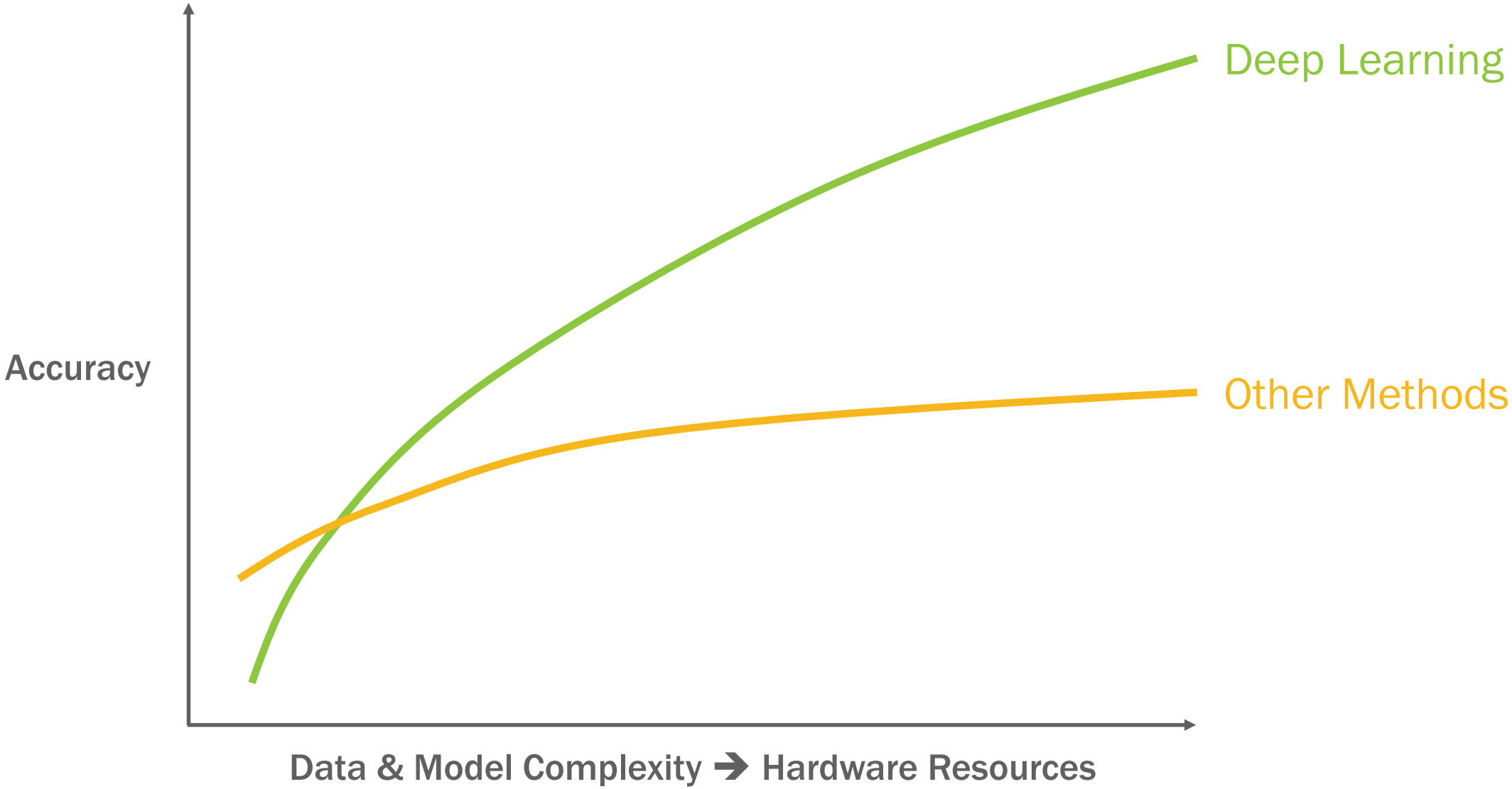
facebook

AI System Co-Design: How to Balance Performance & Flexibility

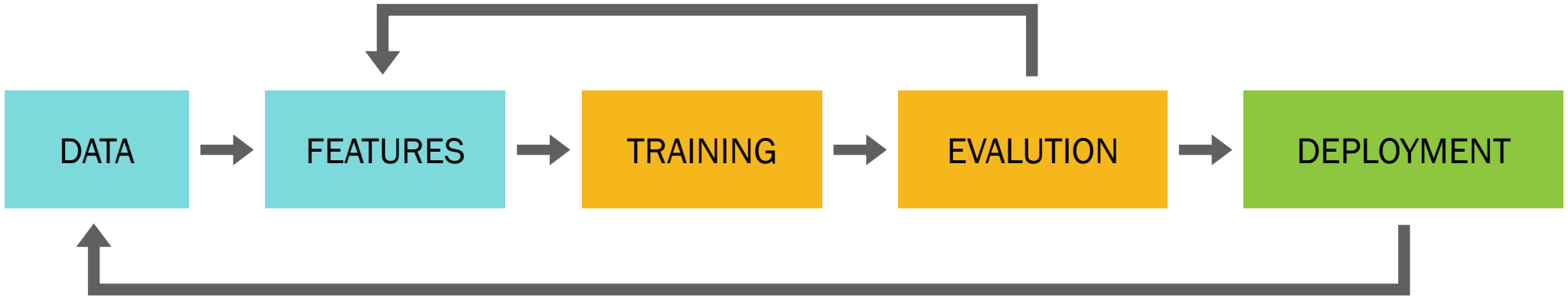
Misha Smelyanskiy
Director, AI Systems Co-Design — Facebook

AI Hardware Summit, September 17, 2019

Deep Learning is Unique



ML Growth and Scale at Facebook



ML data growth

- Usage in **2018**: 30%
- Usage **today**: 50%
- Growth in **one year**: 3X

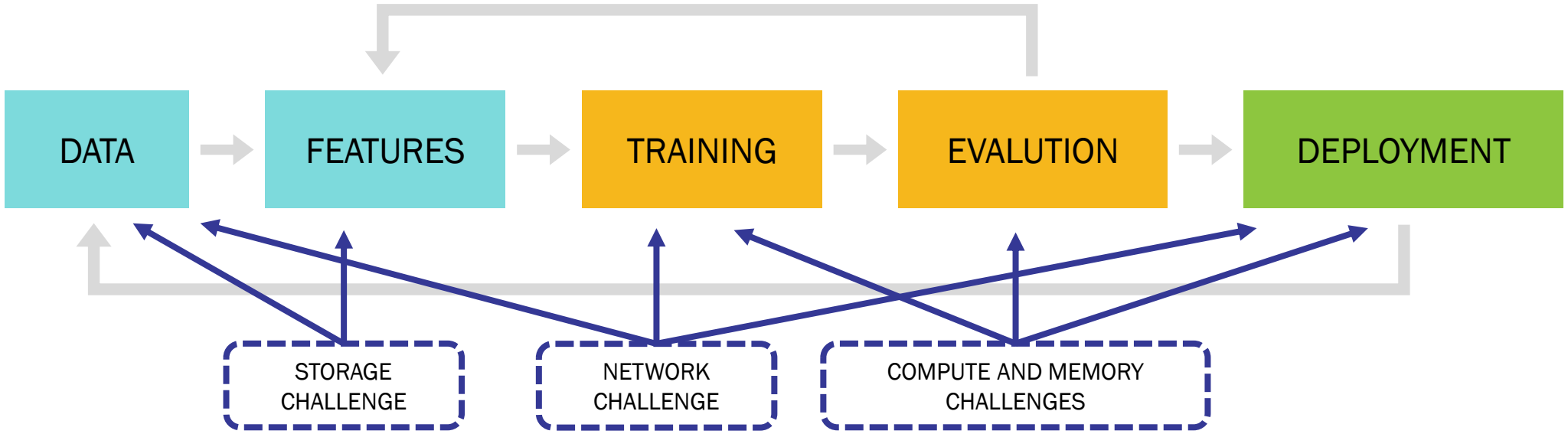
1-year Training growth

- Ranking engineers: 2X
- Workflows trained: 3X
- Compute consumed: 3X

Inference Scale per Day

- # of predictions: 200T
- # of translations: 6.5B
- Fake accounts removed: 99%

Infrastructure Challenge



- Strains compute, memory, storage, and network
- Speed of innovation requires
  - High performance
  - HW and SW flexibility and developer efficiency

Flexibility Challenge

On SW side

- DL workloads, their characteristics and diversity evolve quickly
- Perfect storm of HW requirements: compute, mem BW & capacity, fabric BW
- ML engineers need to experiment fast with new operators, models, algorithms

On HW side

- Ease of platform serviceability
- Ability to accommodate different HW: compute, memory, interconnect
- Ease of scaling out

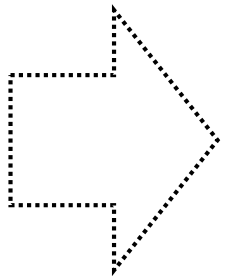
➔ **Motivates need for SW/HW co-design of ML platform**

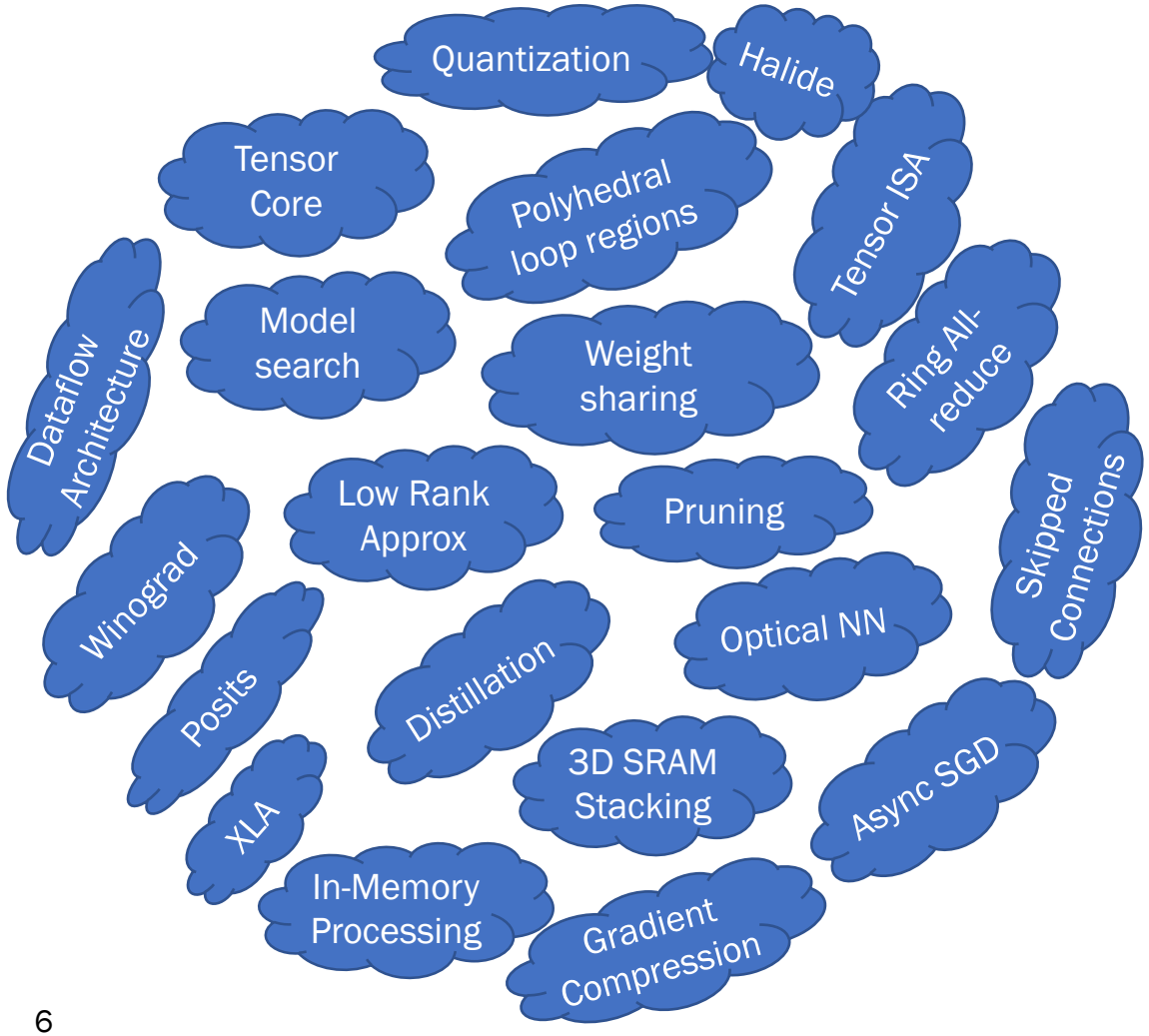
HW/SW Co-Design

"**Co-design** refers to the **concurrent** design and optimization of several aspects of the system, including **hardware** and **software**, to achieve a set **target** for a given system metric, such as **throughput**, **latency**, **power**, **size**, or a combination thereof.", *Modeling, validation, and co-design of IBM Blue Gene/Q: Tools and examples"*

Application
Algorithm
Language
Compiler

System
Arch & uArch
Circuits
Device Physics



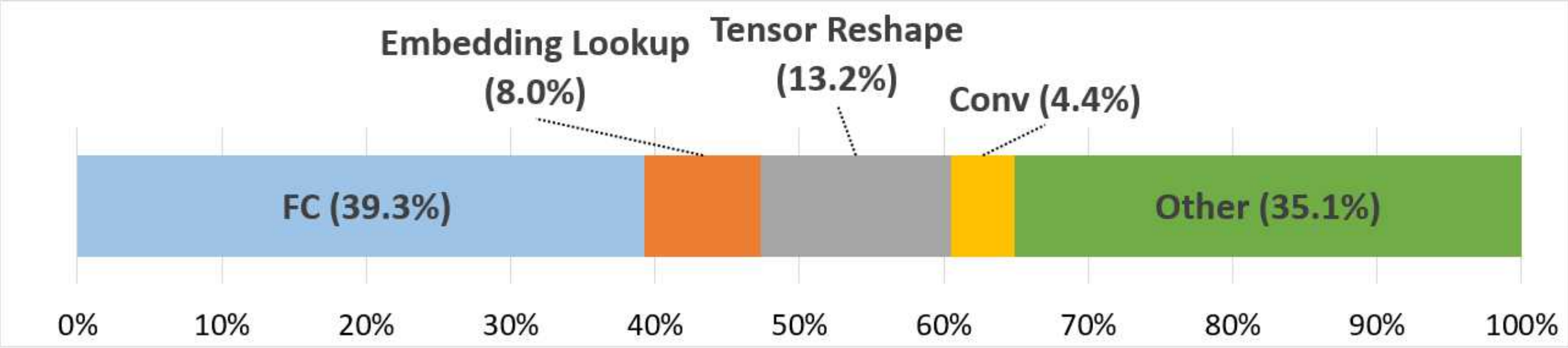


Main AI Services @ Facebook

- Ranking and recommendation
  - news feed, and search
- Computer vision
  - image classification, object detection, and video understanding
- Language
  - translation, content understanding

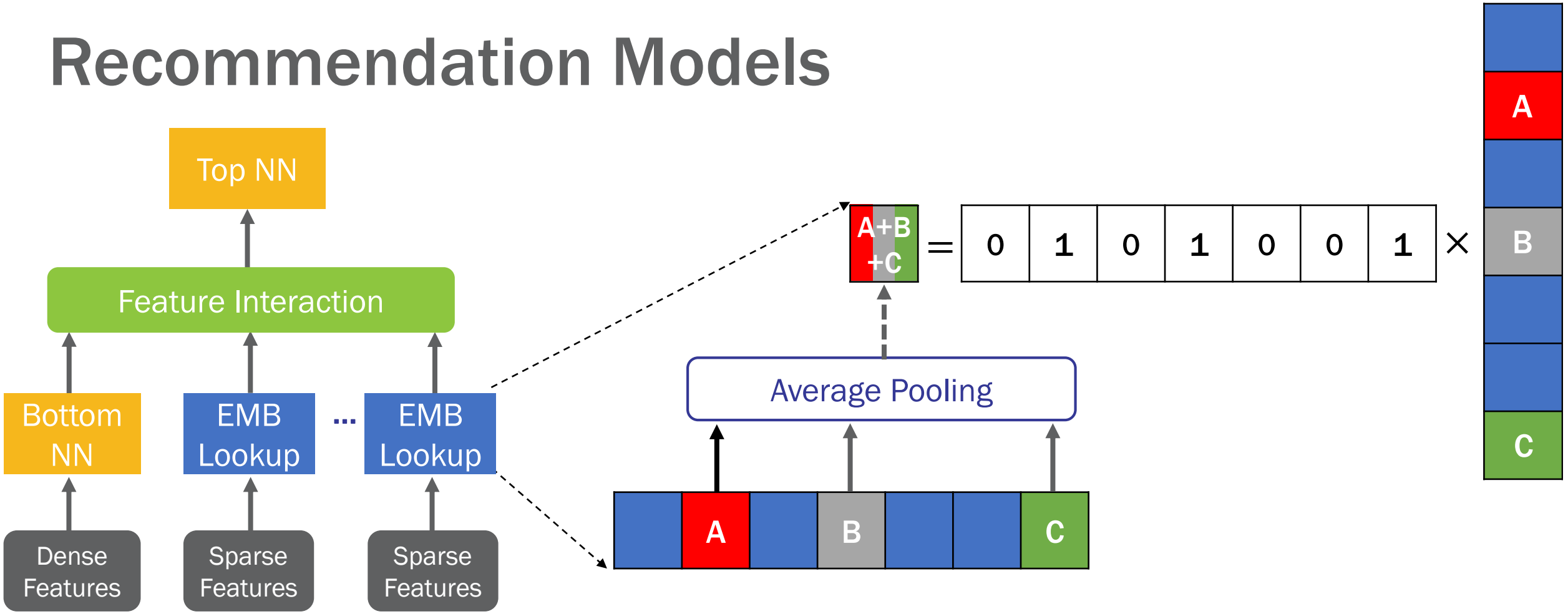
- Recommendation models are among most important models

It’s not all about GEMMs



- Only ~40% is spent in GEMMs in FB production
- Should not over-design hardware for GEMMs and convolutions
- Flexibility requires general-purpose support, but balancing is hard

Recommendation Models



- DL recommendation models help user choose small set of items out of many
- Embedding look-up is sparse matrix by dense matrix multiplication

Inference Model Characteristics

Category	Model Types	Model Size (# params)	Maximum Live Activations	Op Intensity (w.r.t. weights)	Op Intensity (w.r.t. act & weights)
Recommendation	FCs	1-10M	> 10K	20-200	20-200
	Embeddings	>10 Billion	> 10K	1-2	1-2

High memory capacity (>10s of GBs) and low op intensity

➔ HBMs are too small, NVMs are too slow

➔Need high capacity, high bandwidth memory

Accesses to some tables exhibit locality

➔cache is more flexible than SW-controlled scratchpad

Pooling compresses multiple embedding vectors into a single one

➔Interesting use case for in-memory processing

Open-sourced as a deep learning recommendation model benchmark

Inference Model Characteristics, cont.

Category	Model Types	Model Size (# params)	Maximum Live Activations	Op Intensity (w.r.t. weights)	Op Intensity (w.r.t. act & weights)
Computer Vision	ResNeXt101-32x4-48	43-829M	2-29M	Avg. 380 Min. 100	Avg. 188 Min. 28
	ResNeXt3D-101	21M	58M	Avg. 22K Min. 2K	Avg. 172 Min. 6
Language	GRU/LSTM/Transforme r	10M-1B	>100K	2-60	2-60

Model and activations are big, operational intensity is small

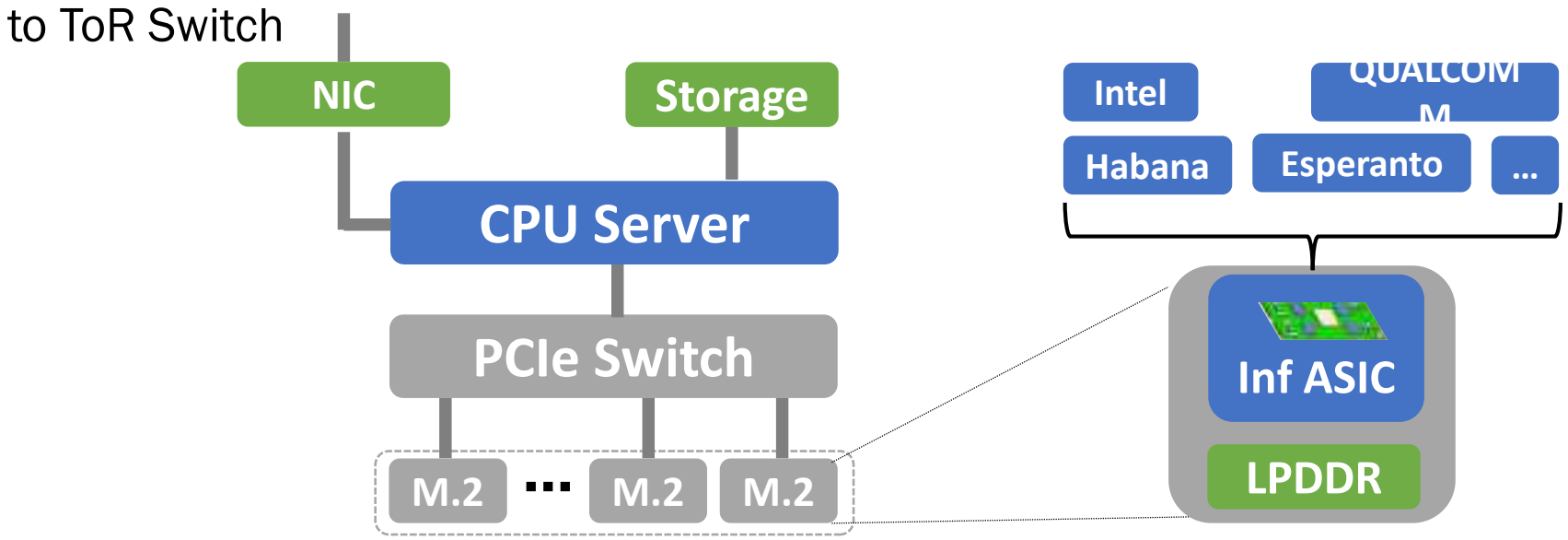
➔Larger on-chip memory helps and gives compiler more flexibility

Skinny GEMMs due to depth- and group-wise convolutions (CV), small batch / beam (language)

➔ Need many smaller tensor units (rather then few big ones)

➔ Need higher on-chip BW

Yosemite V2 Inference Platform



Where does flexibility come from?

➔ Scale-up compute, mem/SRAM capacity & BW: tightly couple via PCIe switch

➔ Aggregate 100 TOPs of INT8, 100 MBs of SRAM, and 100s of GB/s memory bandwidth

➔ Common M.2 module, common compute and memory requirements for vendors

➔ Community-driven approach to programmability via GLOW compiler

Training Workload Characteristics

Application	Memory Capacity	Memory Bandwidth	Interconnect Bandwidth	Compute Rate
Recommendation	Very high	High	High	High
Computer Vision	High	Low	Low	Very High
Language	High	Medium	High	High

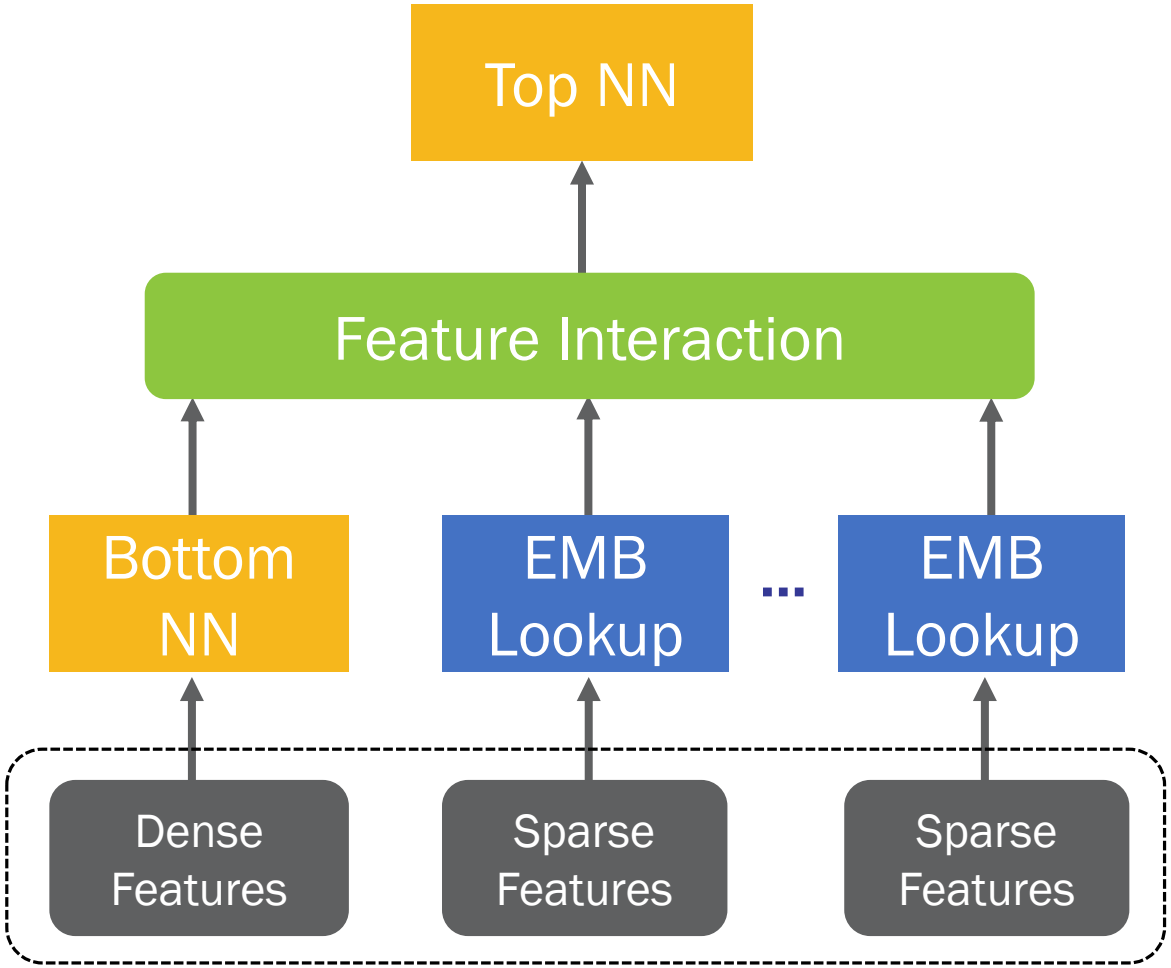
Training is highly experimental: ML engineers expect **speed** & **flexibility**

➔ Experiment with new operators and different models, to improve model quality

➔ Experiment with new numerical algorithms, faster training / better model quality

➔ Ease of experimentation, allow for fast prototyping and evaluation

Recommendation Systems Example



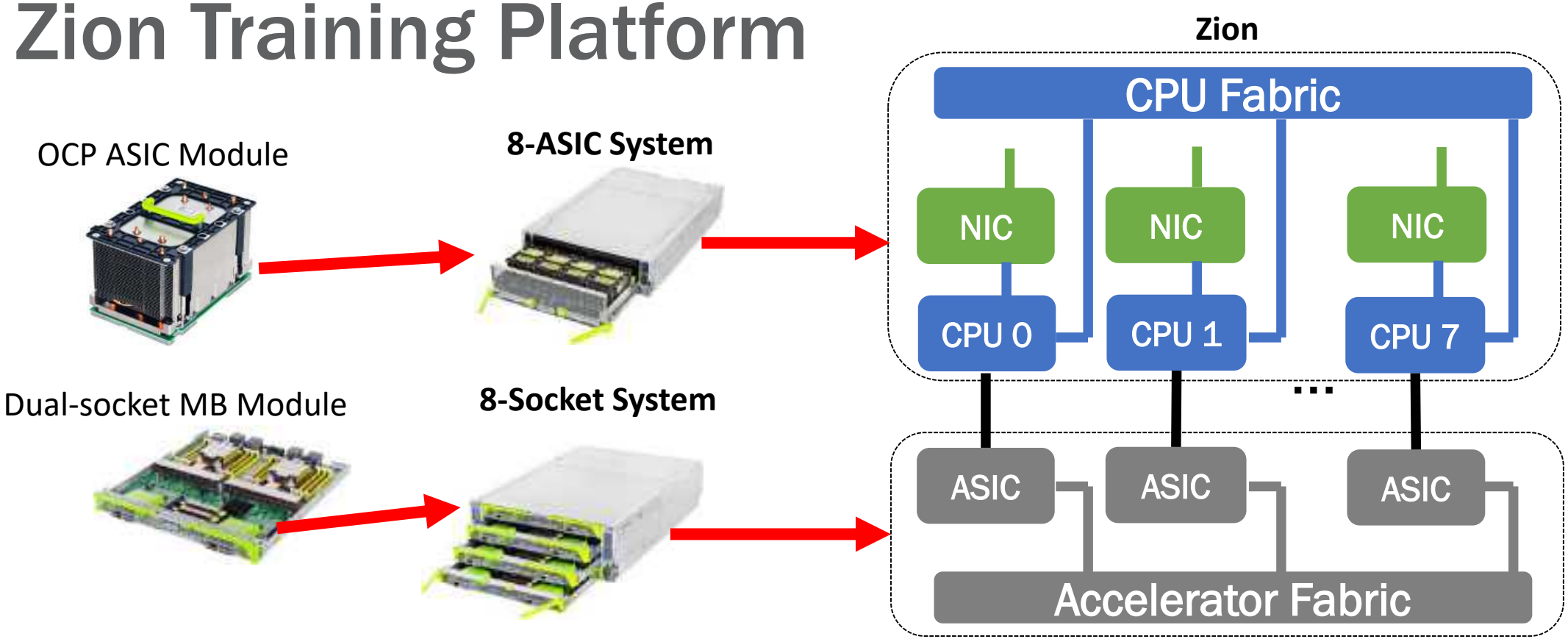
Compute & communication

Communication

Mem capacity & bandwidth

Inputs from Network

Zion Training Platform



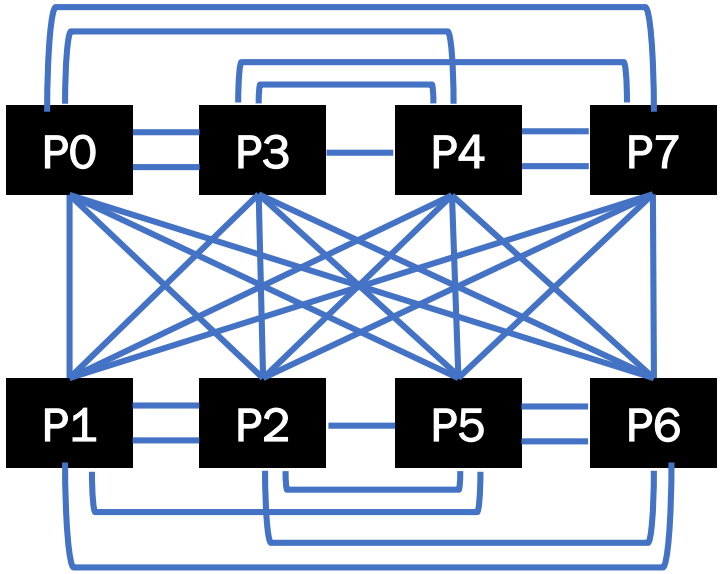
- Unified 100s TFLOPs of BFLOAT16, high capacity DDR, high bandwidth HBM
- High-performance, disaggregated CPU and ASIC fabric
- Scale-out capability via host NIC, P2P, RDMA, …

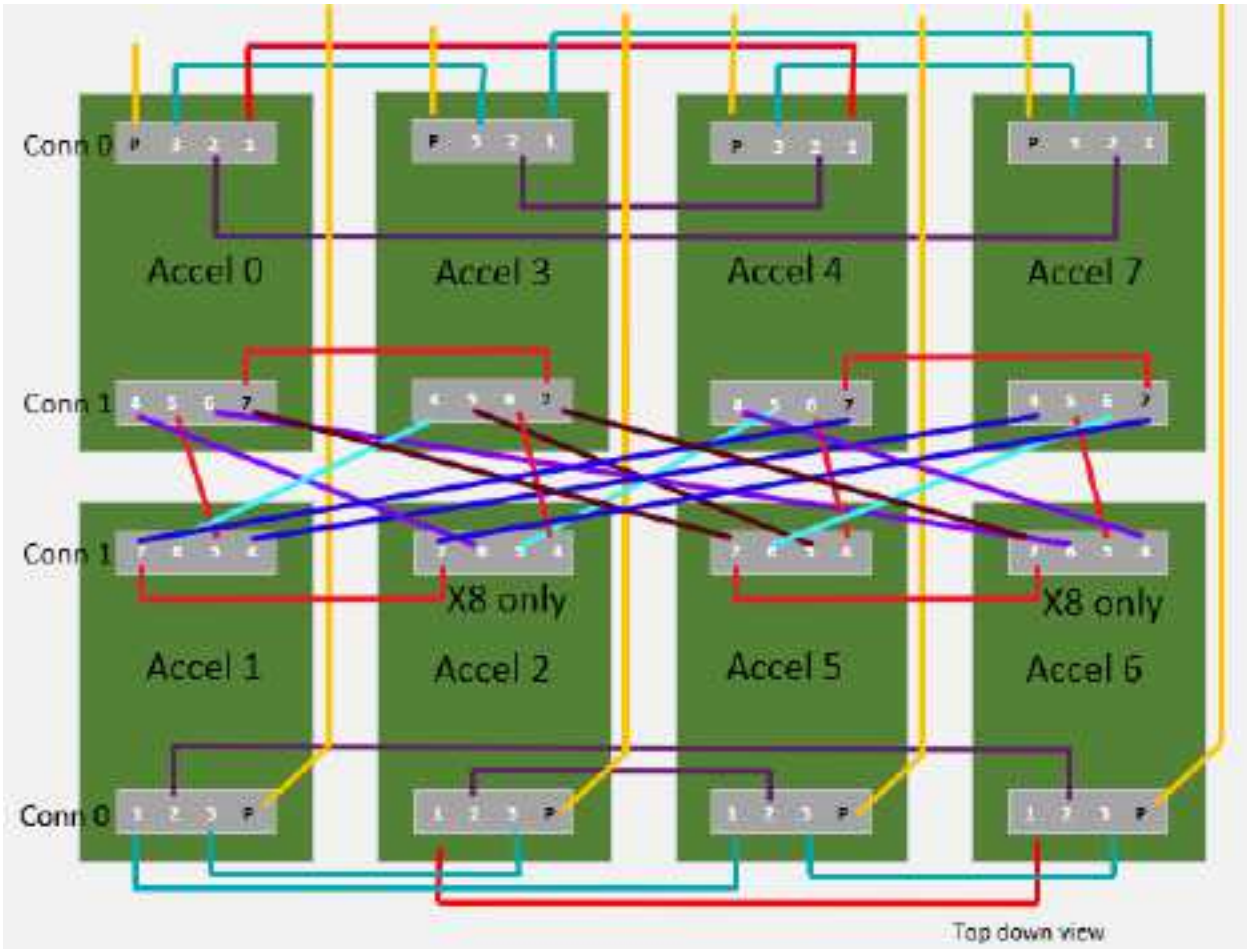
Zion: Accelerator Flexibility

- Challenge: which accelerator do we use?
  - Very large number of accelerators
  - Limited resource to enable multiple systems
- Solution: OCP Accelerator Module(OAM)
  - Facebook led efforts
  - Define vendor-agnostic common form factor

Zion: Interconnect Topology Flexibility

- Challenge: vendors support different topologies: FC, AFC, HCM, …
- Solution: superset physical topology





Zion: Software Flexibility

User can gradually increase SW complexity (and performance)

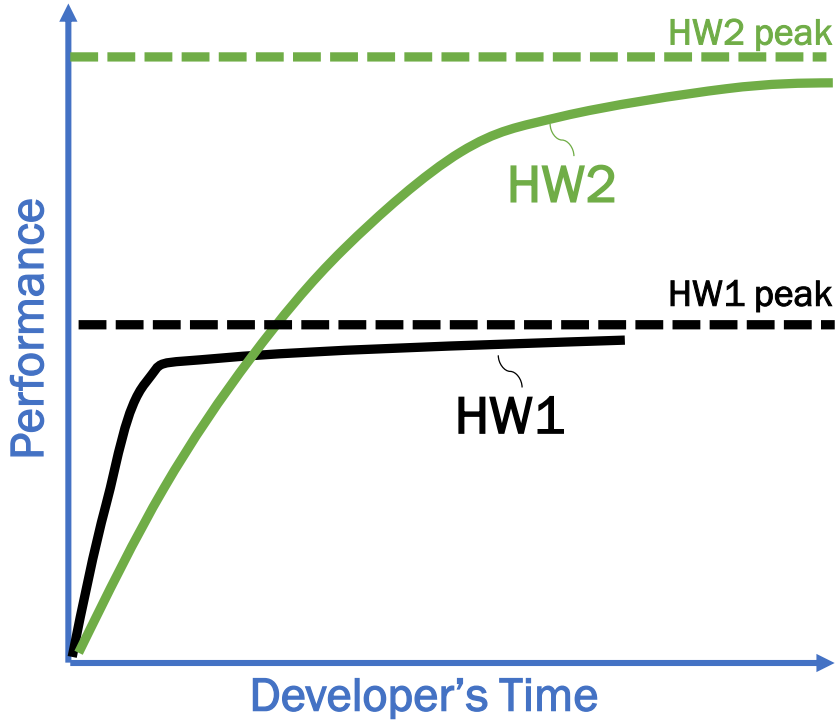
1. CPU-only
2. CPU for embeddings + Accelerators for MLP
3. Use Accelerator HBM for embeddings as well
   - Challenge: table accesses have different frequencies
   - Benefits from run-time profile driven table partitioning
4. Distributed training

Creates continuum of dev efficiency vs performance tradeoffs

More details about Zion were presented in HotChips'19

Parting Words on Programmability

"If the system is difficult to program, [you] won't have software. If you don't have software, you don't have an accelerator", *Krste Asanović, Accelerating AI: Past, Present, and Future.*



Convolutional Capsules Microbenchmark			
Compiler	**Device**	**Compilation**	**Execution**
gcc	x86 (1 core)	500ms	64.3ms
gcc -fopenmp	x86 (6 cores)	500ms	11.7ms
PlaidML	GTX1080	560ms	604ms
Tensor Comp.	GTX1080	3.2s	225ms
Tensor Comp.	GTX1080	64s	18.3ms
Tensor Comp.	GTX1080	1002s	1.8ms
CUDA	GTX1080	48h	1.9ms

Source: Machine Learning Systems are Stuck in a Rut

- DL models are programs
- Specialization can be at odds with programmability (scalar < vector < 2D tensor < 3D tensor)
- Companies building ML HW need to think about SW at scale

Summary

- Moore's Law slow-down requires specialization and co-design
- ML is moving fast : performance and SW / HW flexibility are important
- Facebook AI hardware is an example of how to target speed and flexibility
  - Announced at 2019 OCP (Open Compute Project) Global Summit
  - Many designs and specs publicly released through the OCP
- Will continue improving through SW / HW co-design, but cannot do this alone
- More than enough problems for everyone to tackle → Let us work together!

Questions?