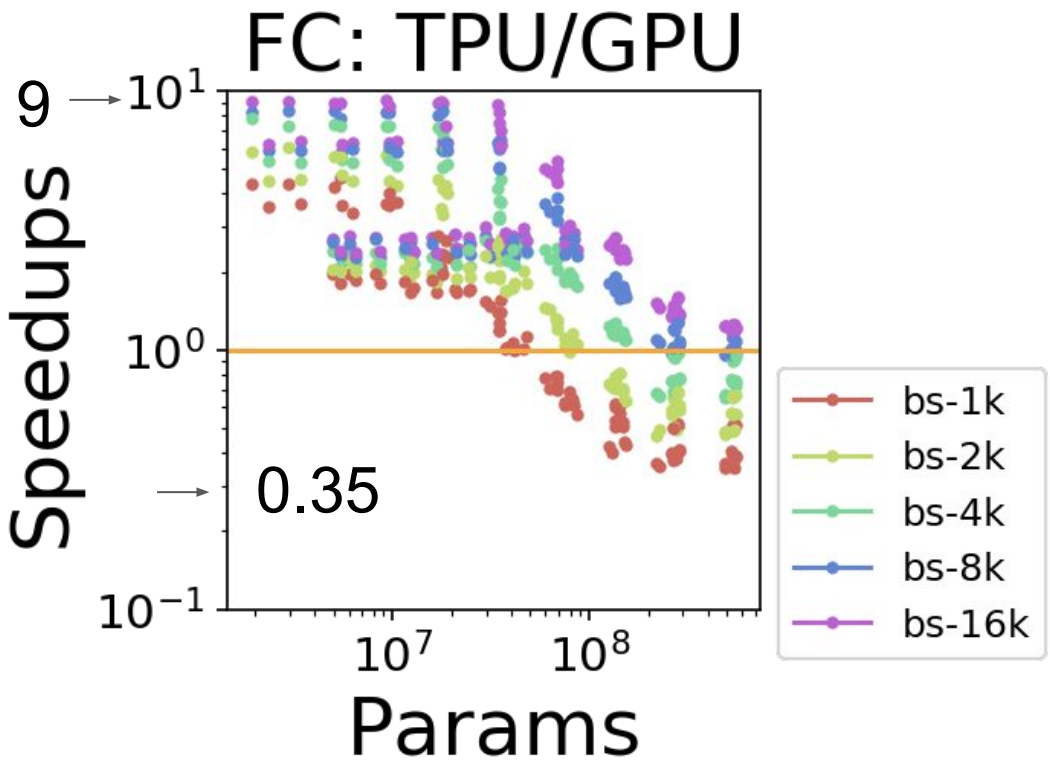
FC and CNN


FC
W
A
FC
FC Gradient
G
Weighted Sum
CNN
Fewer Weights
W
A
Conv
Larger Conv ops
Conv Gradient
G
Weighted Sum

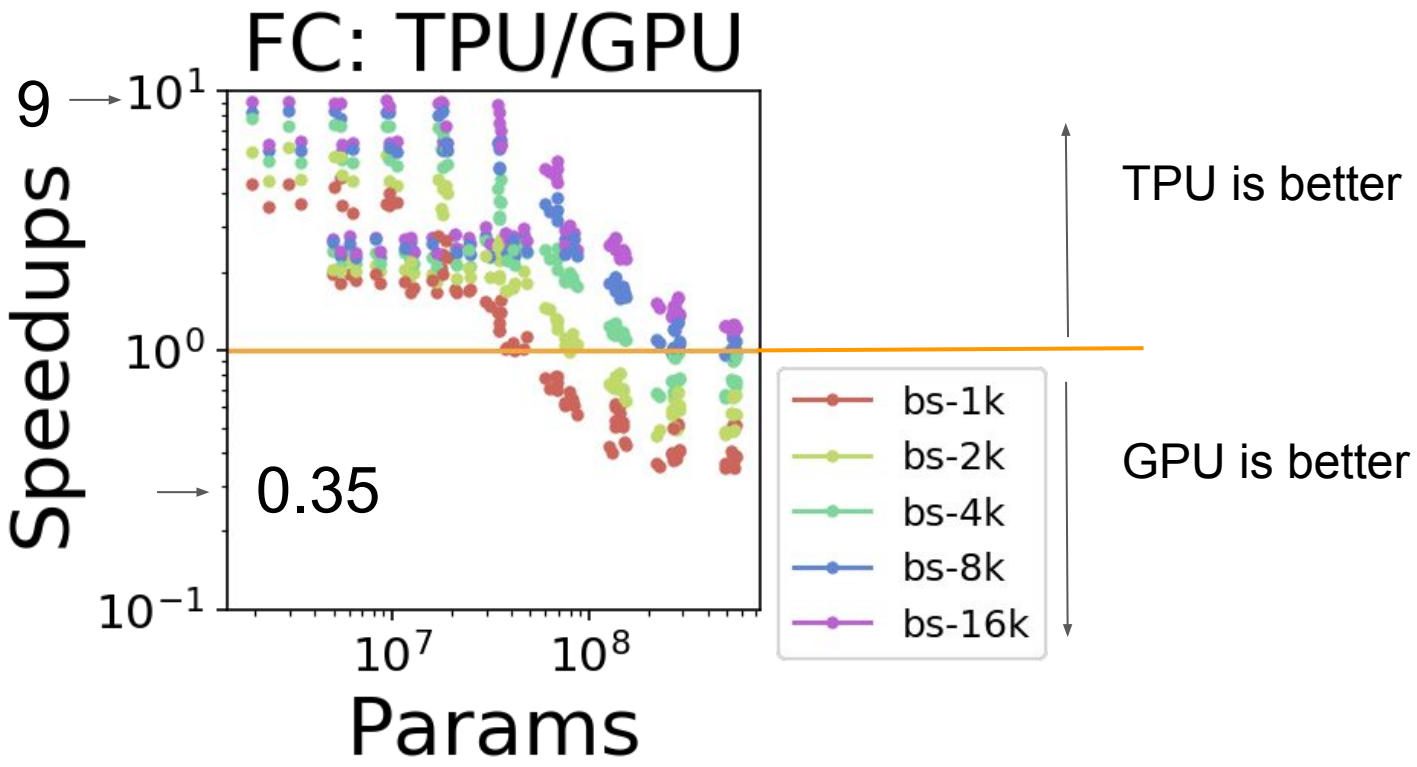
Hardware Platforms

Platform	*Unit*	*Version*	*Mem Type*	*Mem (GB)*	*Mem Bdw (GB/s)*	Peak *FLOPS*
GPU (DGX-1)	1 Pkg	V100 (SXM2)	HBM2	16	900	125T
TPU	1 Board (8 cores)	v2	HBM	8	2400	180T
					300 GB/s per core	

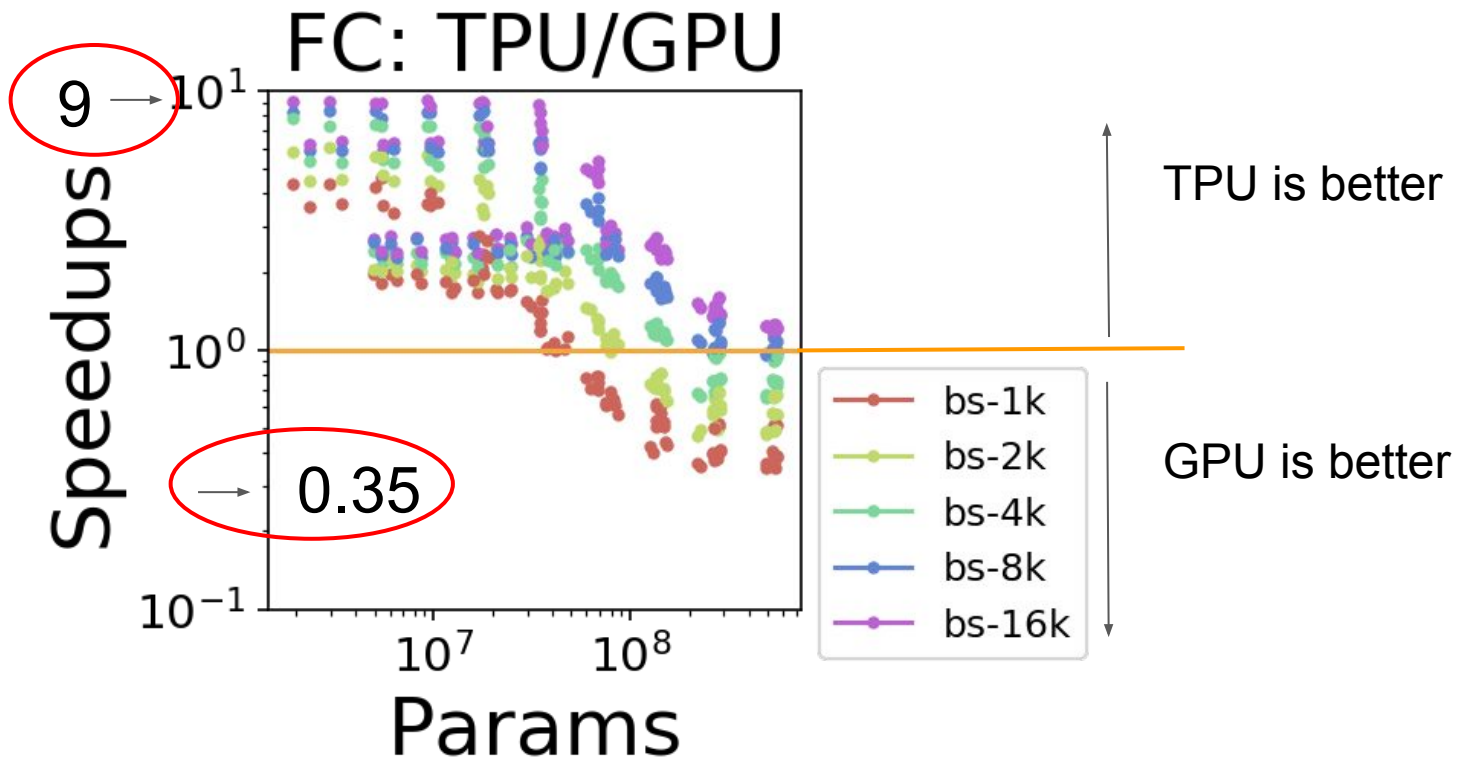
FC TPU/GPU Speedup colored with Batch Size



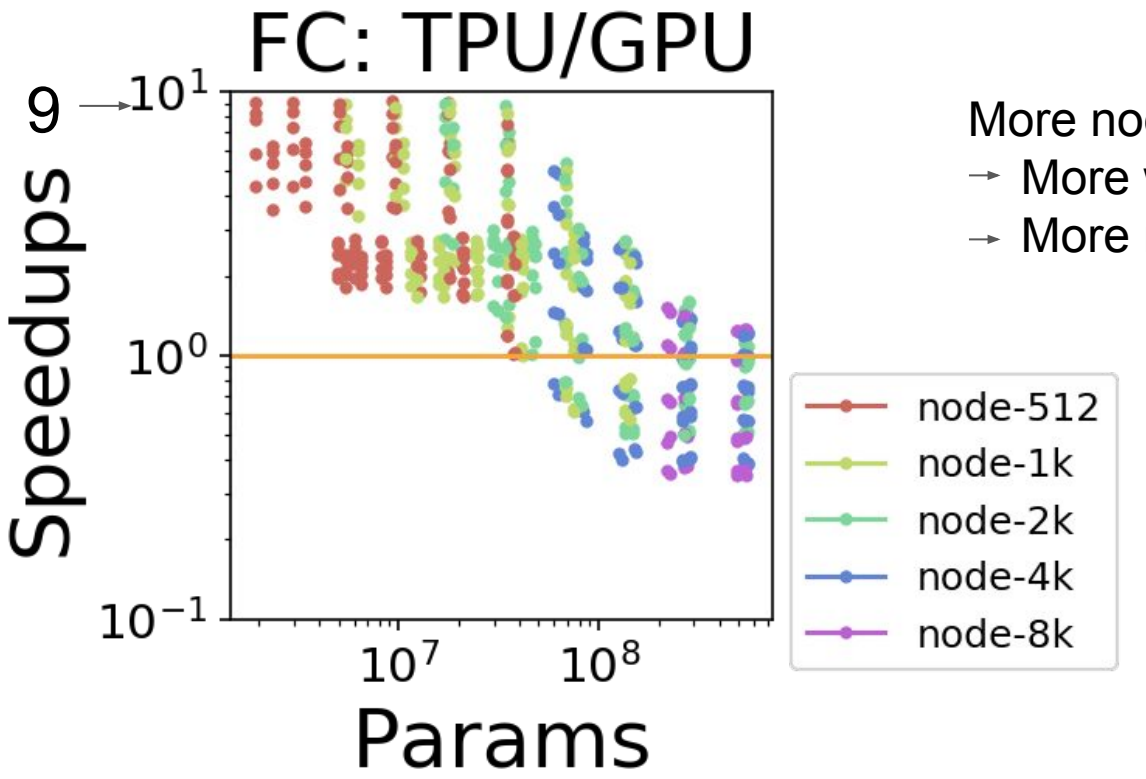
FC TPU/GPU Speedup colored with **Batch Size**



FC TPU/GPU Speedup colored with **Batch Size**



FC TPU/GPU Speedup colored with **Node Size**



More nodes
→ More weights
→ More memory-bound

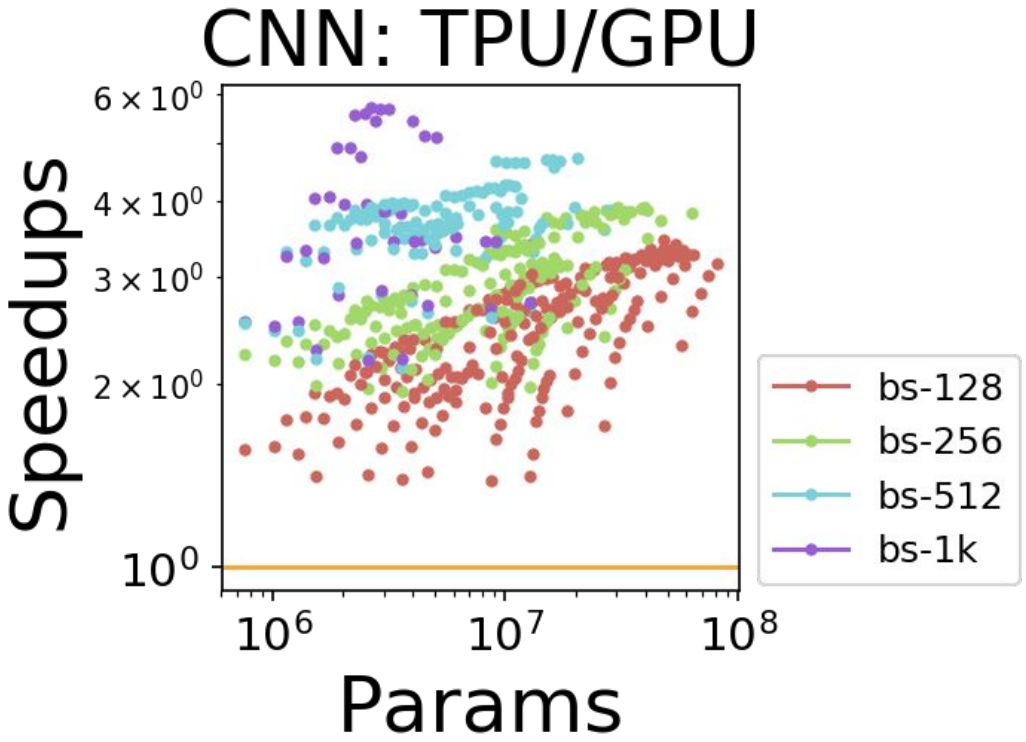
Hardware Platforms

1.44x

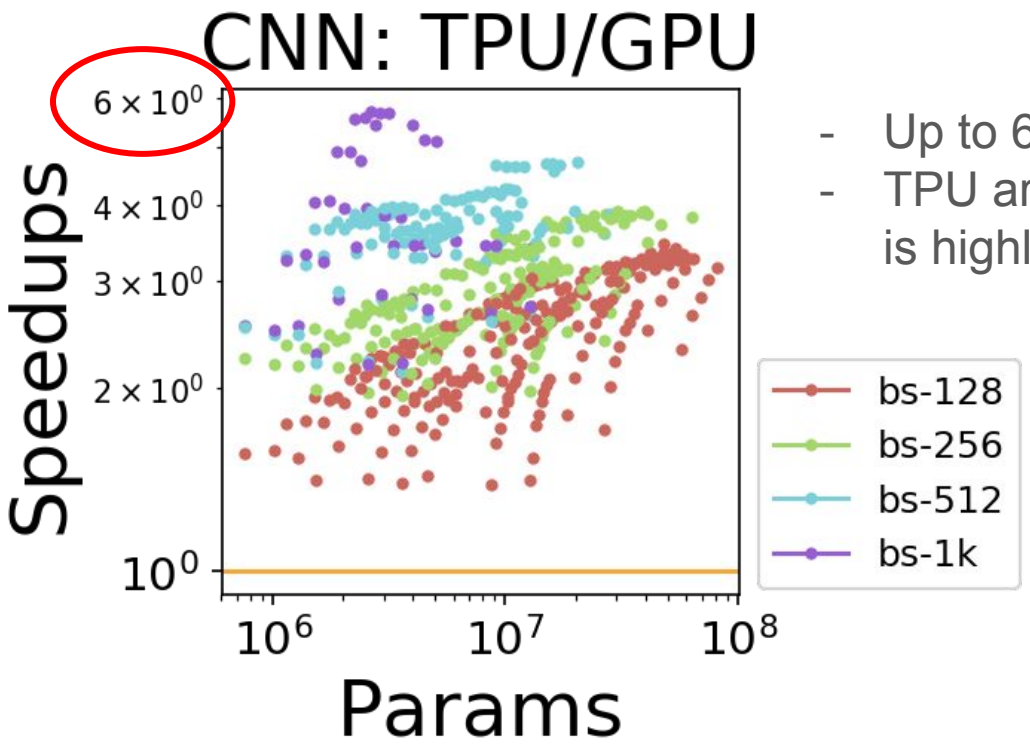
Platform	*Unit*	*Version*	*Mem Type*	*Mem (GB)*	*Mem Bdw (GB/s)*	Peak *FLOPS*
GPU (DGX-1)	1 Pkg	V100 (SXM2)	HBM2	16	900	125T
TPU	1 Board (8 cores)	v2	HBM	8	2400	180T

300 GB/s per core

CNN TPU/GPU Speedup colored with **Batch Size**

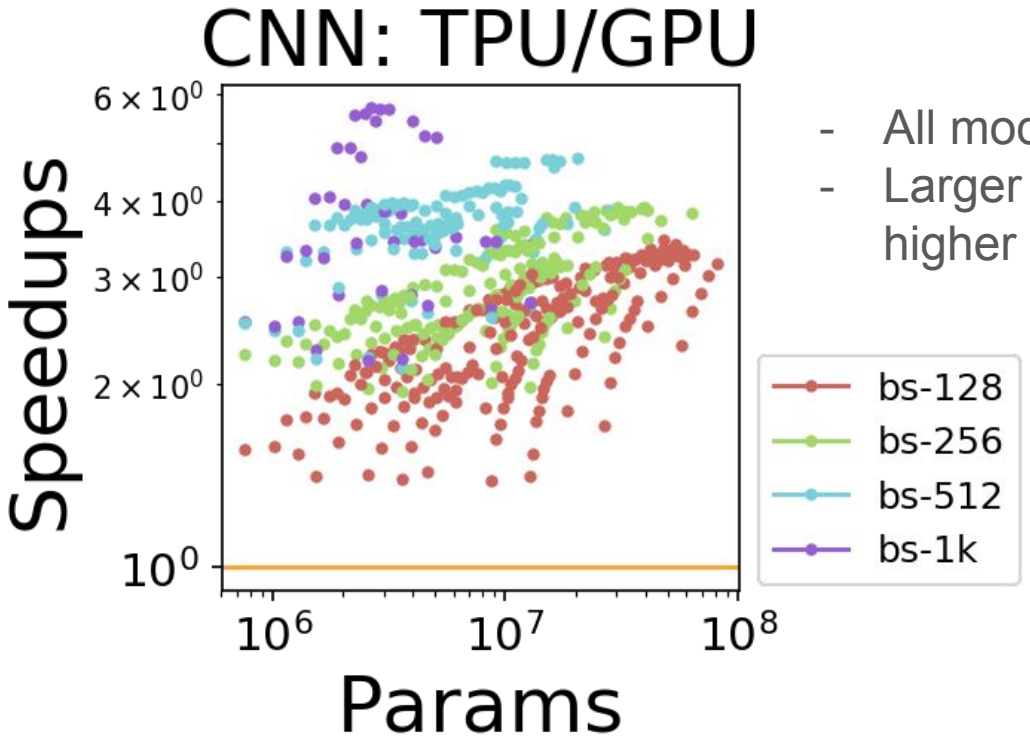


CNN TPU/GPU Speedup colored with **Batch Size**



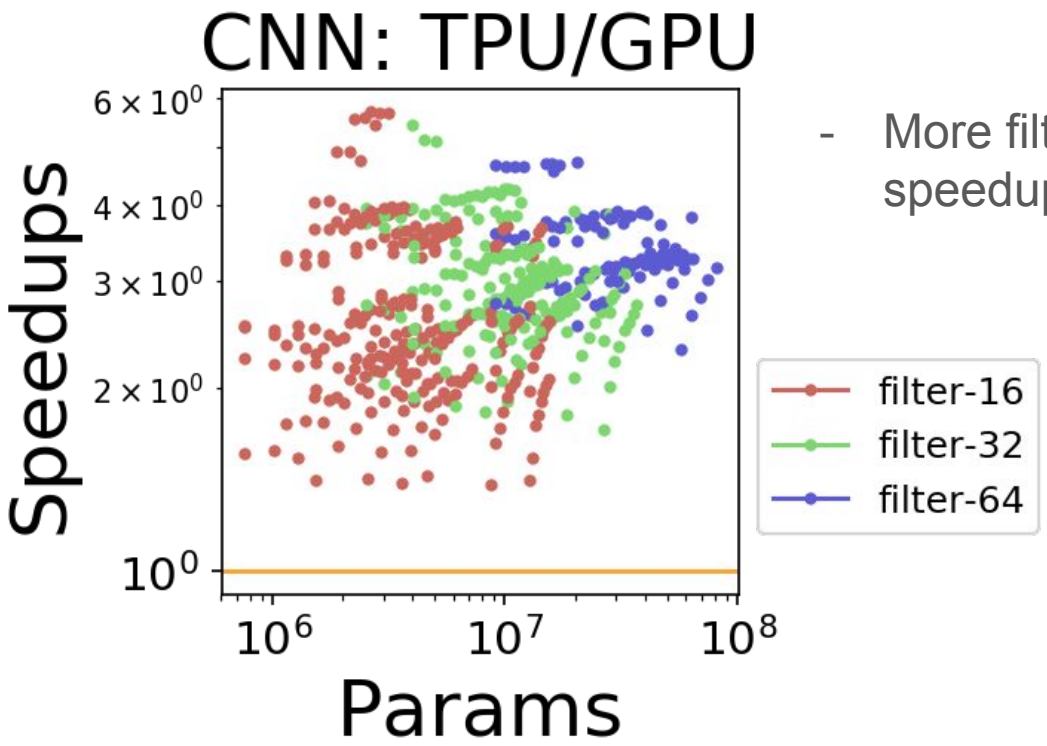
- Up to 6x speedup
- TPU architecture and software is highly optimized for CNNs

CNN TPU/GPU Speedup colored with **Batch Size**



- All models runs faster on TPU.
- Larger batch sizes lead to higher speedups.

CNN TPU/GPU Speedup colored with **Filters**



- More filters have higher speedup lower bounds

Conclusion

- Parameterized methodology: ParaDnn + a set of analysis methods
- Single platform analysis: TPU v2
- Homogenous platform comparison: TPU v2 vs v3
- Heterogeneous platform comparison: TPU vs GPU

Limitations of this Work

- Does not include:
  - Inference
  - Multi-node system: multi-GPU, or TPU pods
  - Accuracy, convergence
  - Cloud overhead
- Tractability
  - Limit the range of hyperparameters and datasets
    - Small batch sizes (<16) and large batch sizes (> 2k) are not studied
    - Synthetic datasets do not include data infeed overhead
  - Iterations of TPU loop is 100. Larger numbers can slightly increase the performance.

ParaDnn

Available: github.com/Emma926/paradnn

Questions?

HARVARD
John A. Paulson
School of Engineering
and Applied Sciences