MTIA: Meta's First Generation of In-House AI Accelerator

Meta Platforms Inc.

ISCA 2023

Presentation: Constant Park

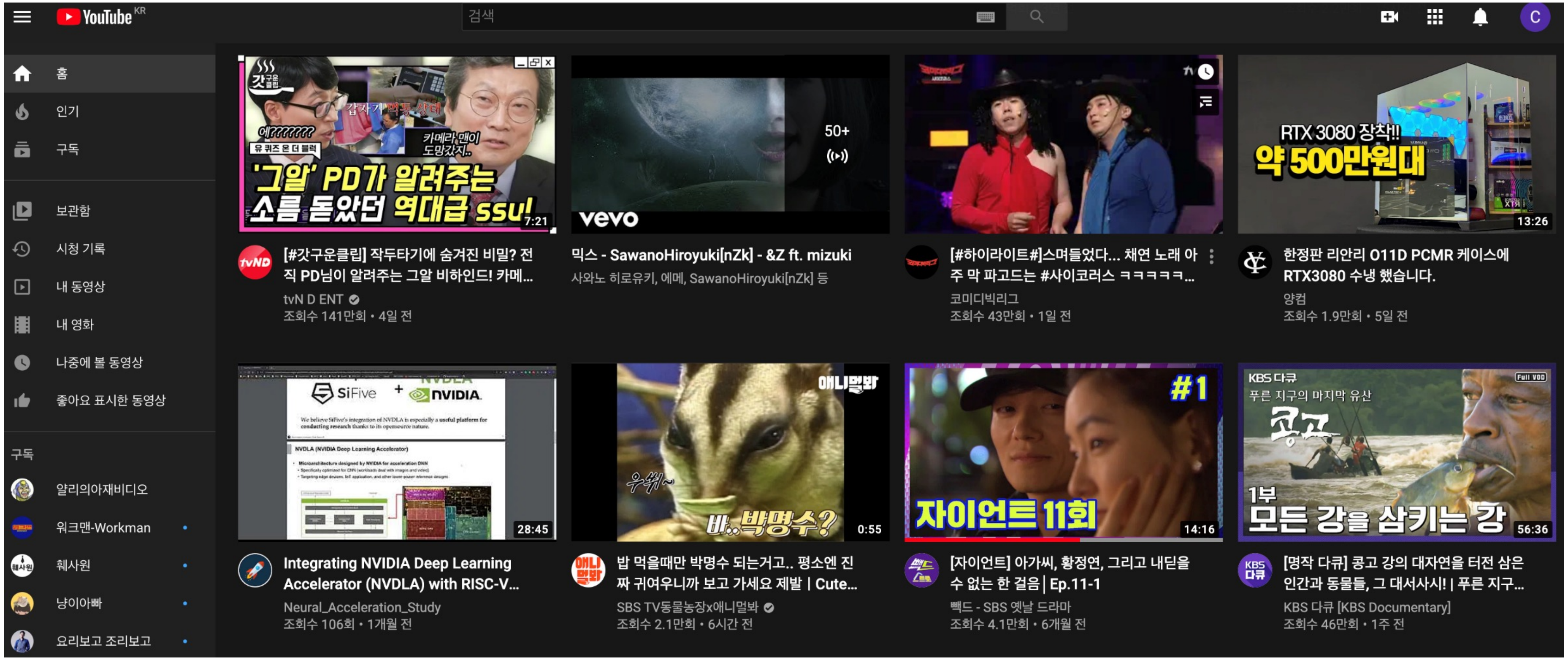
(sonicstage12@naver.com)

Contents

*This presentation is based on ISCA paper (MTIA: First Generation Silicon Targeting Meta's Recommendation Systems/TensorDIMM) and Scale metarial (https://youtu.be/kO4a75fKDF4)

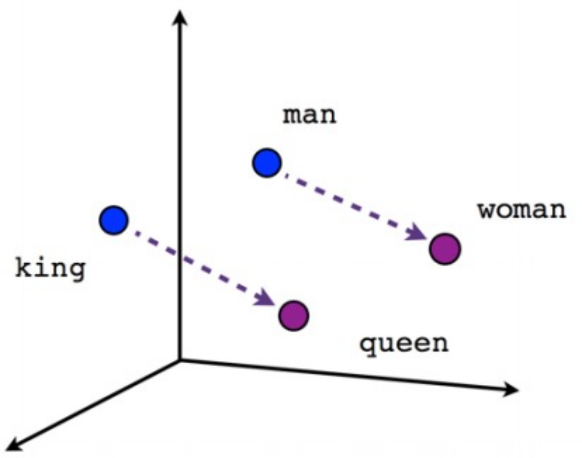
Recommendation System: Overview

- **Personalized recommendation for contests**
  - **Sparse embedding layers** are a bottleneck

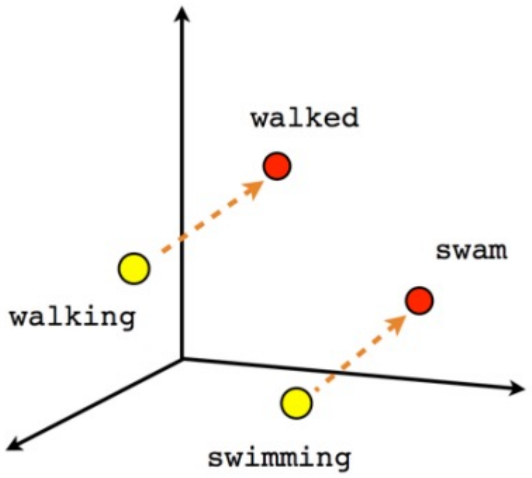


Recommendation System: Embedding

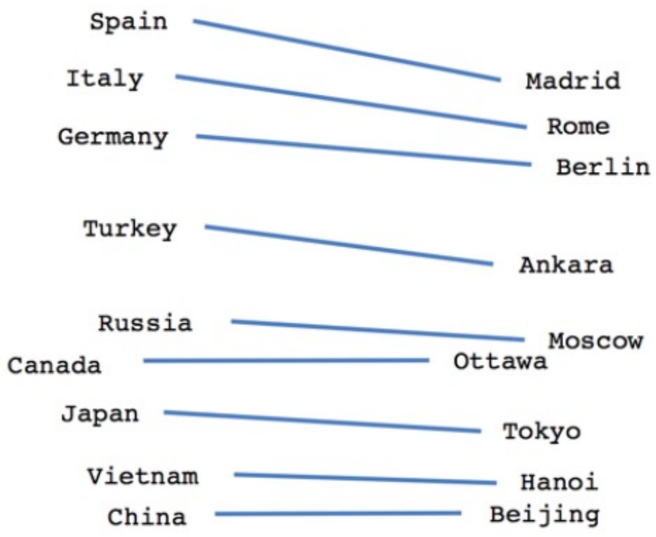
- **Words are mapped to vectors of real numbers**
  - Word embedding, Neural item embedding for Collaborative filtering



Male-Female



Verb tense

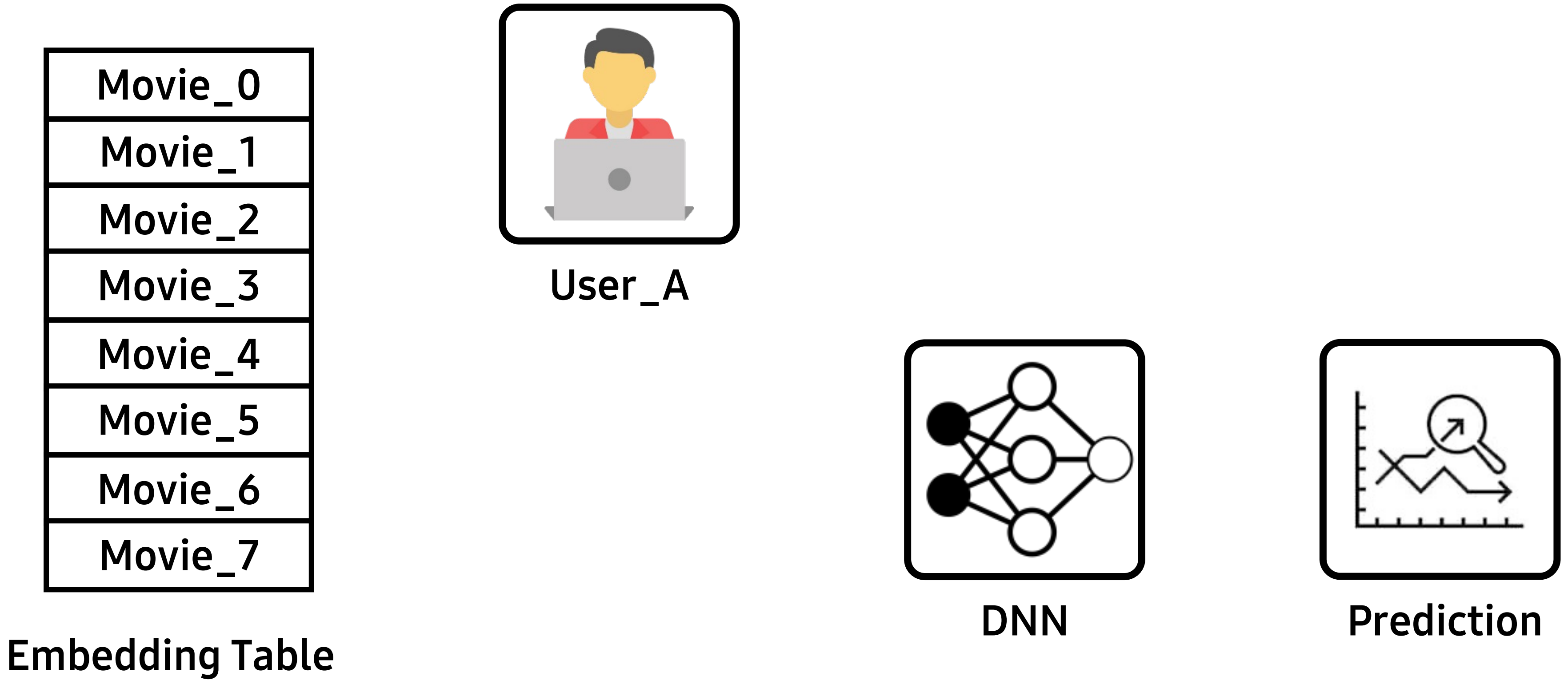


Country-Capital

*https://towardsdatascience.com/mapping-word-embeddings-with-word2vec-99a799dc9695

Recommendation System: Example

- Goal: Predicting preference of user-item pair (Movie)



Recommendation System: Example

- Goal: Predicting preference of user-item pair (Movie)

Harry Porter

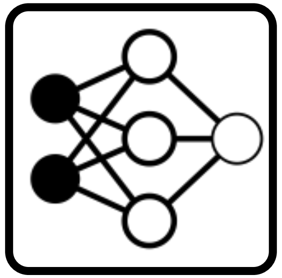
Batman

Ironman

Movie_0
Movie_1
Movie_2
Movie_3
Movie_4
Movie_5
Movie_6
Movie_7

Embedding Table

User_A

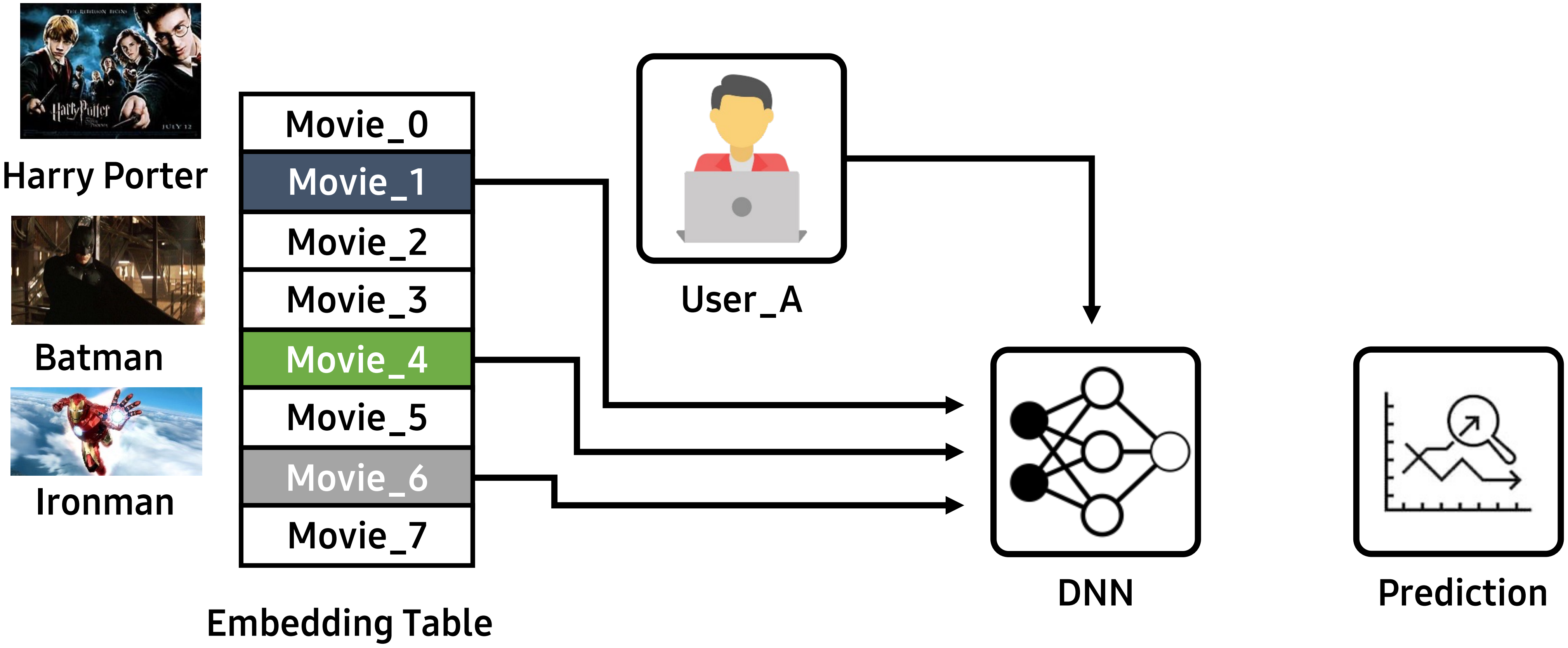


DNN

Prediction

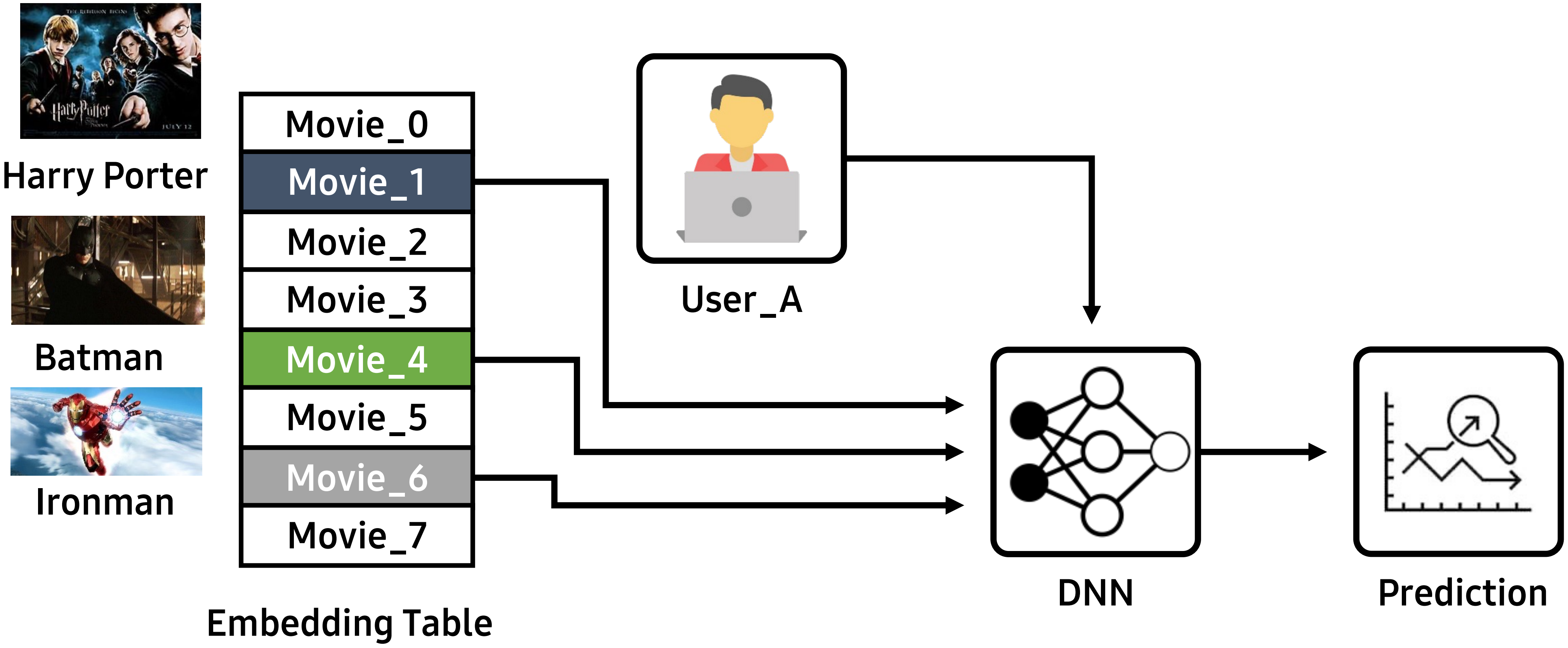
Recommendation System: Example

- Goal: Predicting preference of user-item pair (Movie)



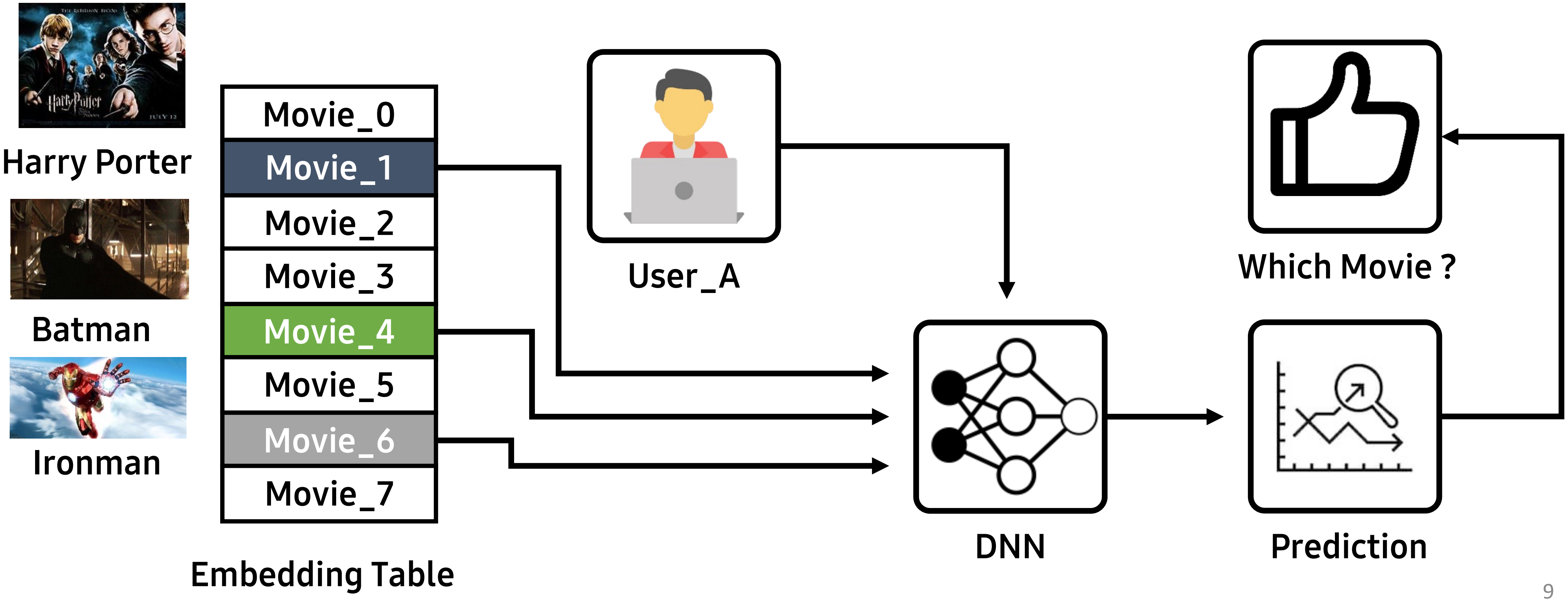
Recommendation System: Example

- Goal: Predicting preference of user-item pair (Movie)



Recommendation System: Example

- Goal: Predicting preference of user-item pair (Movie)



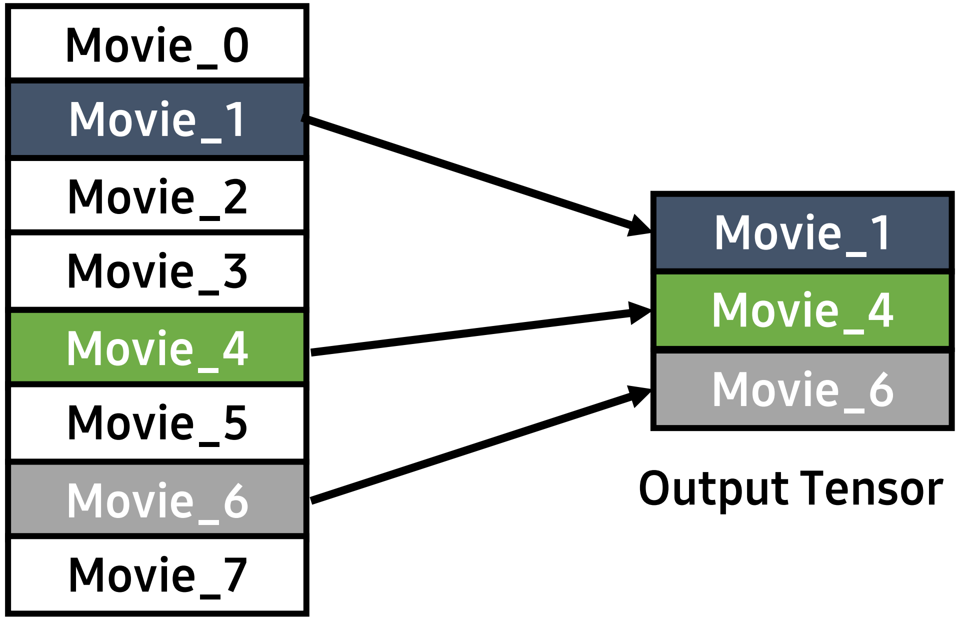
Recommendation System: Example

- Gather: Copying embeddings into contiguous address space

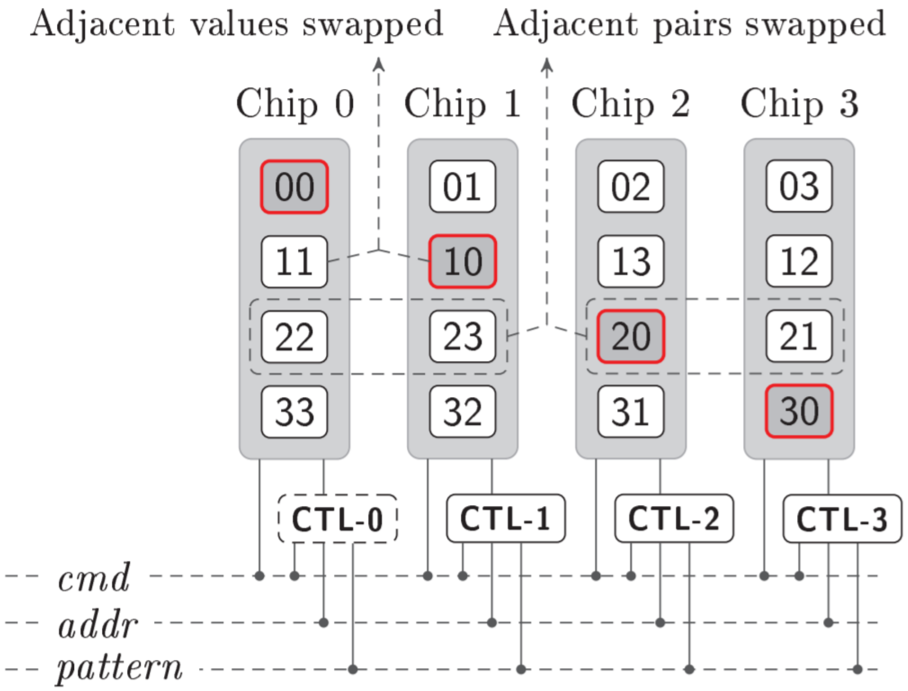
Harry Porter

Batman

Ironman



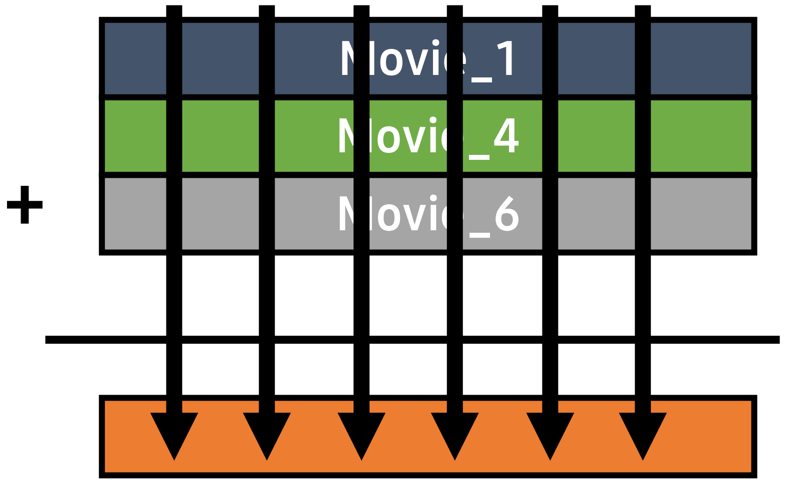
Embedding Table



*Rank-Level Parallelism in DRAM, IEEE Transaction on Computers, 2017

Recommendation System: Example

- Reduction: Multiple embeddings, element-wise ADD/MUL



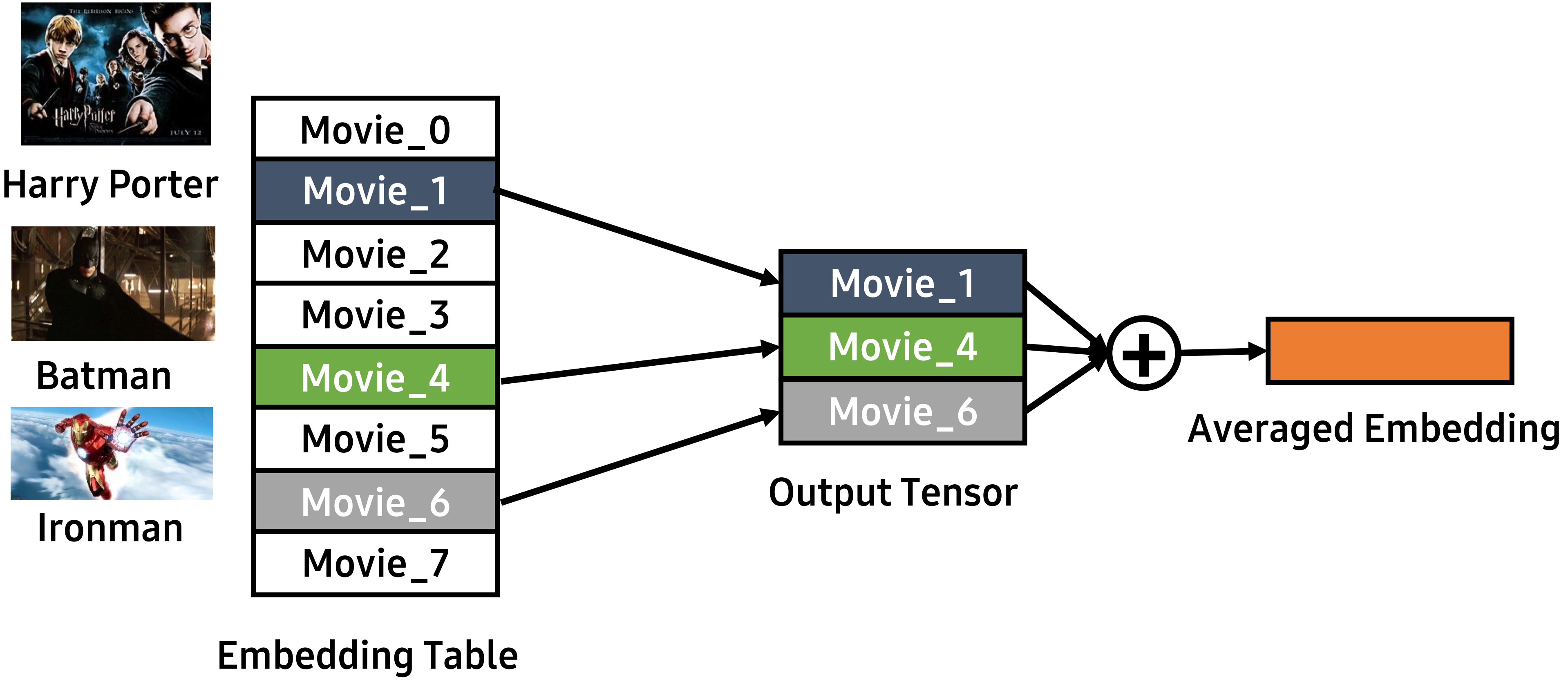
Harry Porter

Batman

Ironman

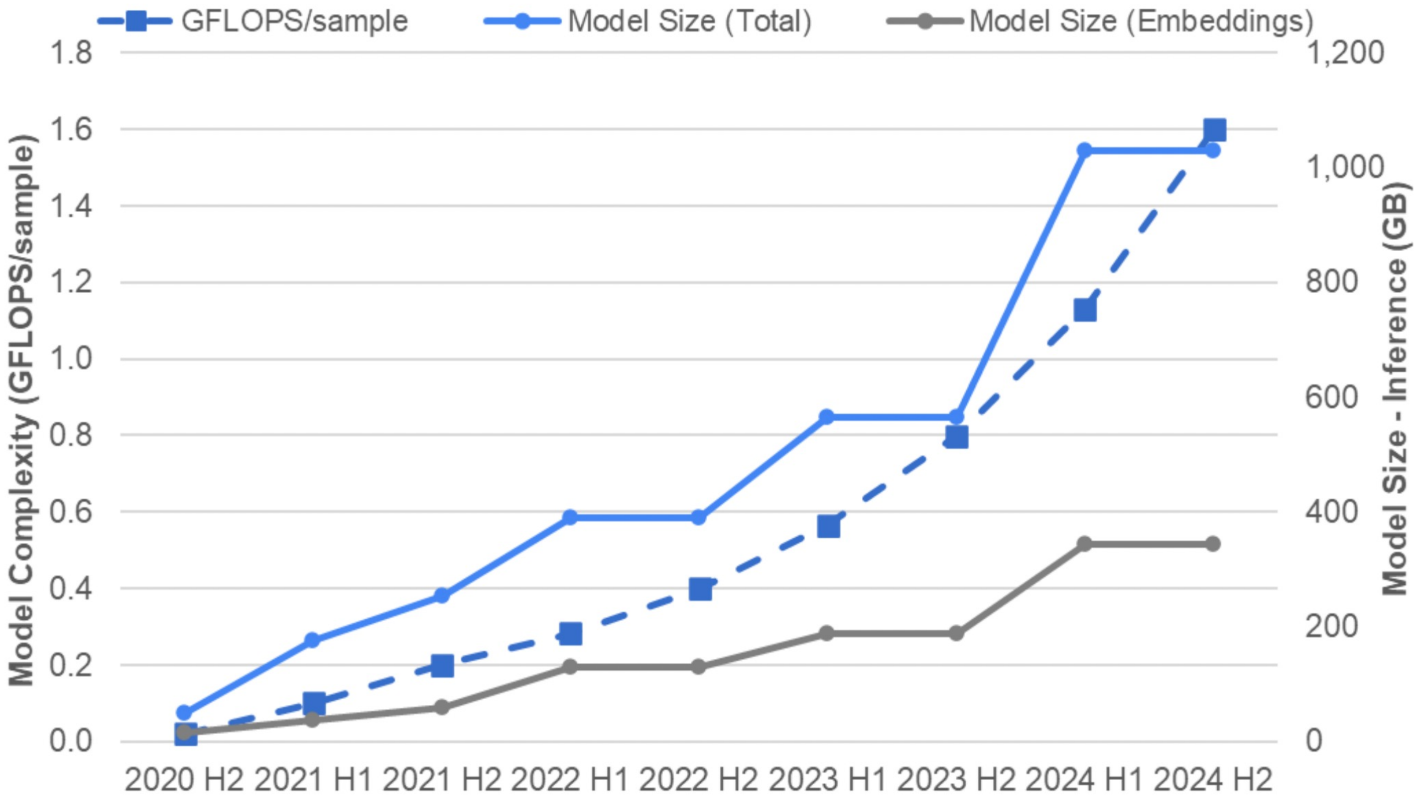
Recommendation System: Example

- **Gather/Reduction operation in embedding layer**
  - This is memory-bandwidth sensitive operation



Motivation: Inference Workloads

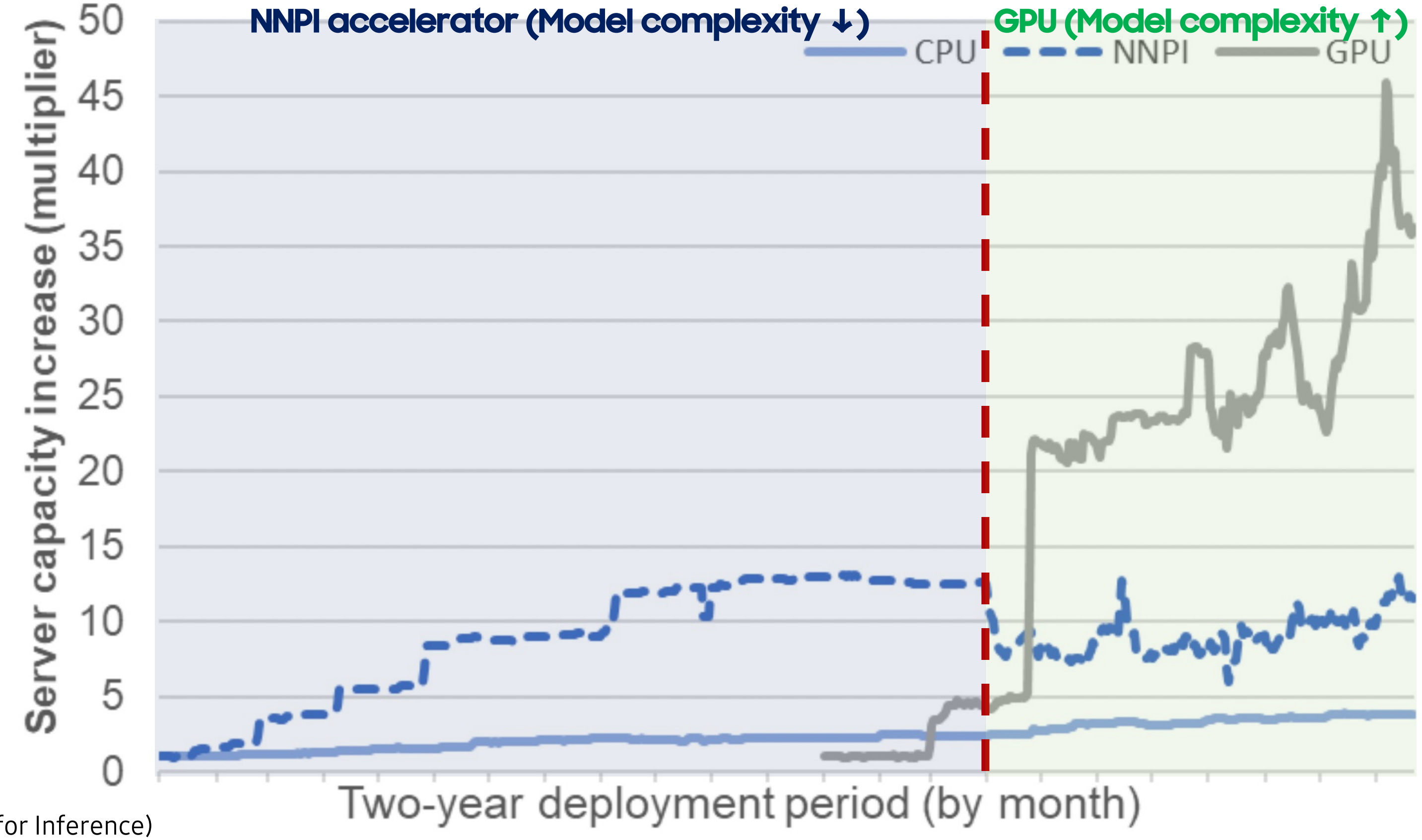
- **Trends of inference models at Meta's service workload**
  - Significant growths in **model size (GB)** and **complexity (GFLOPS)**



*DLRM (Deep Learning Recommendation Model)

Motivation: Inference Server Demand

- Accelerator to meet model demands and efficiency requirements
  - GPUs are not designed for inference (Low efficiency w/ SW optimizations)

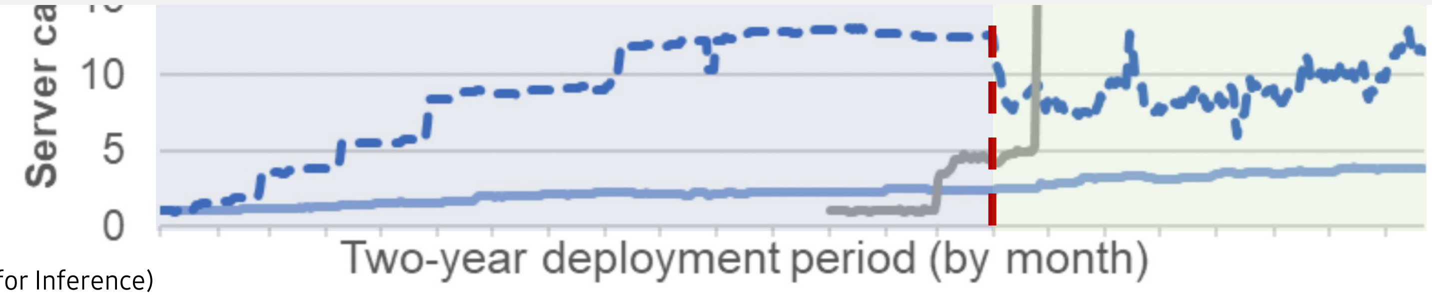


*NNPI (Intel Nervana Neural-Net Processors for Inference)

Motivation: Inference Server Demand

- Accelerator to meet model demands and efficiency requirements
  - GPUs are not designed for inference (Low efficiency w/ SW optimizations)

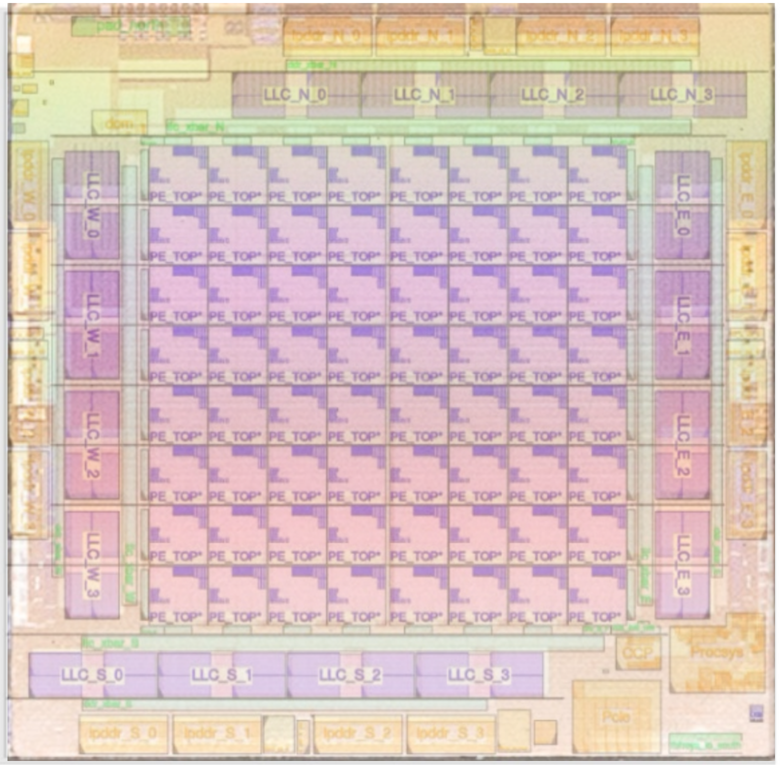
Architecture should also provide enough generality and programmability, to support future versions of these workloads and potentially other types of NN models.



*NNPI (Intel Nervana Neural-Net Processors for Inference)

Architecture: Specification

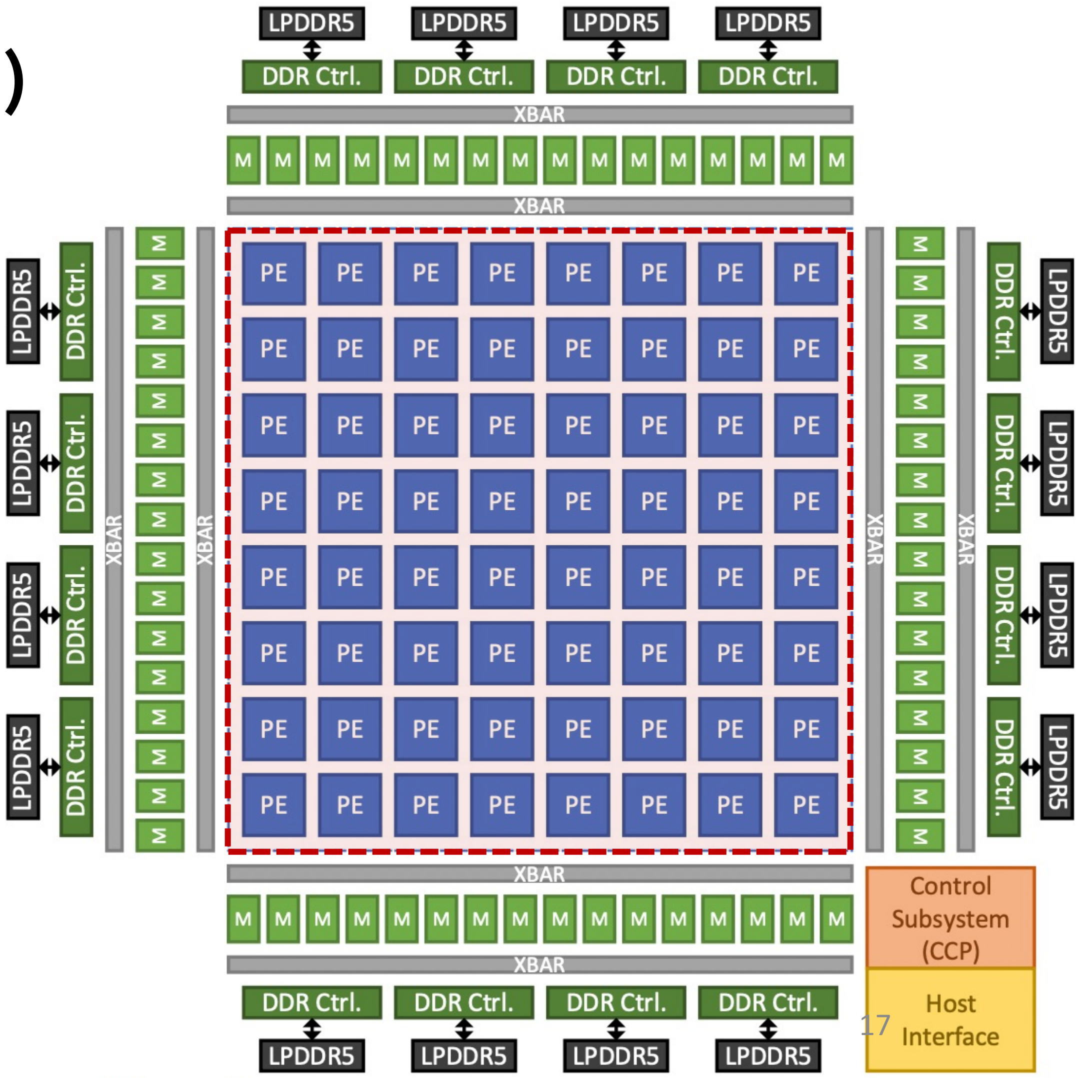
- MITA features and parameters



Technology	TSMC 7nm
Frequency	800MHz (Up to 1.1GHz)
Dimensions	19.34*19.1mm (~2,800 Pins)
TDP	25W
Peak Perf. (GEMM)	102.4TOPS (INT8), 51.2 (FP16)
Memory Bandwidth	800GB/s (SRAM), 176GB/s (DRAM)
Memory Capacity	128MB (SRAM), LPDDR5 (64GB)

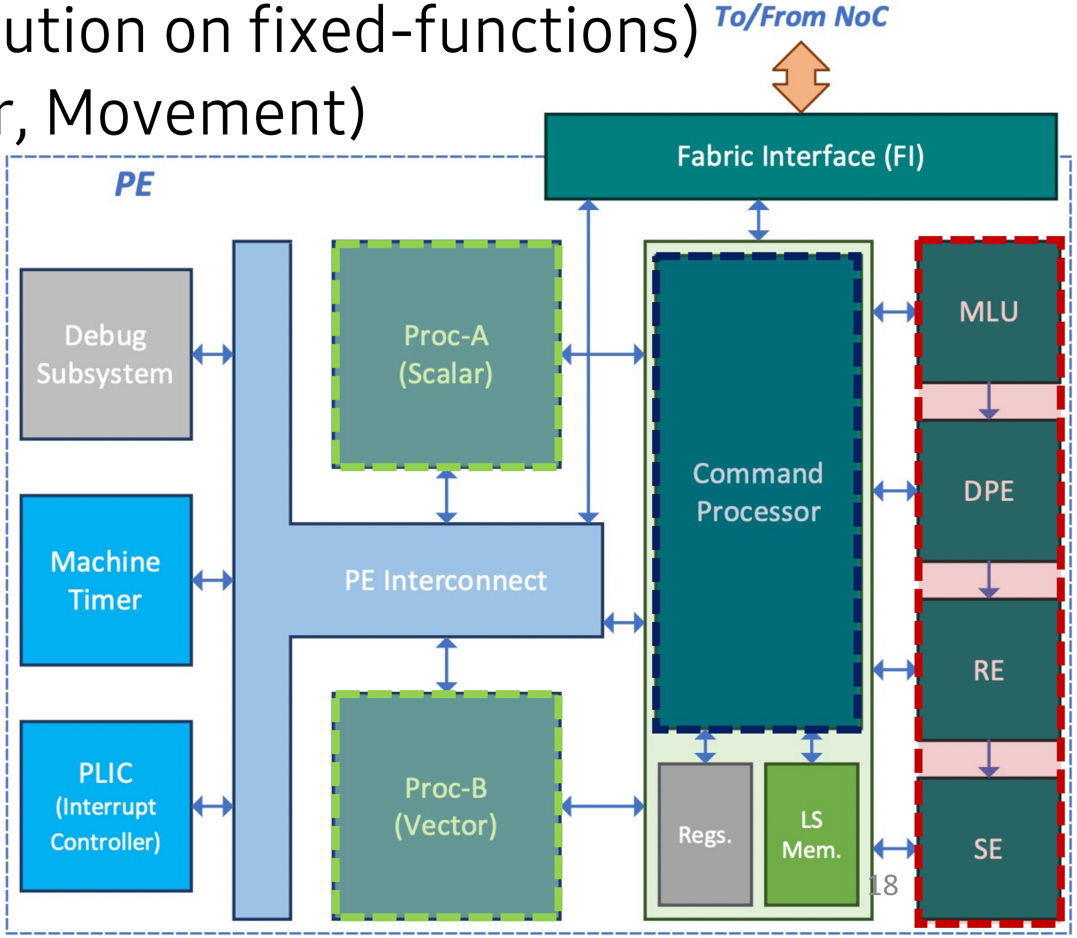
Architecture: Chip Overview

- **8×8 Grid of processing elements (PEs)**
  - **128MB SRAM** residing on edges of mesh
  - 16 channels of **LPDDR5** (Up to 64GB)
  - Control subsystem & Host interface



Architecture: Processing Element (PE)

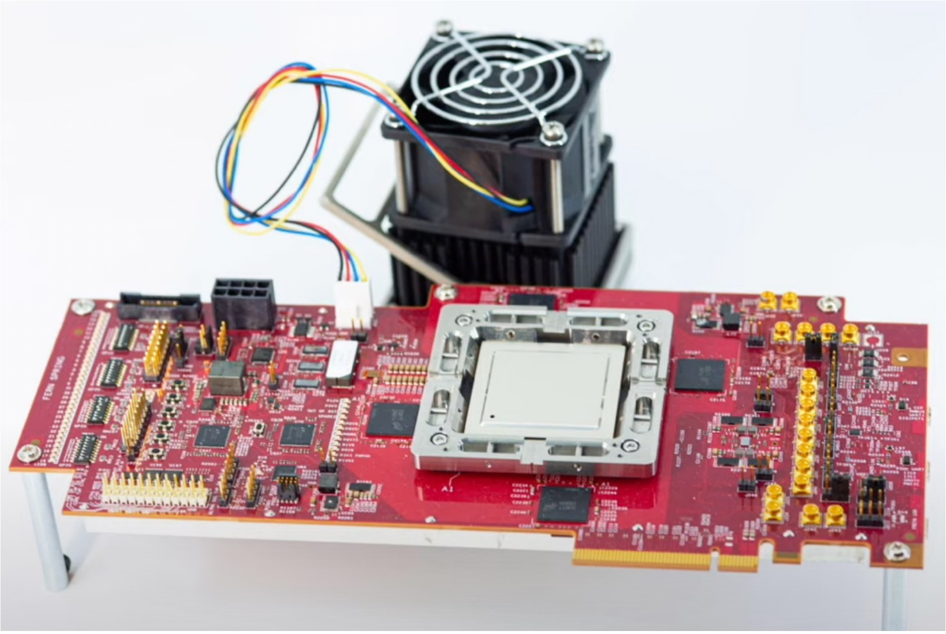
- RISC-V cores and Fixed-function units
  - **Command Processor** (Coordinating execution on fixed-functions)
  - **Fixed-function units** (GEMM, Non-Linear, Movement)
  - **Two RISC-V cores** (One with vector)
  - 128KB of local memory

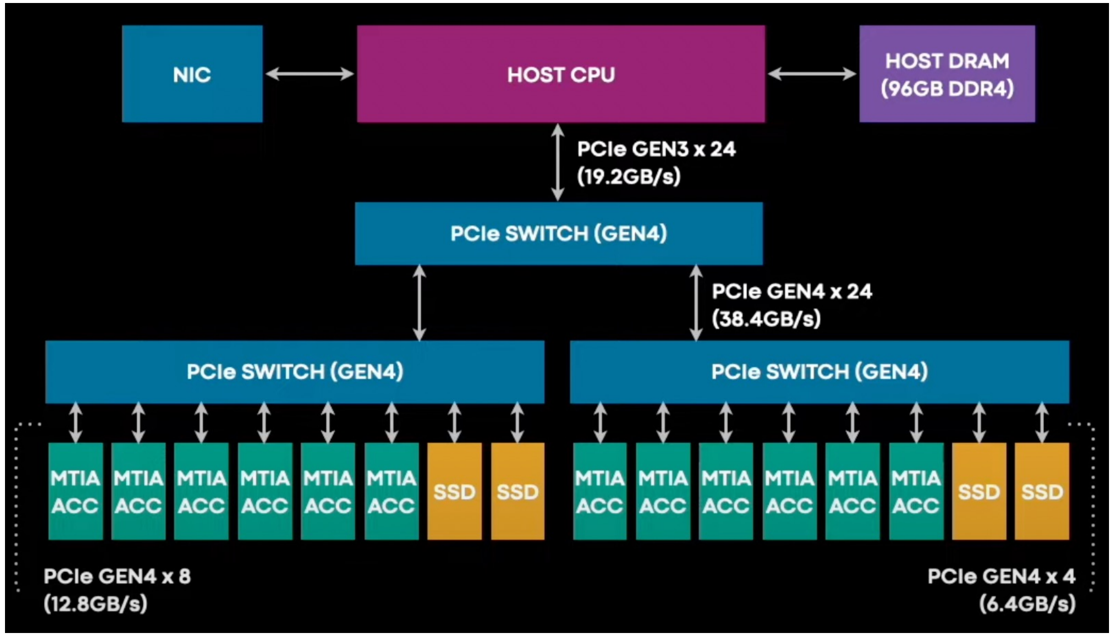


*MLU (Memory Layout Unit), DPE (Dot-Product Engine), RE (Reduction Engine), SE (SIMD Engine)

Architecture: Prototype Board

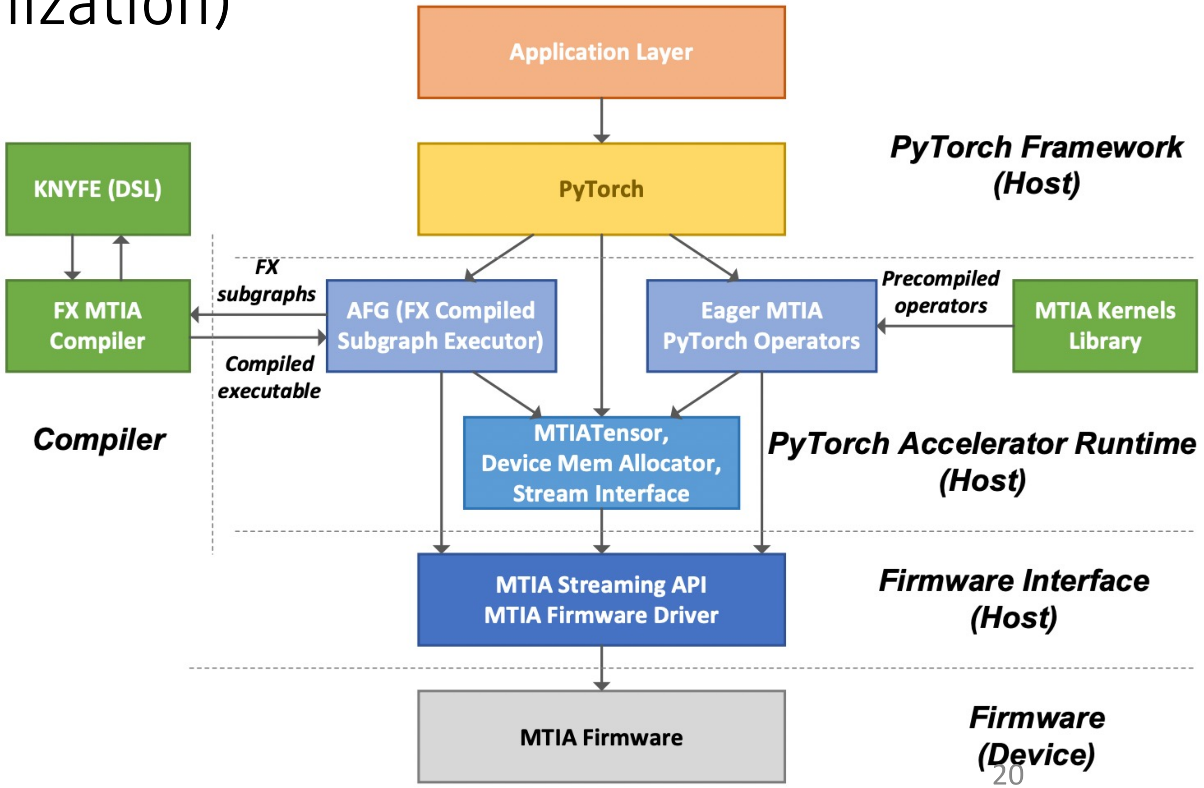
- **Dual M.2 form factor**
  - Board TDP of 35W, PCIe 4 x8 (12.8GB/s)
  - 4*LPDDR5 (4ch, 64b, 32GB), Yosemite v3 server (12 MTIAs)





Architecture: SW Stack (Compiler)

- Providing developer efficiency and high performance
  - **FX-based mode** (Model-level transformations/optimization)
  - **LLVM-based mode** (Low-level optimization)



Result: Experimental Setup

- **Operator-based benchmarks as well as full DRLM models**
  - Evaluation of Dense/Sparse Computation and DLRM models
  - Breakdown of important operators and kernels

‹ DLRM models used for evaluation ›

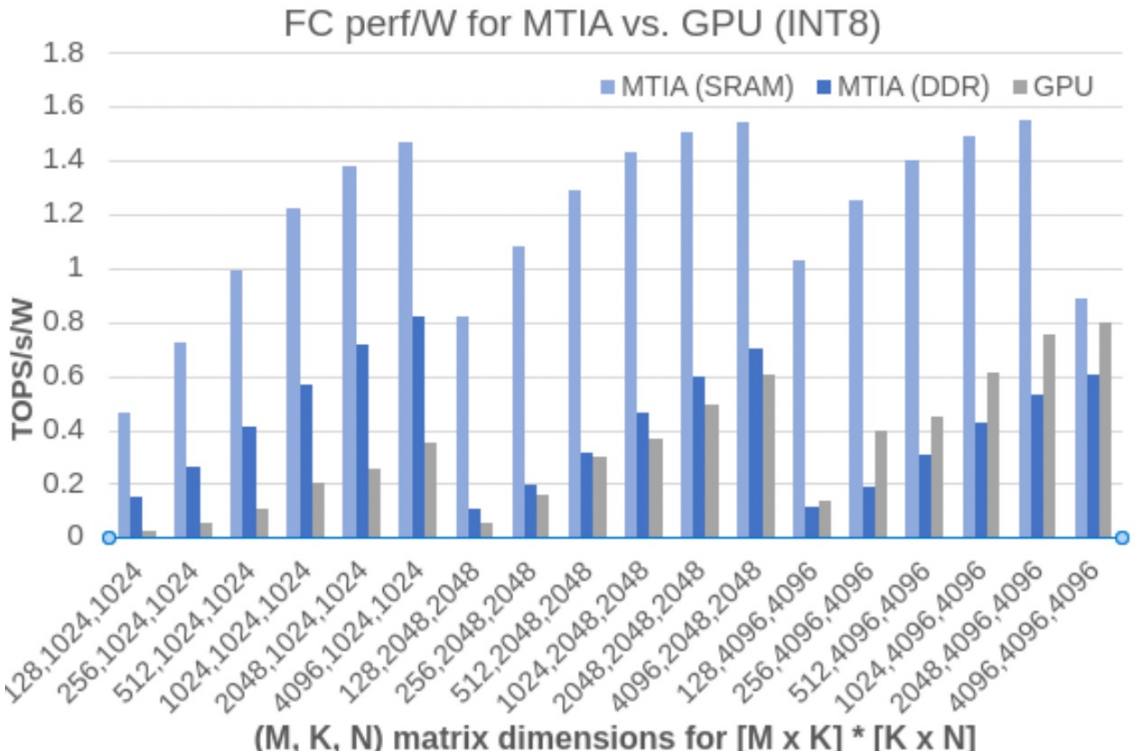
DLRM Model	Size (GB)	Complexity (GFLOPS/batch)
Low Complexity 1 (LC1)	53.2	0.032
Low Complexity 2 (LC2)	4.5	0.014
Medium Complexity 1 (MC1)	120	0.140
Medium Complexity 2 (MC2)	200	0.220
High Complexity (HC)	725	0.450

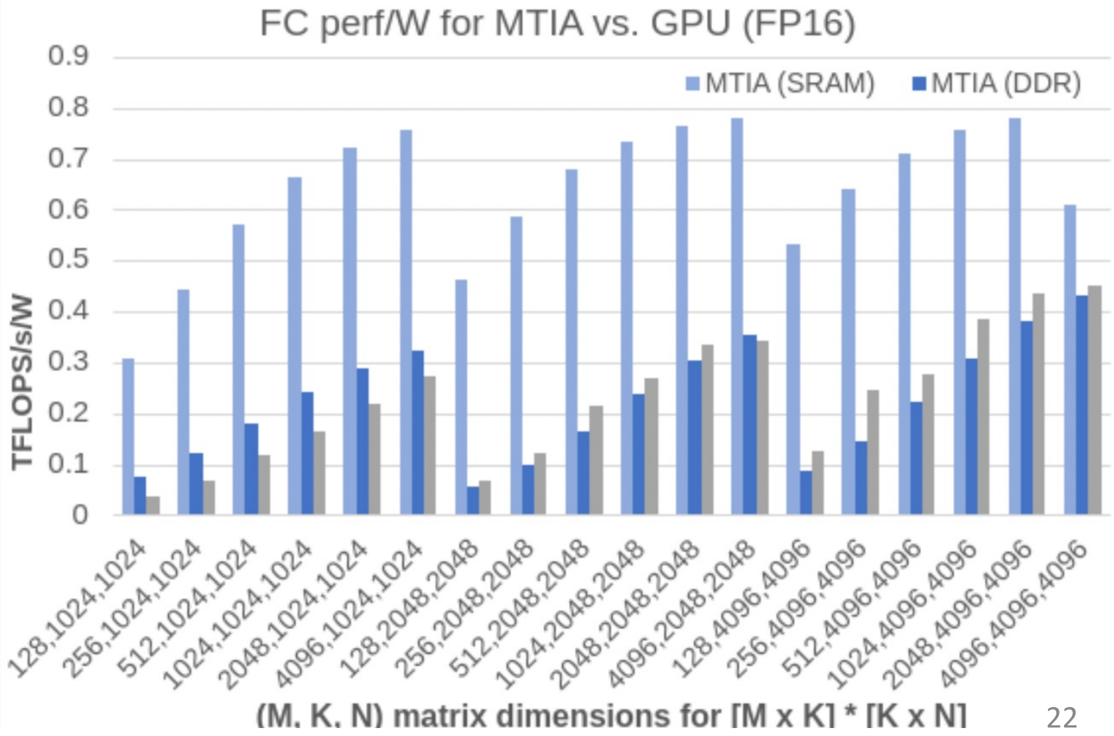
‹ Operator breakdown, MC2 ›

Operator	Batch size 64	Batch size 256
FC (Fully Connected)	42.10 %	32.4%
EB (Embedding Bag)	31.19 %	30.0%
Concat	2.86 %	11.5%
Transpose	8.47 %	5.9%
Quantize	1.55 %	5.3%
Dequantize	2.94 %	3.3%
BatchMatMul	3.30 %	1.7%
Others	7. 59 %	11.0%

Result: Dense GEMM Computation

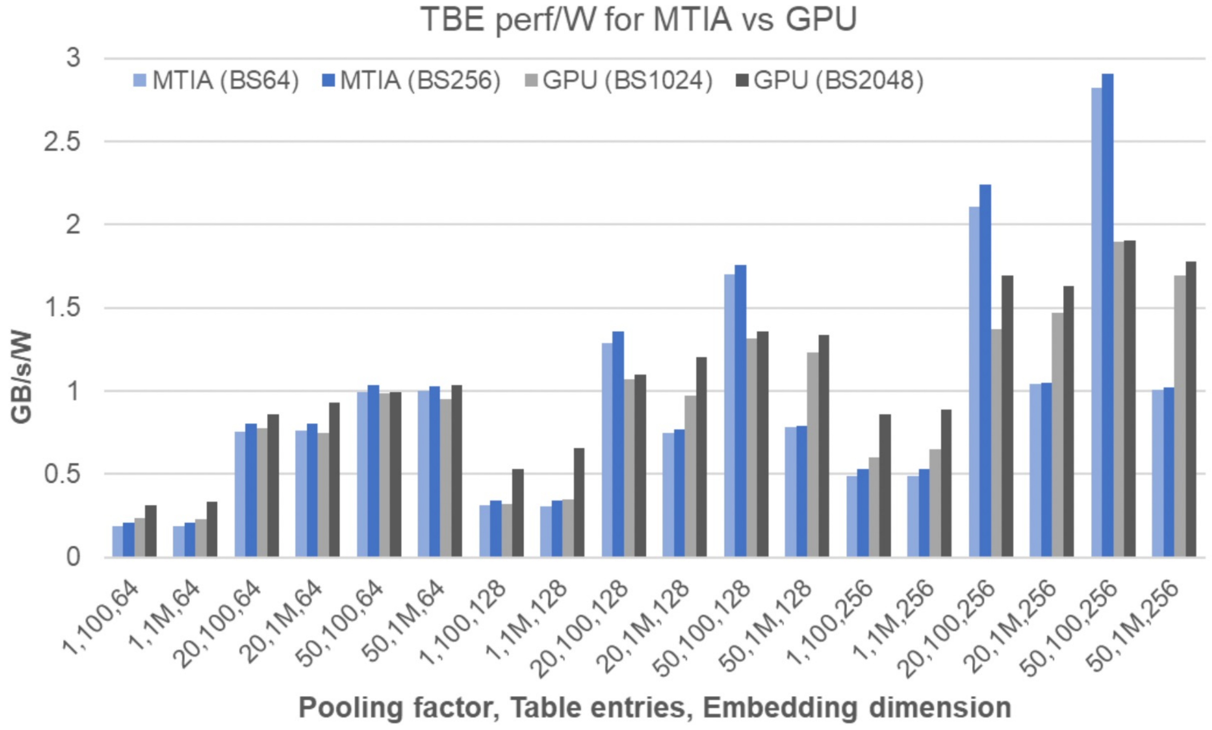
- **Comparison across inference accelerators**
  - Most efficient when tensors can be streamed directly from **SRAM**
  - Effective for **low batch sizes** when serving requests under stringent latency

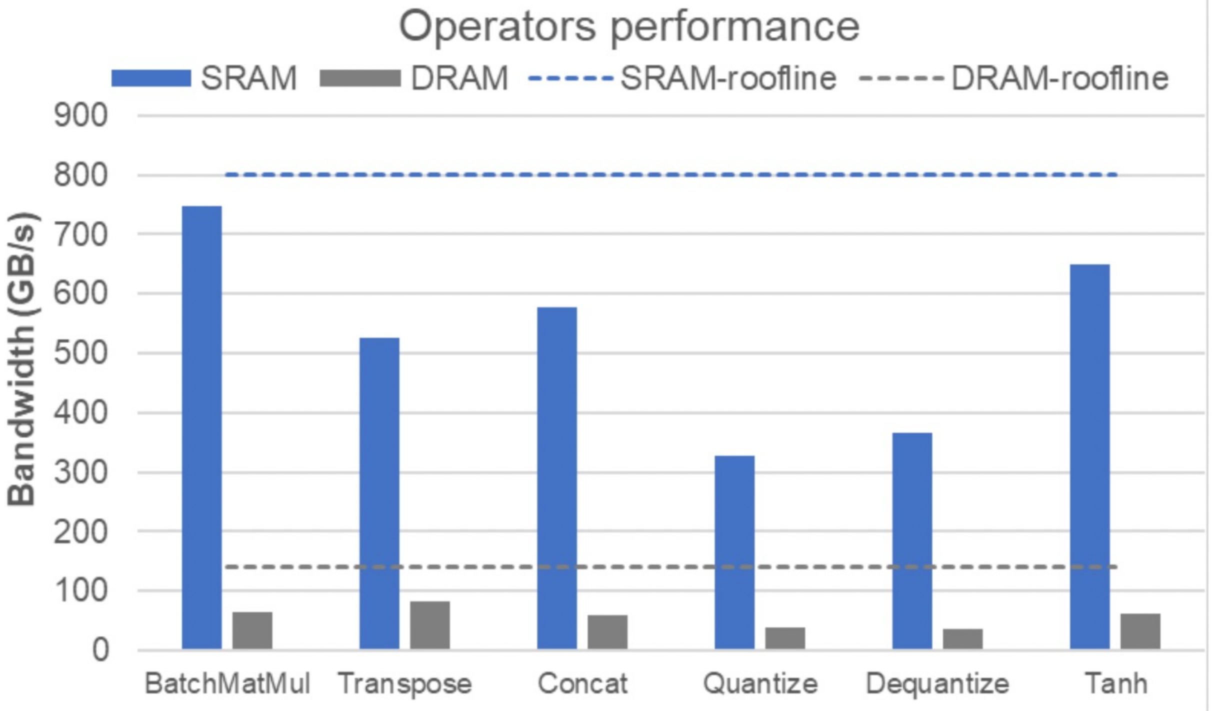




Result: Sparse Computation

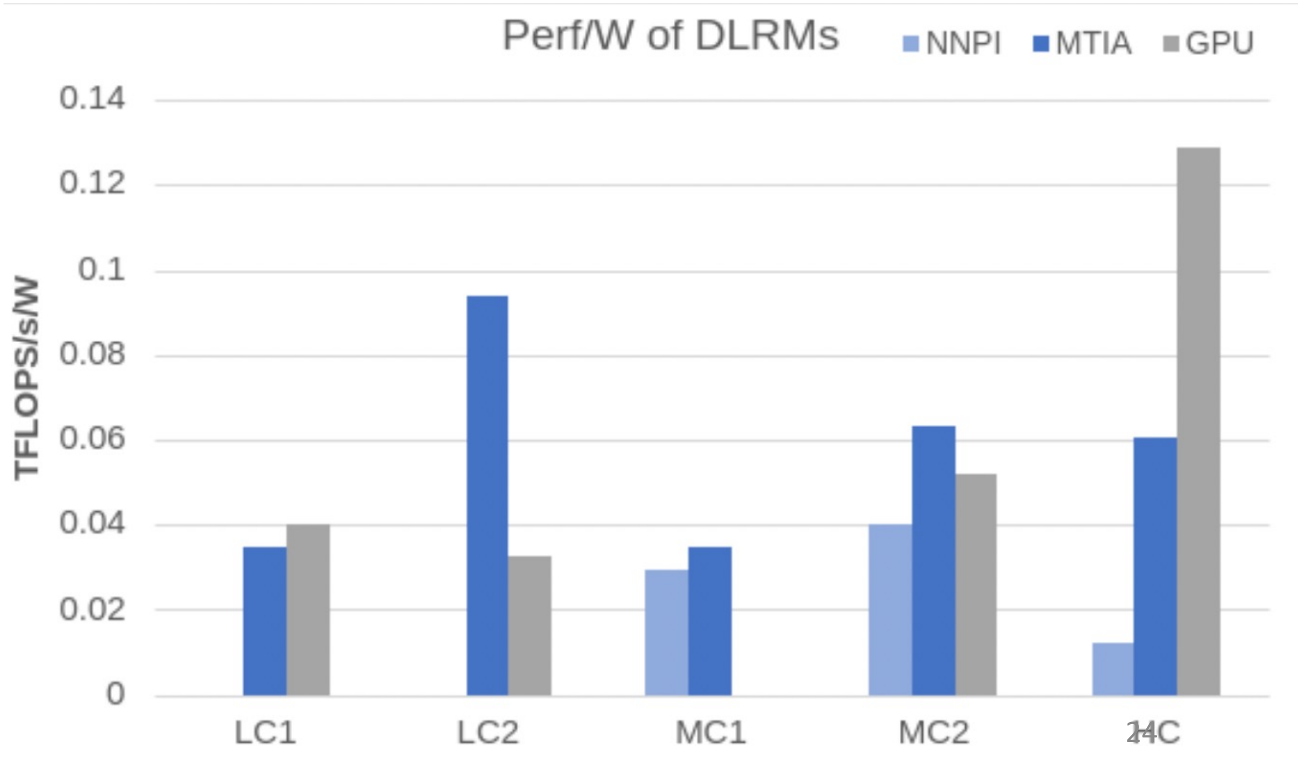
- **DLRM include hundreds of EmbeddingBag (EB) operators**
  - Embedding operation is mostly **memory bound** (GB/s metric)
  - MTIA (10~20% of its memory BW), GPU (60% of its HBM BW)





Result: Model Performance

- **Comparison across inference accelerators**
  - Low complexity model (FC layers with small input shapes)
  - High complexity model (FC layers are less dominant)



Thank You