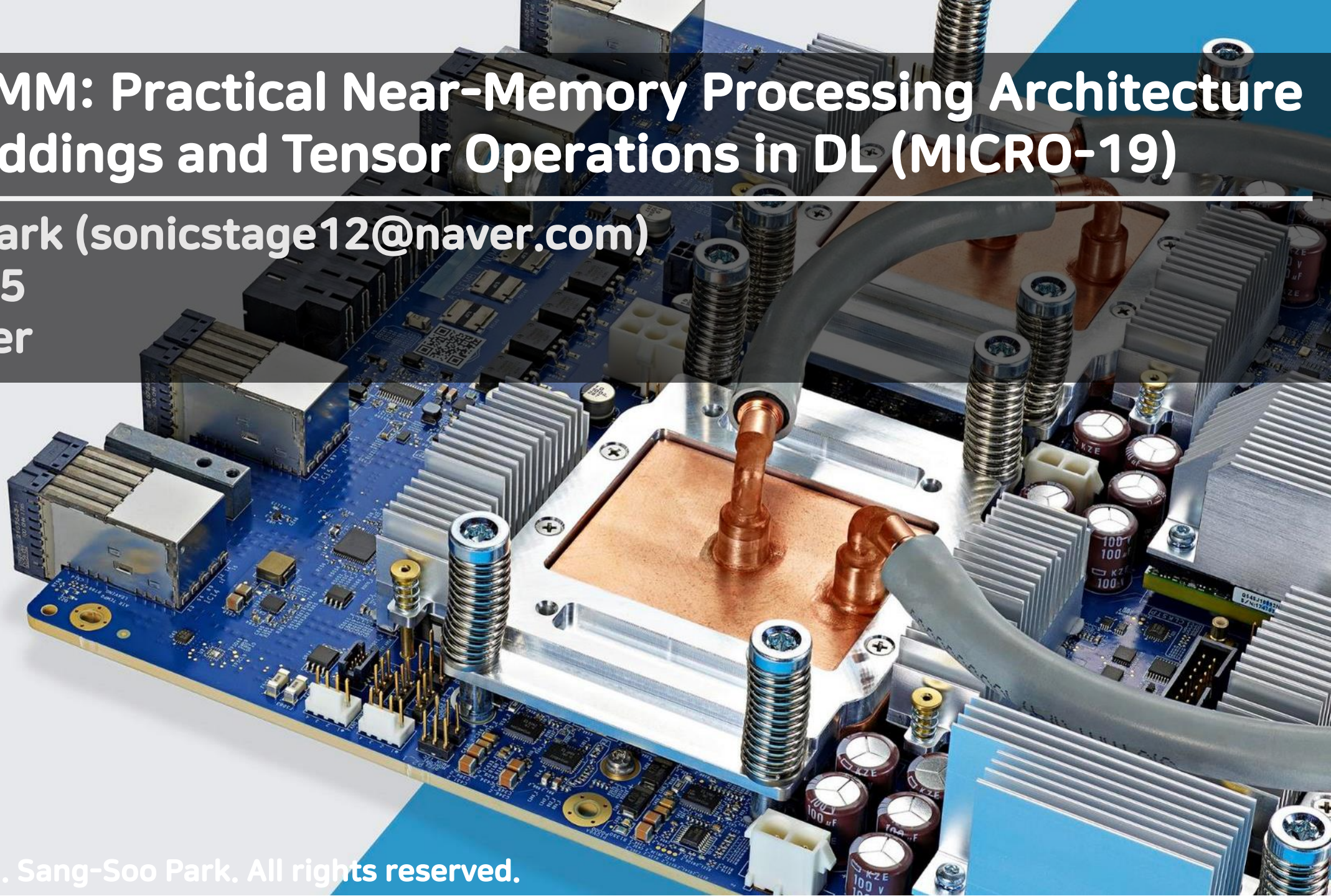
TensorDIMM: Practical Near-Memory Processing Architecture for Embeddings and Tensor Operations in DL (MICRO-19)

Constant Park (sonicstage12@naver.com)

2021. 08. 05

DL_Compiler

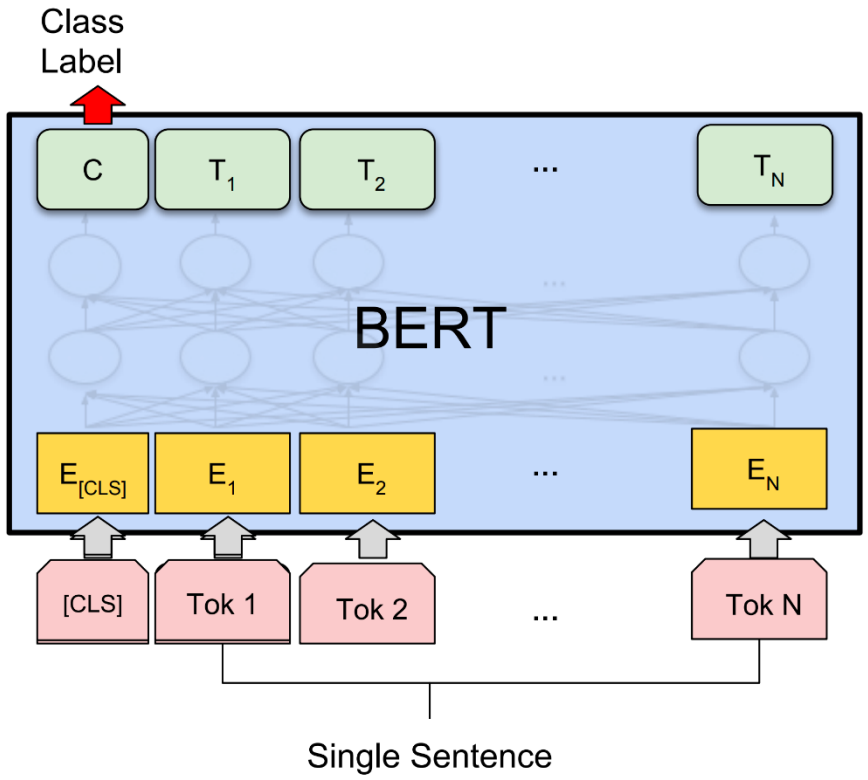


Contents

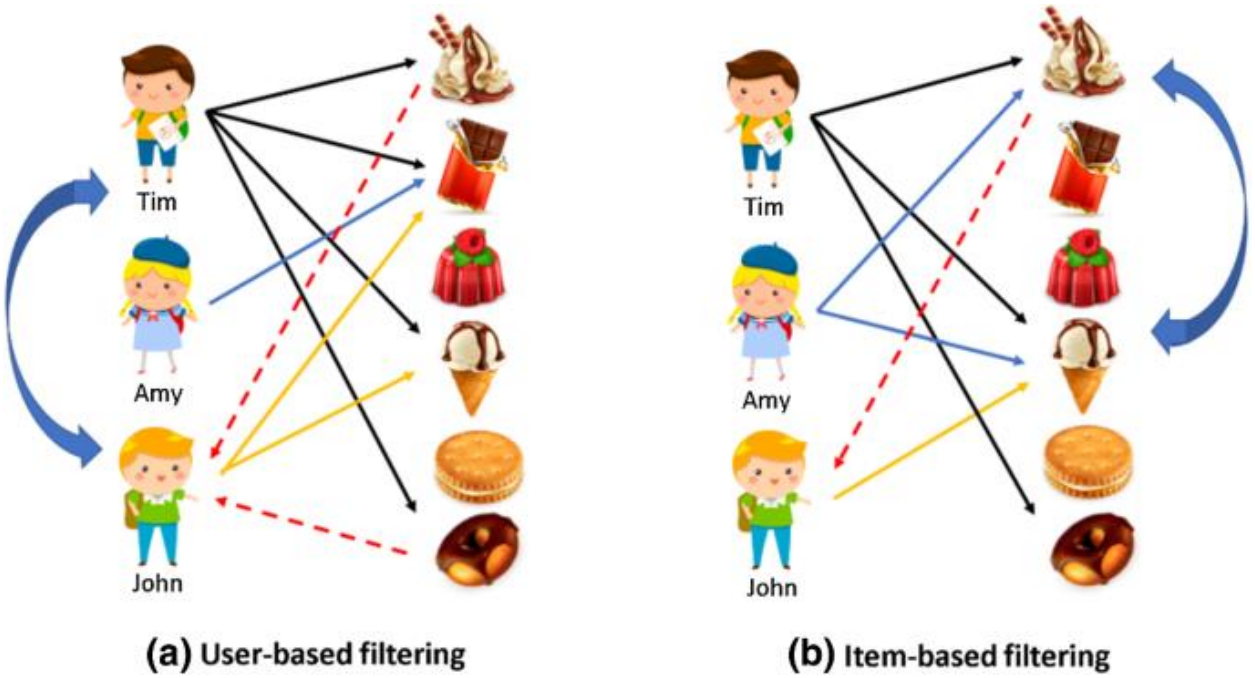
- **Embedding Layer in DNN**
- NDP: HW Architecture for Embedding Layer
- TensorDIMM: Practical accelerator

Emerging DNN Applications

- **Non-conventional DNN layers cause bottlenecks**
  - Attention module (BERT)
  - Collaborative filter (Recommendation System)



BERT Architecture

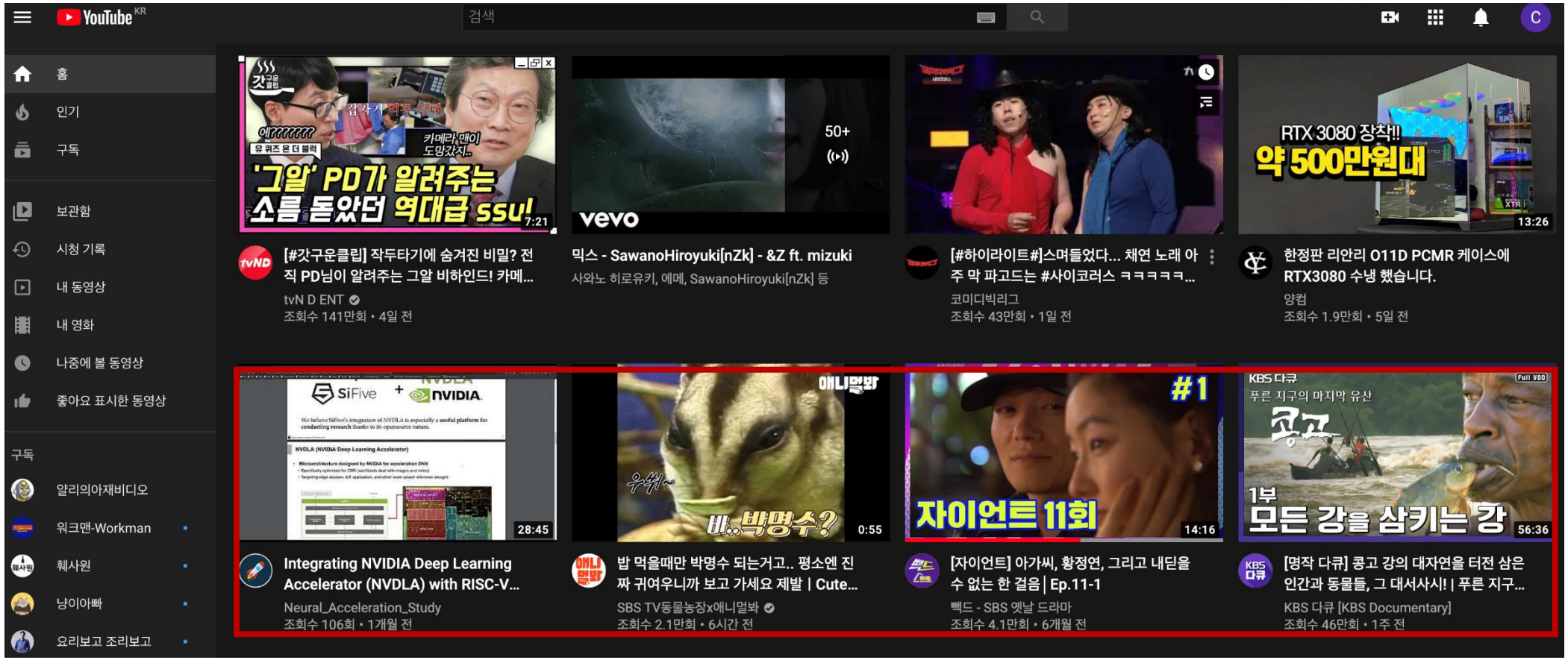


Collaborative Filtering

Recommendation System

Personalized recommendation for contents

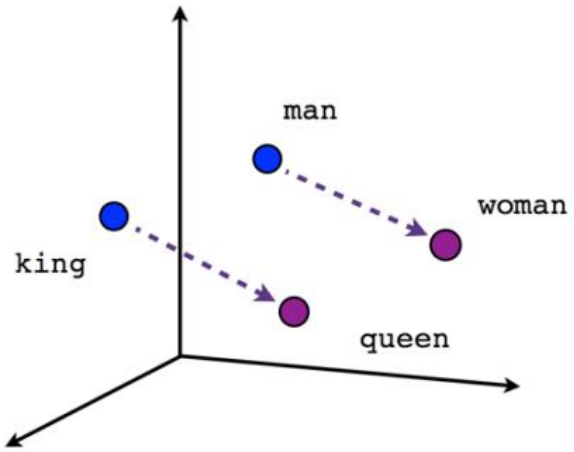
- Sparse embedding layers are bottleneck



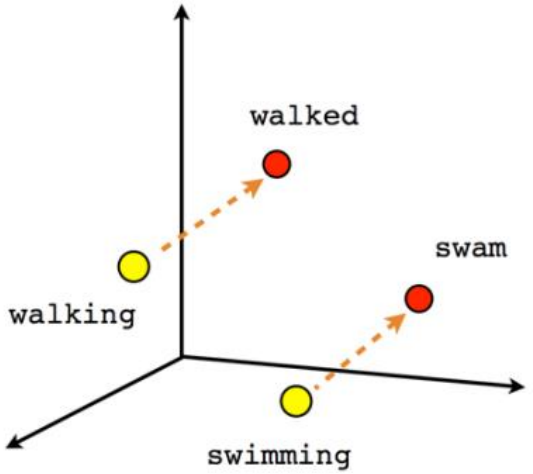
추천시스템의 예: 사용자에게 컨텐츠를 추천하는 YouTube

Embedding Layers: Projection to vector space

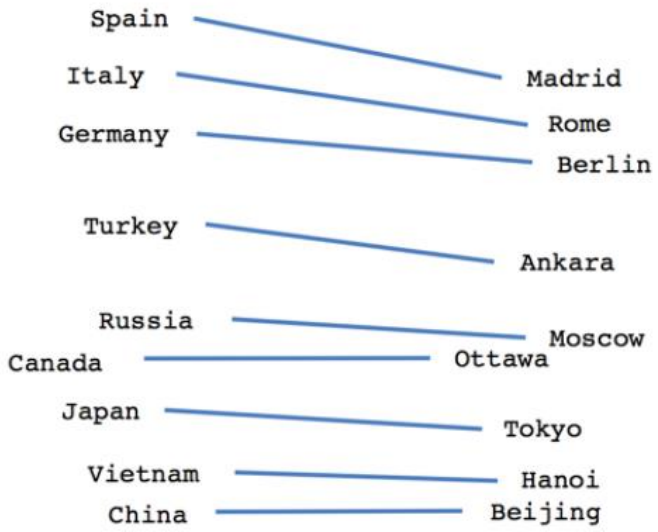
- **Words or phrases from the vocabulary are mapped to vectors of real numbers**
  - Word embedding (Word2Vec)
  - Neural Item Embedding for Collaborative Filtering (Item2Vec)



Male-Female



Verb tense

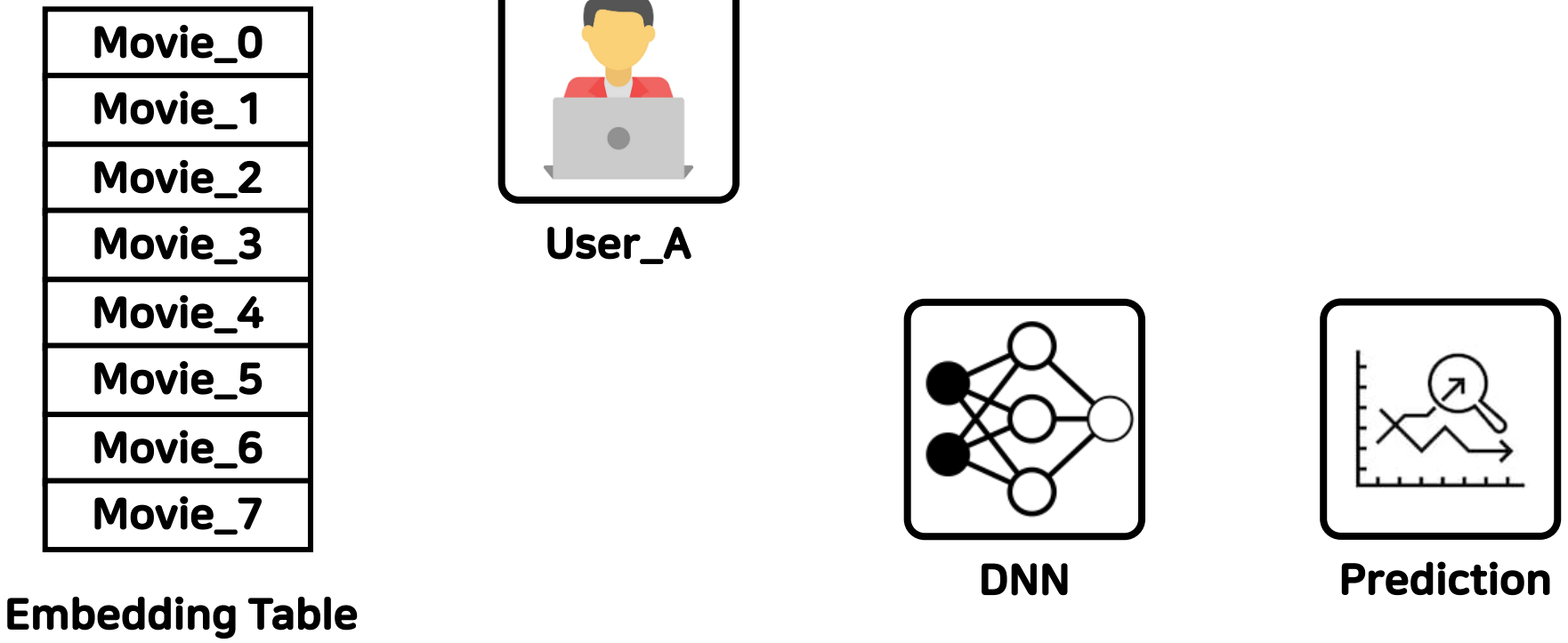


Country-Capital

단어를 벡터로 바꾸는 모델: 임베딩 모델 (Word2Vec)

Recommendation System: Overview

- **Goal: Predicting preference of user-item pair**
  - Movie recommendation



추천시스템의 예: 영화 추천

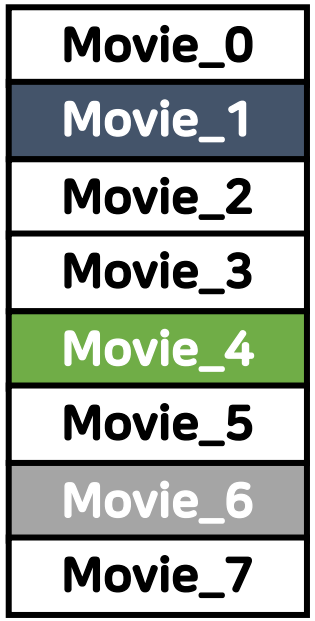
Recommendation System: Overview

- **Goal: Predicting preference of user-item pair**
  - Movie recommendation

Harry Porter

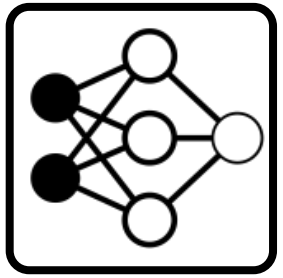
Batman

Ironman



Embedding Table

User_A



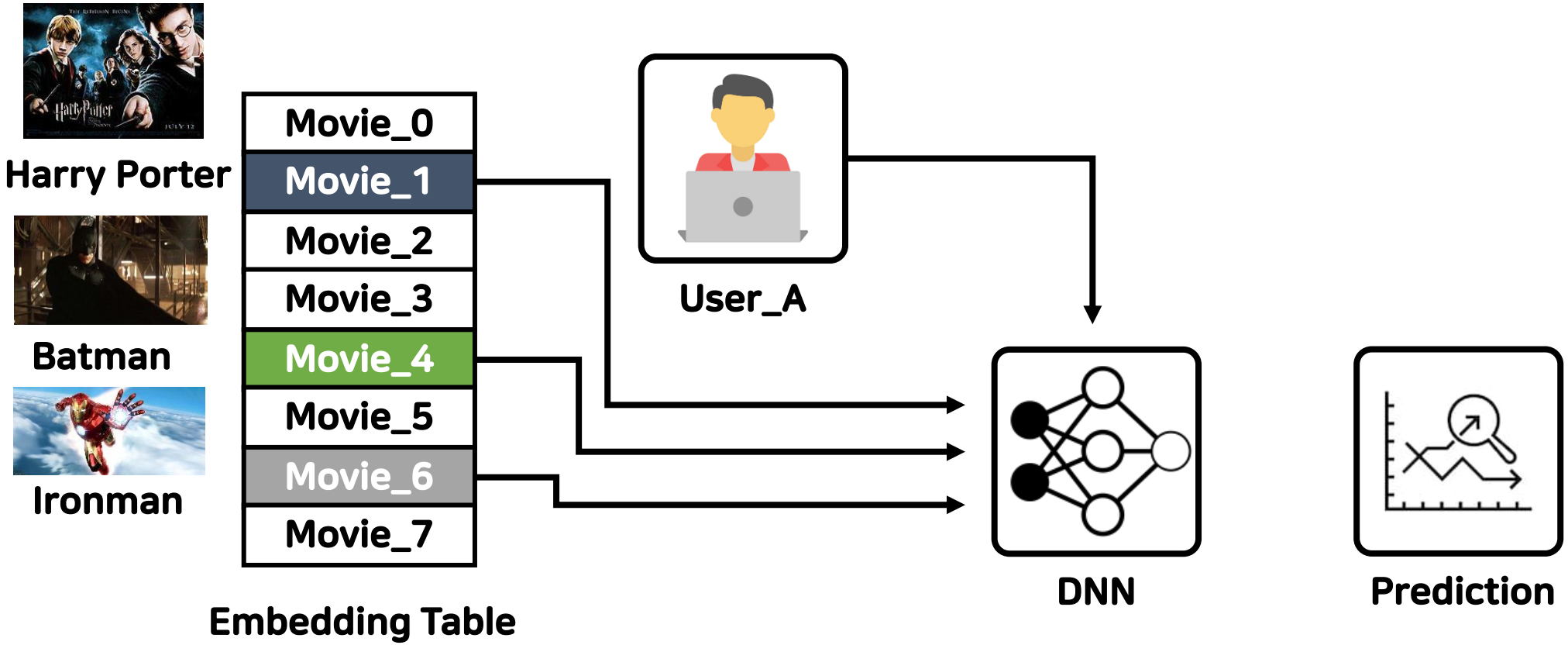
DNN

Prediction

추천시스템의 예: 영화 추천

Recommendation System: Overview

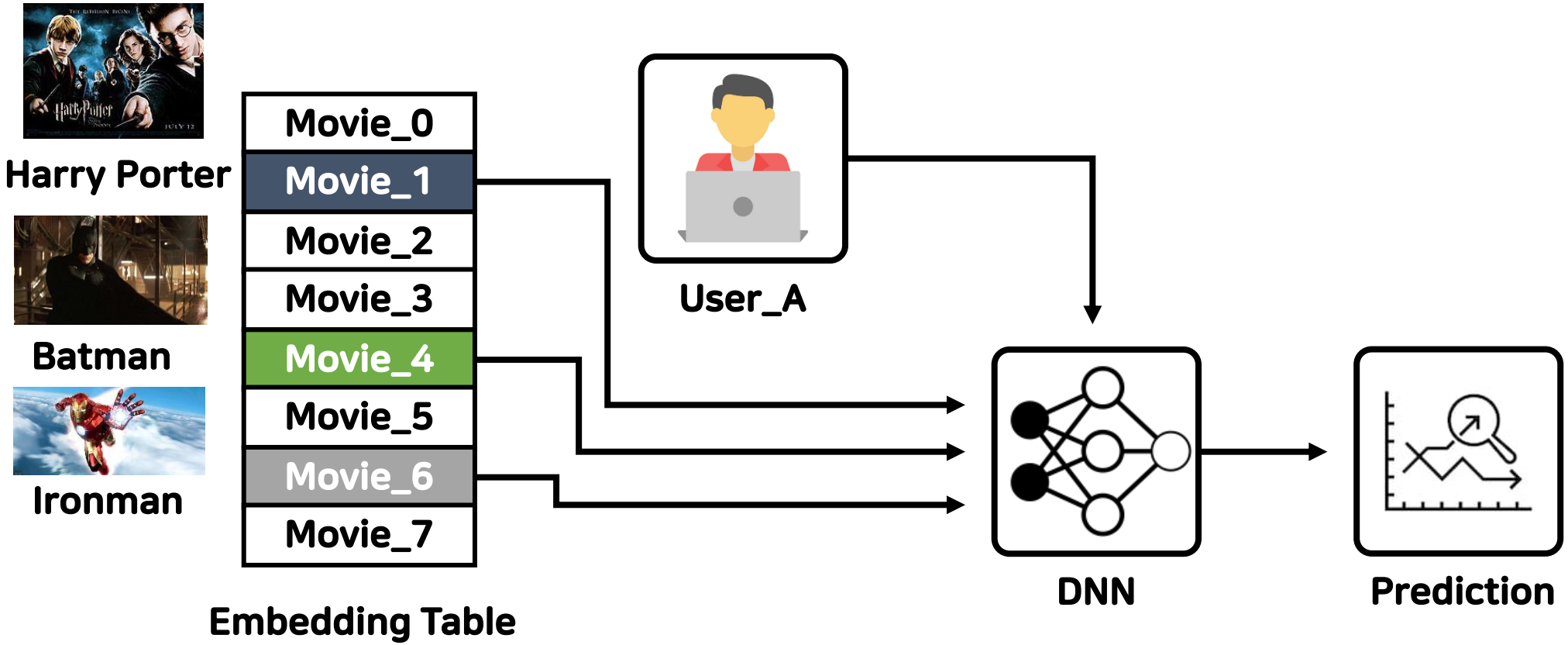
- **Goal: Predicting preference of user-item pair**
  - Movie recommendation



추천시스템의 예: 영화 추천

Recommendation System: Overview

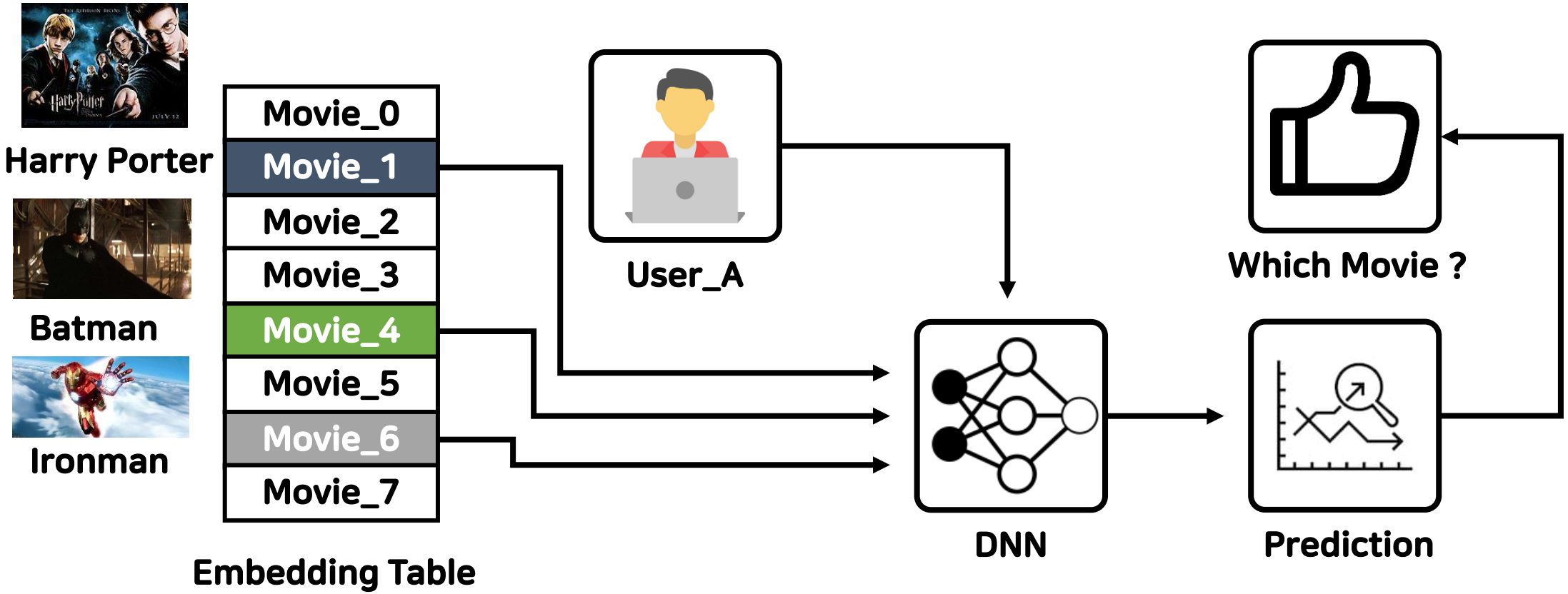
- **Goal: Predicting preference of user-item pair**
  - Movie recommendation



추천시스템의 예: 영화 추천

Recommendation System: Overview

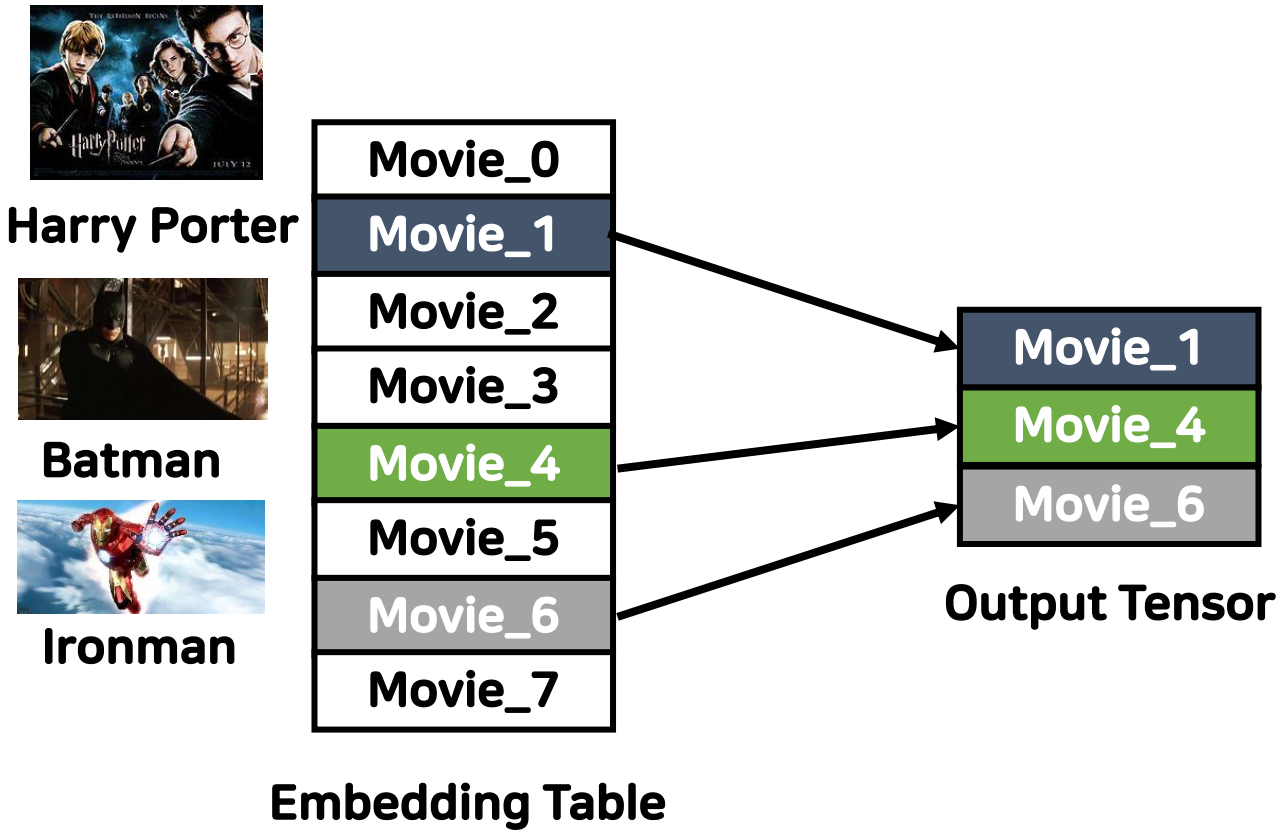
- **Goal: Predicting preference of user-item pair**
  - Movie recommendation

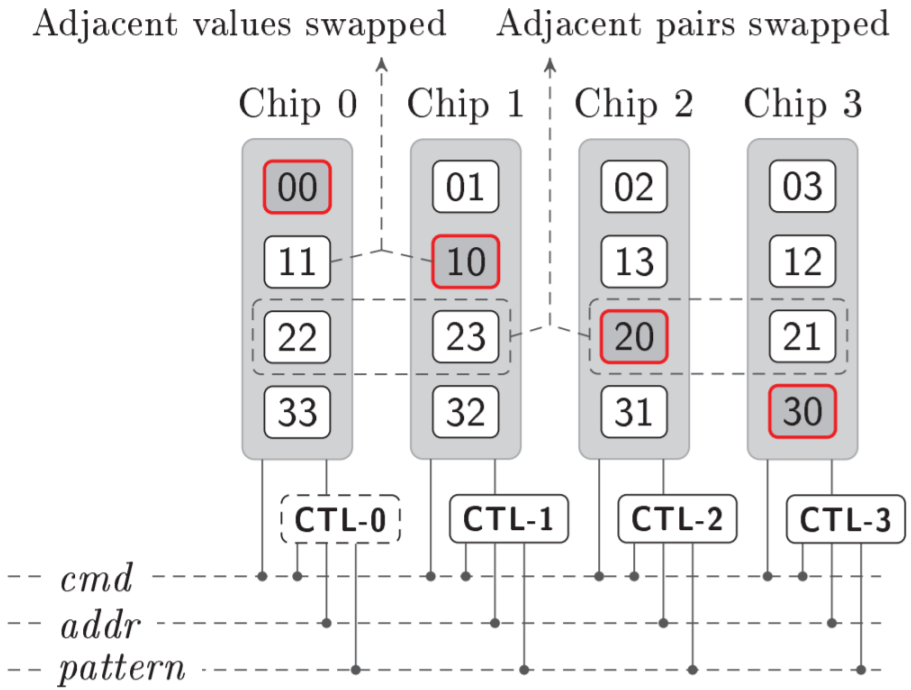


추천시스템의 예: 영화 추천

Recommendation System: Embedding layer

- **Copy target embeddings into contiguous address space**
  - Called “Gather”

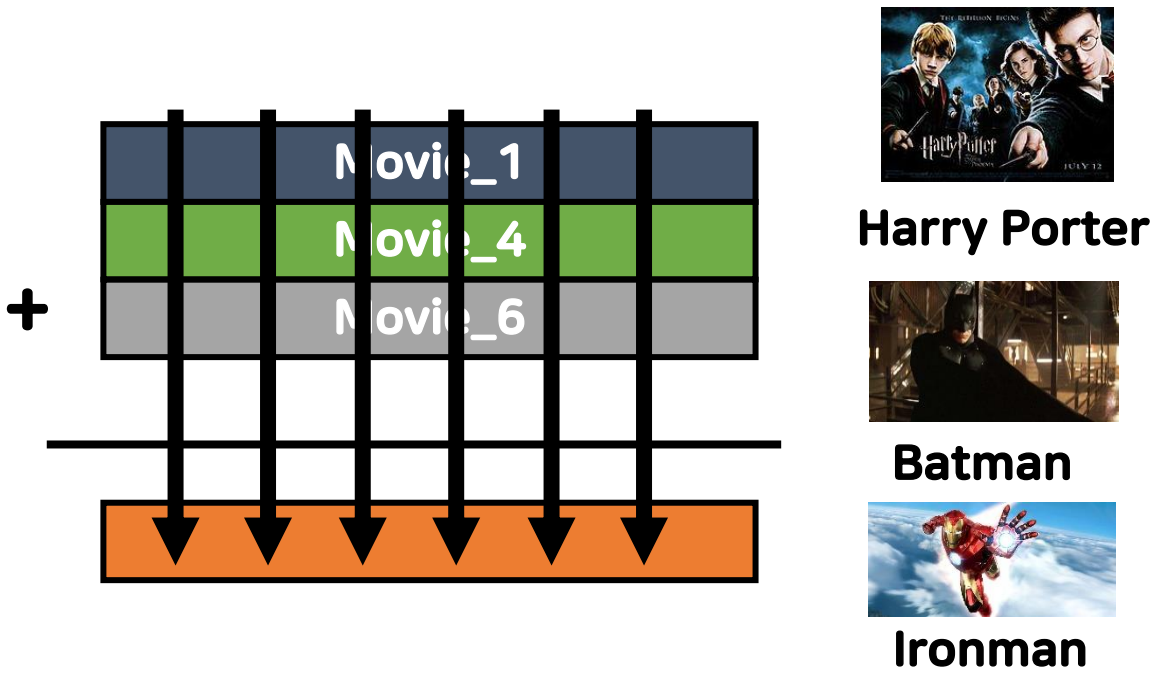




Gather Operation: (Left) Algorithm level, (Right) Hardware level (DRAM)[1]

Recommendation System: Embedding layer

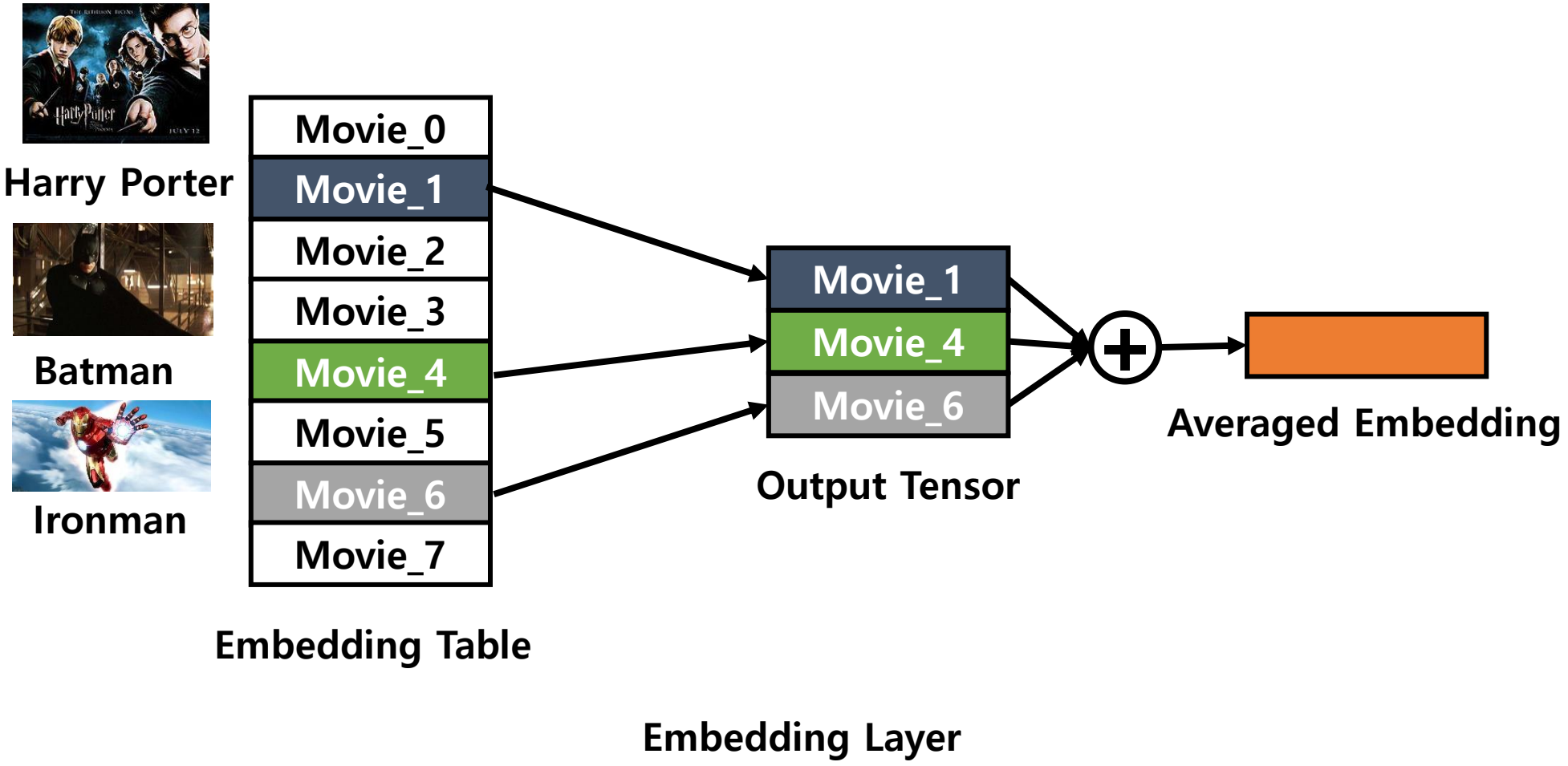
- **Averaging multiple embeddings, element-wise addition/multiplication**
  - Called “Reduction”



Recommendation System: Embedding layer

Gather/Reduction operation in Embedding layer

- This is memory-bandwidth sensitive operation



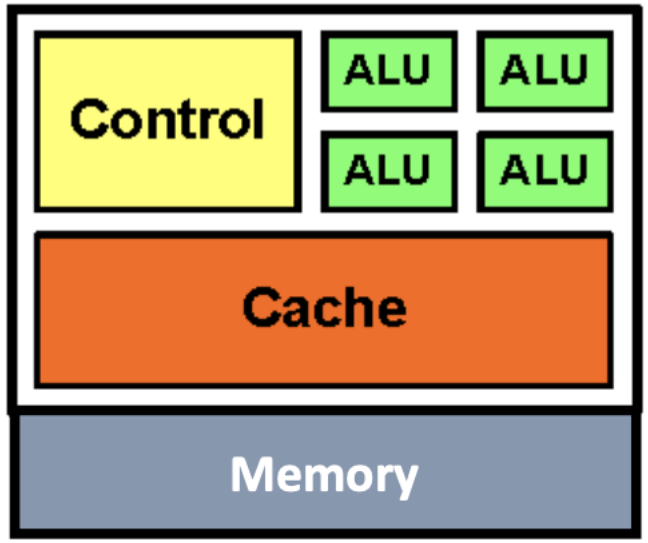
Contents

- Embedding Layer in DNN
- **NDP: HW Architecture for Embedding Layer**
- TensorDIMM: Practical accelerator

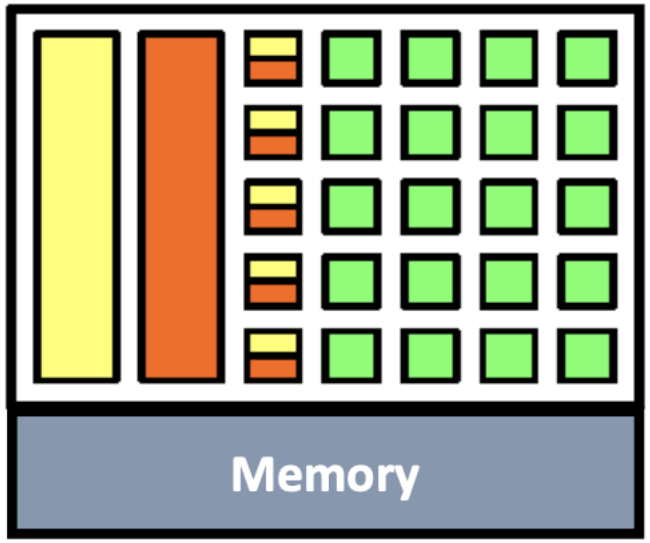
Near Data Processing (NDP): von Neumann

Data movement and Power efficiency

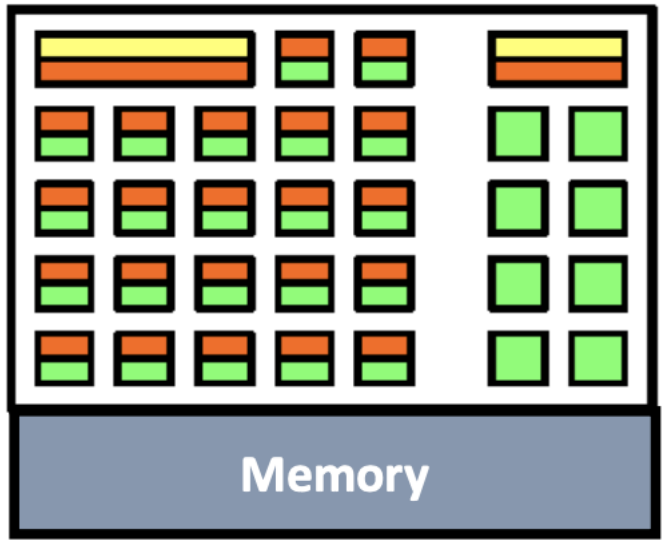
- CPU: Integrated multi-core (Good for complex problem)
- GPU: Many parallel computing unit (Simple and Massive problem)
- NPU: Under 10^4 processing element (Optimized for matrix multiplication)



CPU based



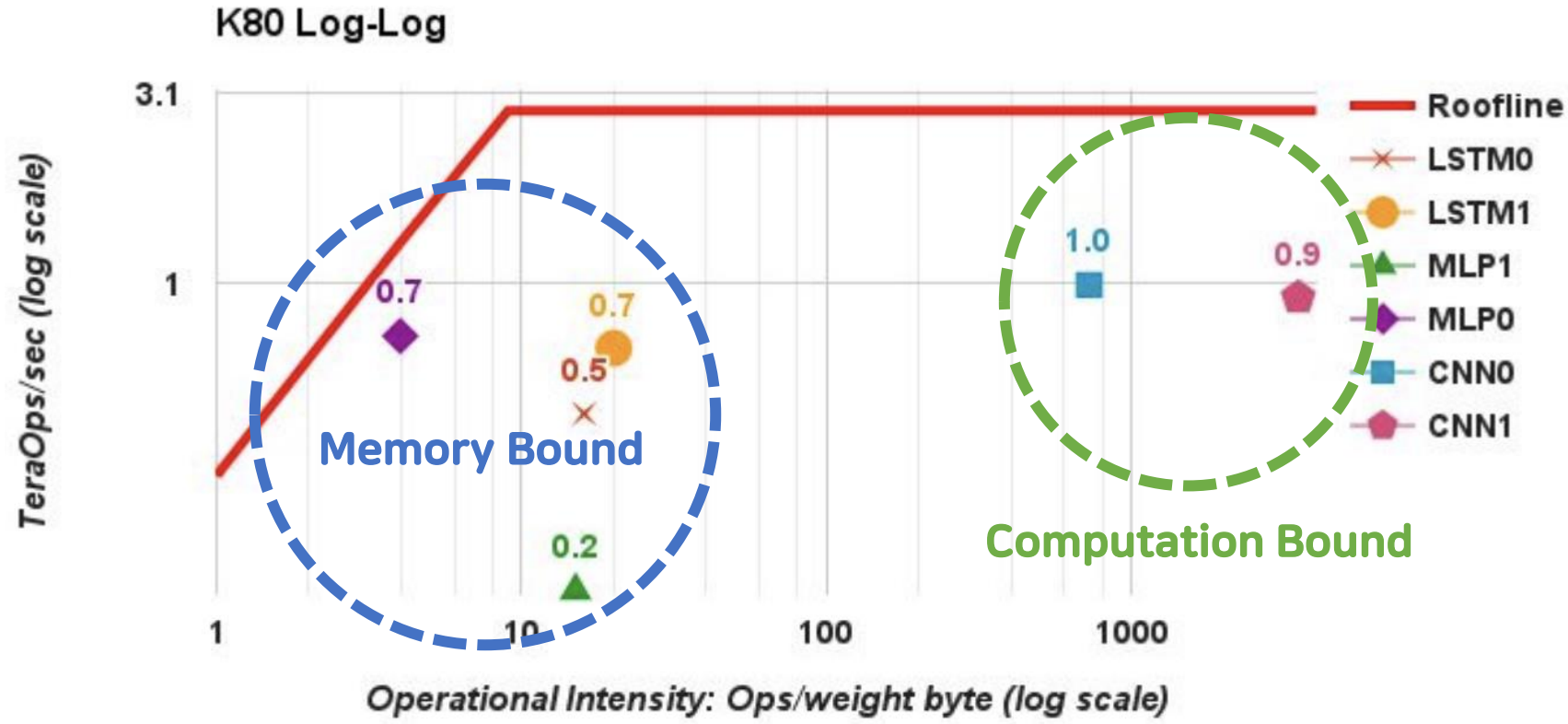
GPU



NPU (ASIC/FPGA)

Near Data Processing (NDP): Bottleneck in Memory

- **Most AI algorithm is memory bandwidth bounded**
  - Recent NN accelerators use internal memory to reduce bottleneck, but not sufficient
  - More than 60% of power is consumed by data movement

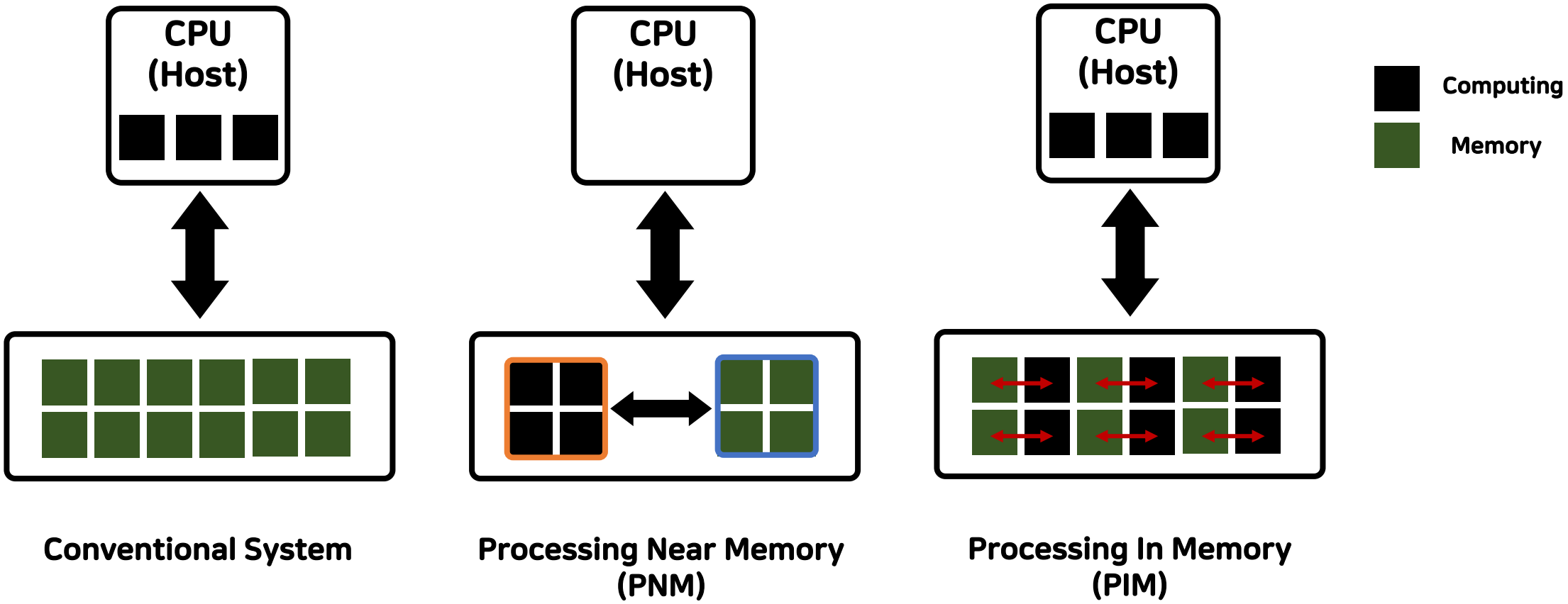


Roofline graph according to neural network[2]

Near Data Processing (NDP): Overview

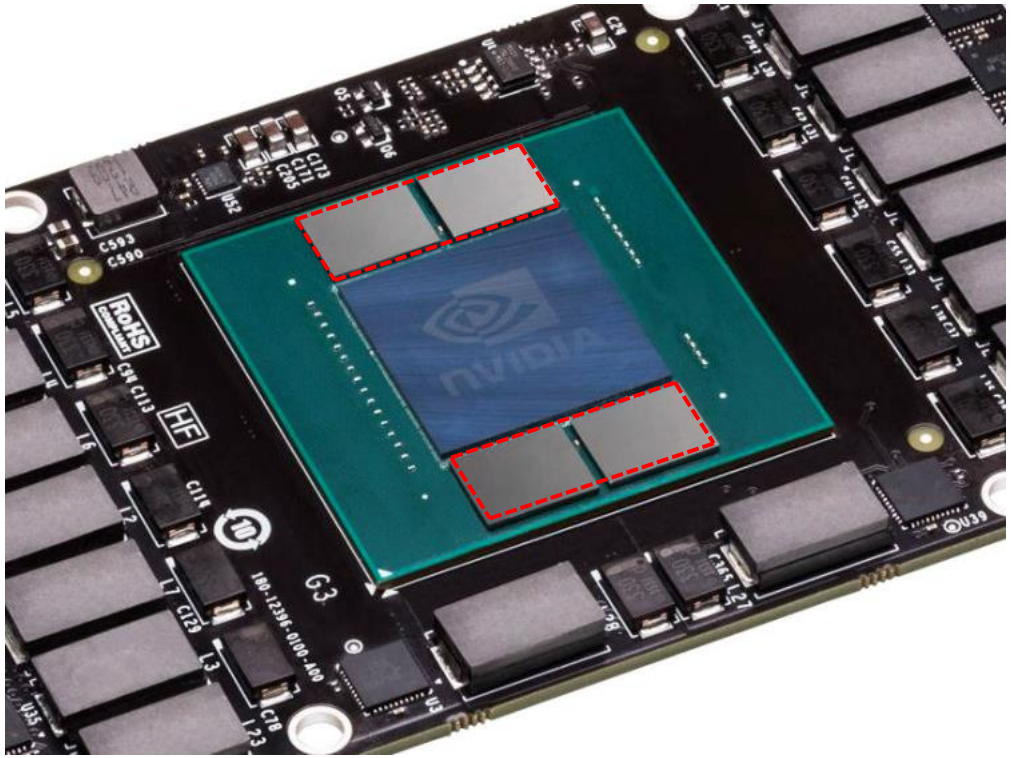
Moving computation to Data

- NDP can reduce energy for data transfer by locating computation unit where data lives
- Pros: Overcome bandwidth limitation between logic and memory devices with low power consumption
- Cons: Not backward compatible with legacy software stack

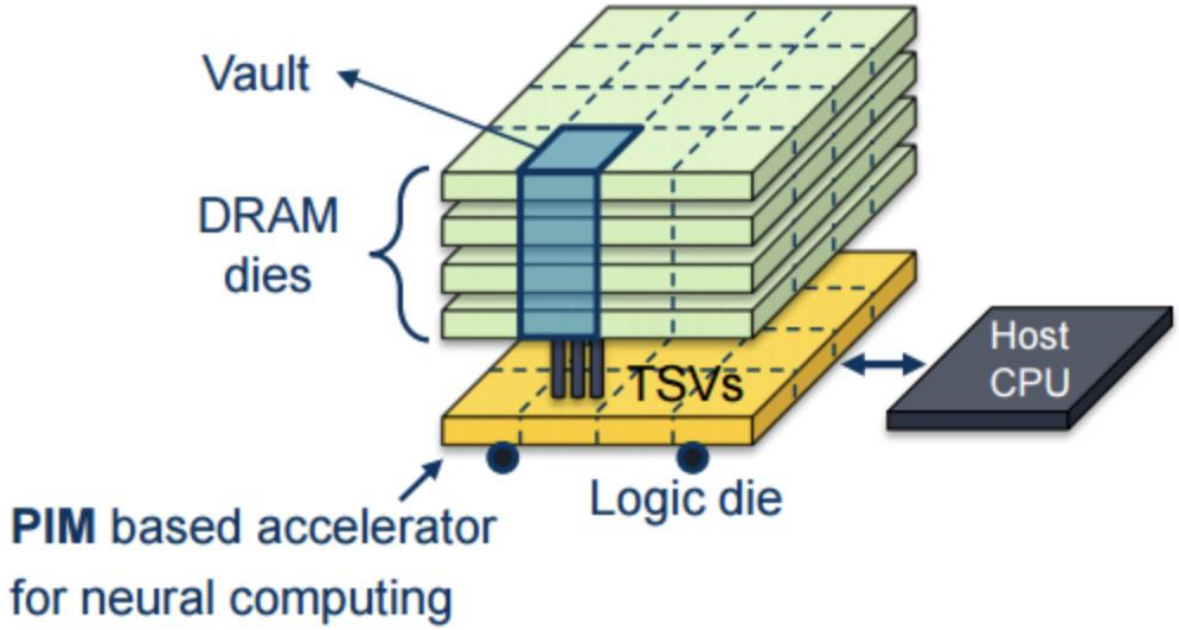


Near Data Processing (NDP): Example

- **HBM (High-bandwidth Memory) and HMC (Hybrid Memory Cube)**
  - HBM: Require less power consumption and latency compared to GDDR
  - HMC: Computation logic under DRAM



HBM in Pascal GPU



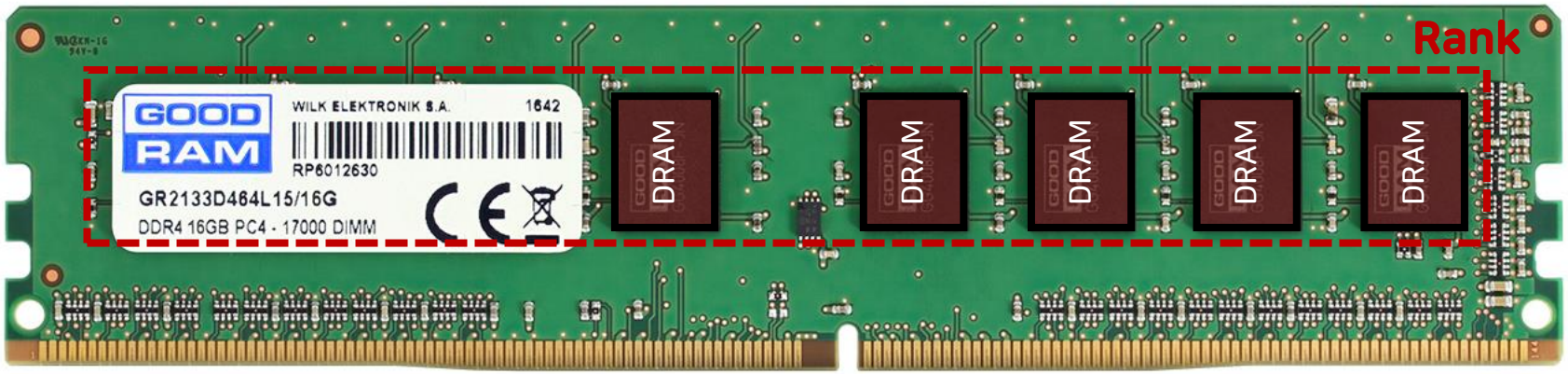
HMC Concept

Contents

- Embedding Layer in DNN
- NDP: HW Architecture for Embedding Layer
- **TensorDIMM: Practical accelerator**

TensorDIMM: Overview

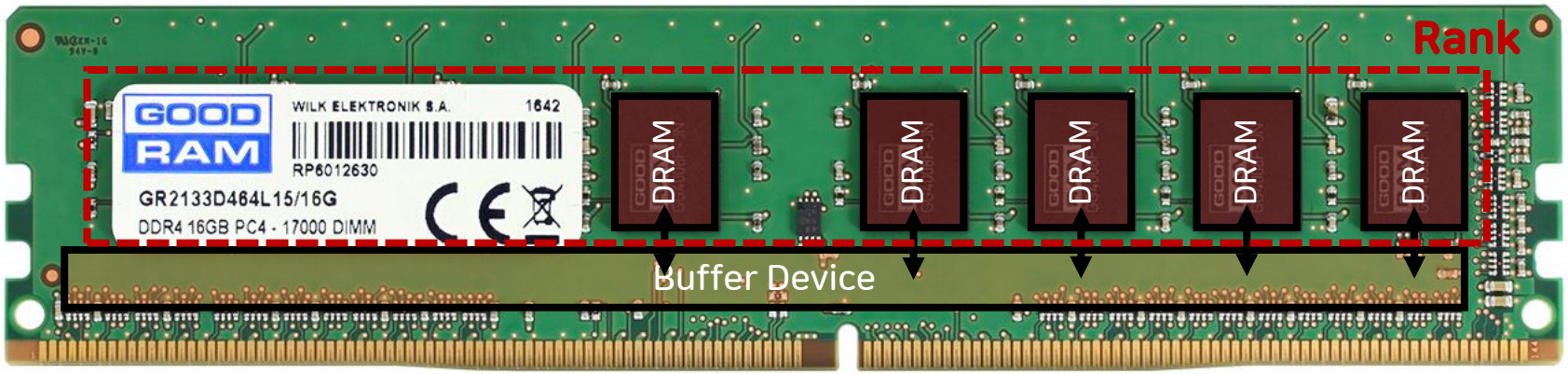
- **Buffer device to add NMP cores for embedding gather/reduction**
  - DIMM (Dual-in-line memory module)
  - NMP (Near-memory processing processor)
    - Vector ALU: Vector multiplication/addition unit



DRAM module (DIMM) and TensorDIMM concept

TensorDIMM: Overview

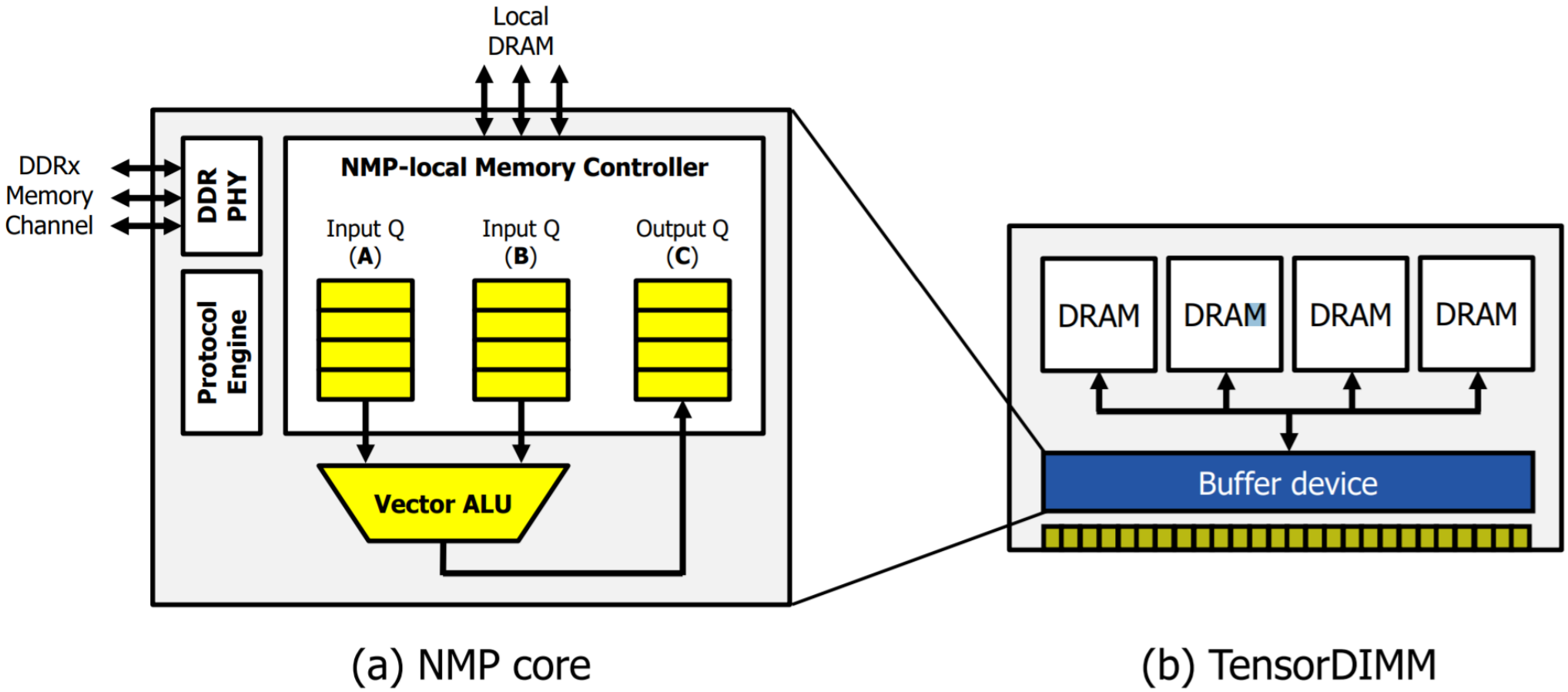
- **Buffer device to add NMP cores for embedding gather/reduction**
  - DIMM (Dual-in-line memory module)
  - NMP (Near-memory processing processor)
    - Vector ALU: Vector multiplication/addition unit



DRAM module (DIMM) and TensorDIMM concept

TensorDIMM: Overview

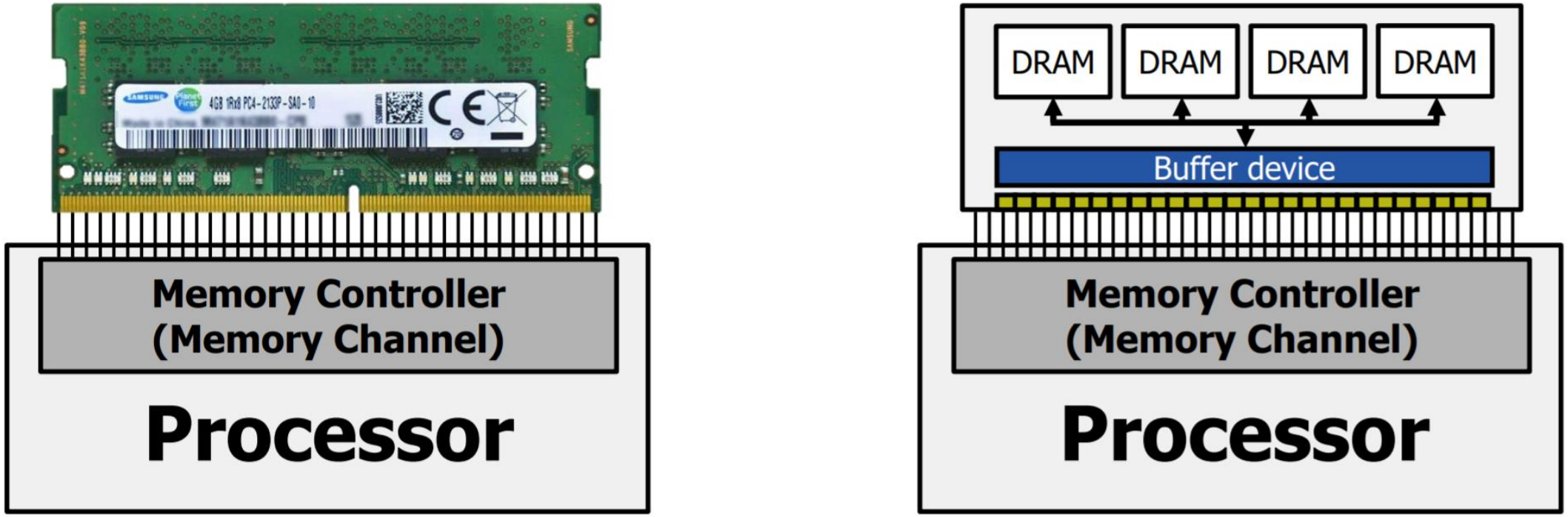
- **Buffer device to add NMP cores for embedding gather/reduction**
  - DIMM (Dual-in-line memory module)
  - NMP (Near-memory processing processor)
    - Vector ALU: Vector multiplication/addition unit



TensorDIMM: How to work

Key advantage of TensorDIMM

- Effieve memory bandwidth scales proportional to the # of DIMMs

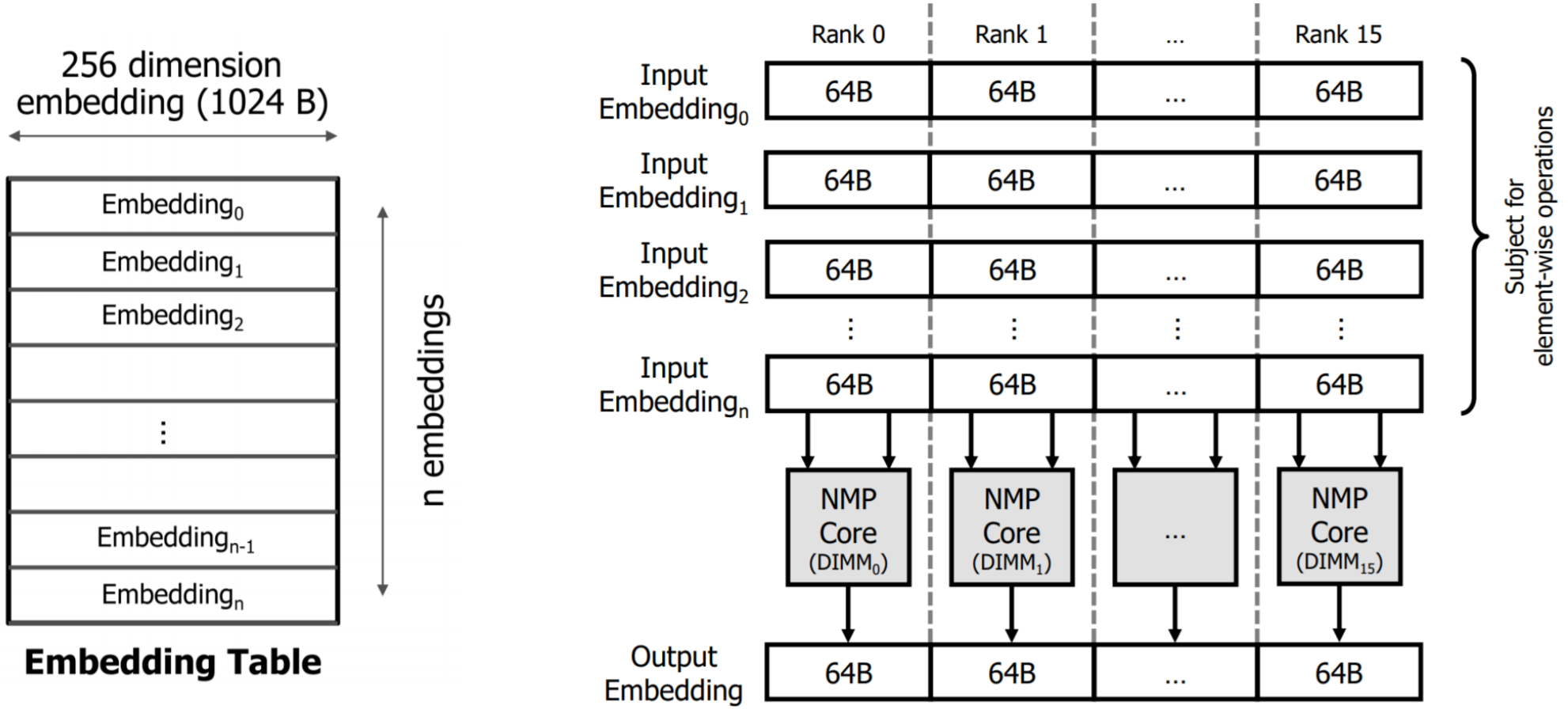


Conventional system VS TensorDIMM approach

TensorDIMM: How to work

Mapping embedded tables in DRAM

- Rank-level parallelislm for maximal bandwidth utilization

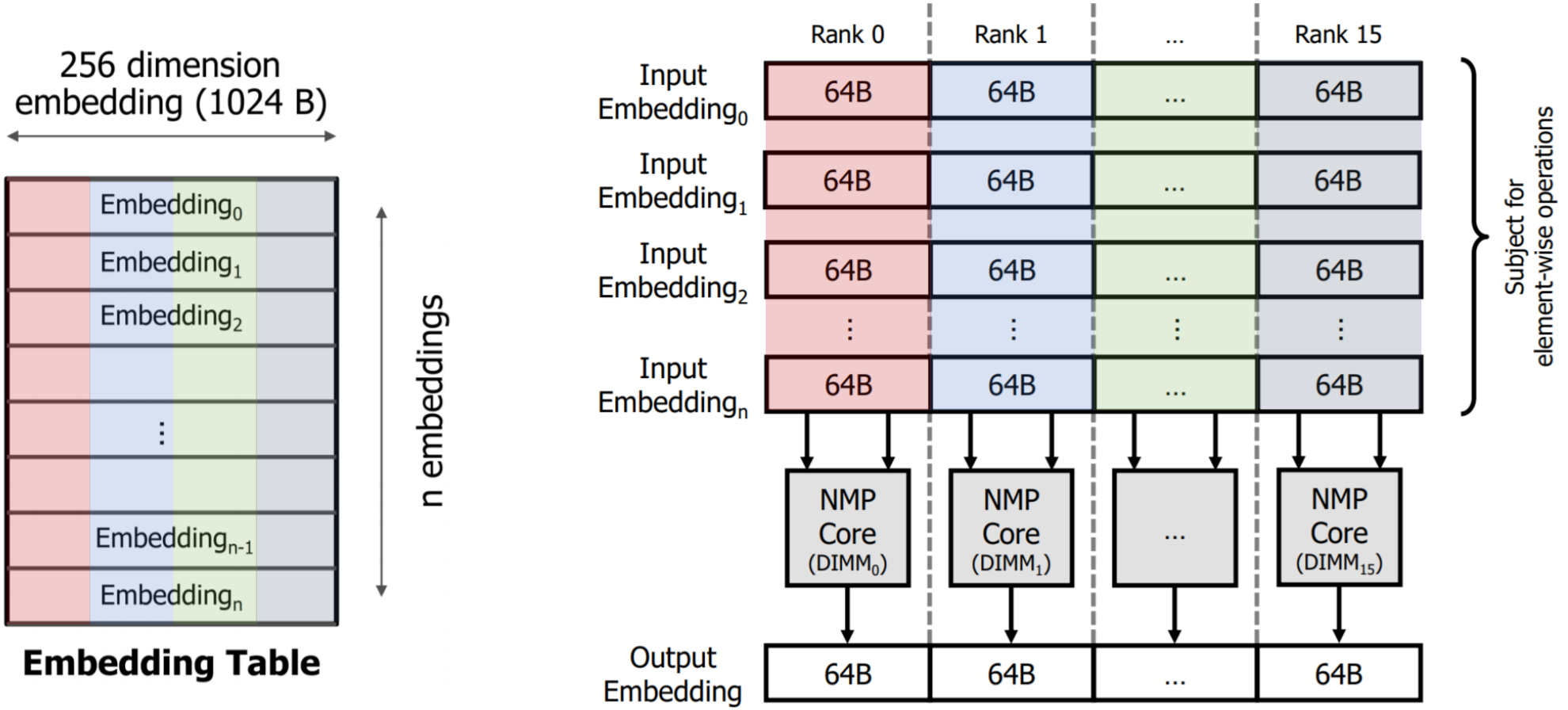


Mapping method in TensorDIMM

TensorDIMM: How to work

Mapping embedded tables in DRAM

- Rank-level parallelislm for maximal bandwidth utilization

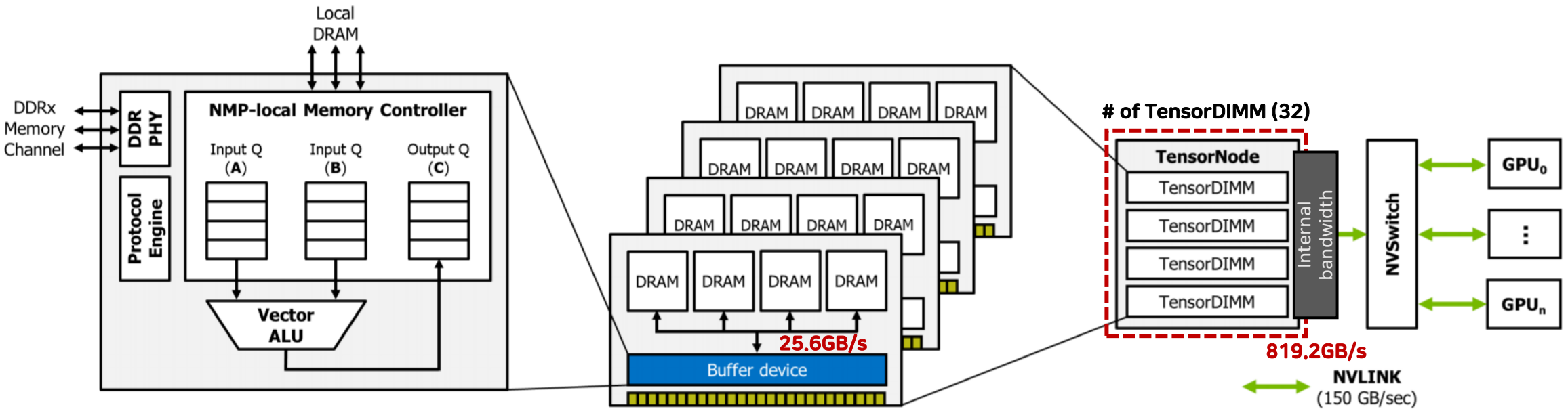


Mapping method in TensorDIMM

TensorDIMM: System architecture

Putting everything together

- Platform for scablable expansion of both memory bandwidth and capacity
- Utilize high-speed link (NVLINK) for inter-device communication



Overview of TensorDIMM Architecture

TensorDIMM: Evaluation

- **Combination of cycle-level simulation and emulation on real ML system**
  - It is hart to implement real hardware and software
    - Using computer architecture simulator (Gem5, DRAM-Sim, Etc.)
  - Cycle-level DRAM simulator
  - Utilize high-speed link (NVLINK) for inter-device communication

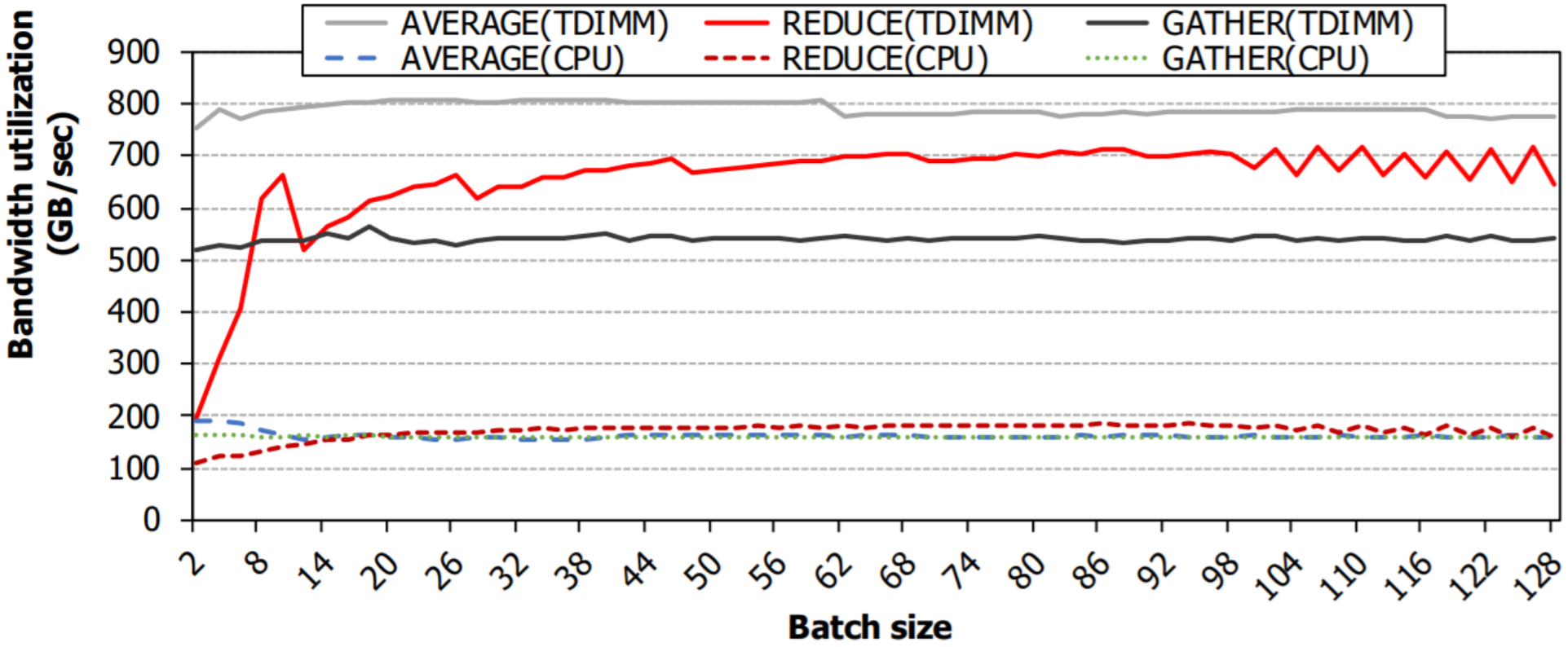
TensorDIMM: Evaluation

- **Combination of cycle-level simulation and emulation on real ML system**
  - It is hart to implement real hardware and software
    - Using computer architecture simulator (Gem5, DRAM-Sim, Etc.)
  - Cycle-level DRAM simulator
    - Memory bandwidth for embedding gather/reduction under restrict addess mapping
  - Utilize high-speed link (NVLINK) for inter-device communication

TensorDIMM: Evaluation

Bandwidth utilization

- Gather (Collect embedding line), Element (), Reduce (Reduction)
- Average 4x incresese compared to CPU acceleration



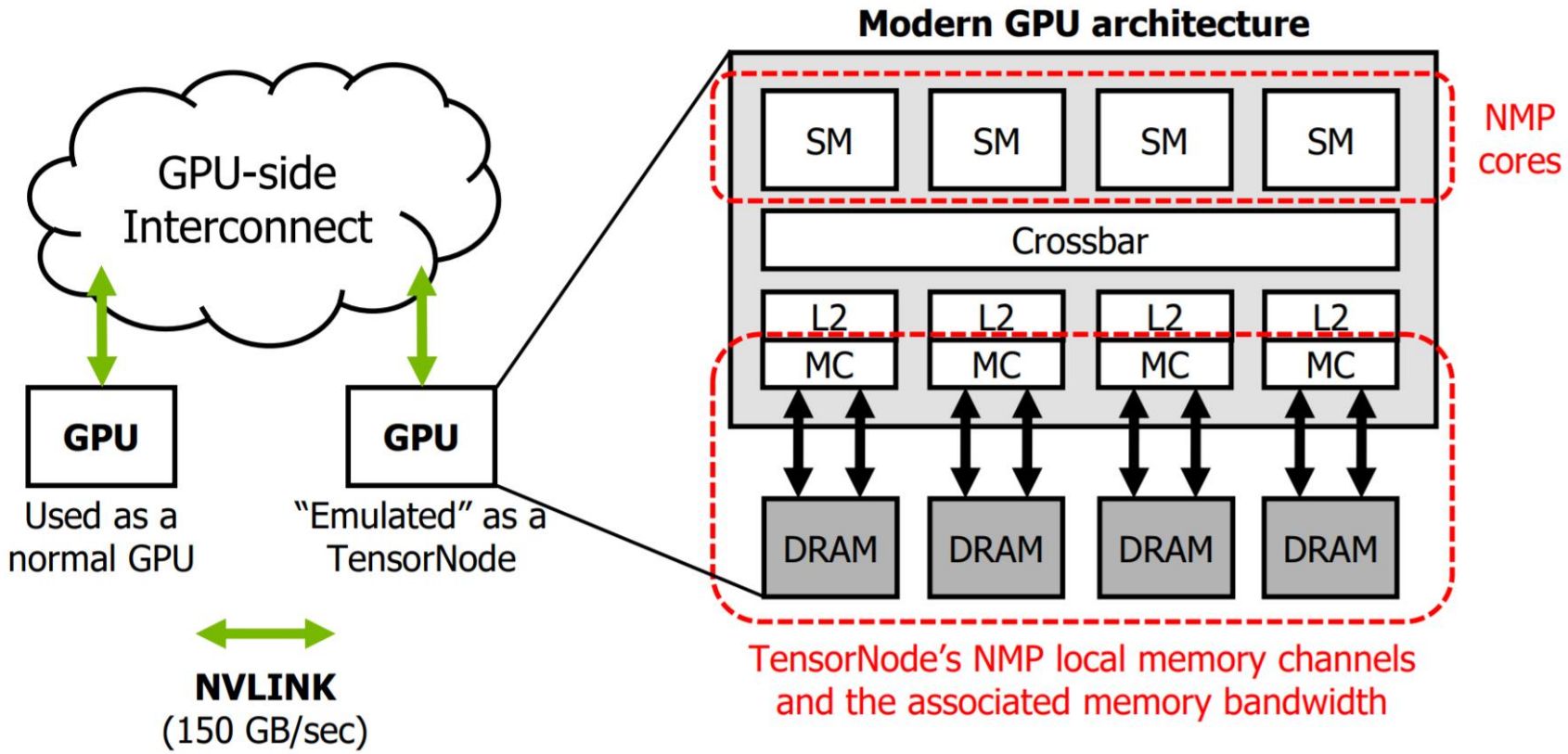
TensorDIMM: Evaluation

- **Combination of cycle-level simulation and emulation on real ML system**
  - It is hart to implement real hardware and software
    - Using computer architecture simulator (Gem5, DRAM-Sim, Etc.)
  - Cycle-level DRAM simulator
    - Memory bandwidth for embedding gather/reduction under restrict addess mapping
  - Utilize high-speed link (NVLINK) for inter-device communication
    - Intel's Math Kernel Library (MKL)
    - NVIDIA cuDNN/cuBLAS
    - In-house CUDA implmentaiotn of other layers
    - NVIDIA DGX-1V (8 V100 GPUs, Two Xeon E5-2698 v4)

TensorDIMM: Evaluation

TensorNode system modeling

- Proof of concept software prototype to emulate TensorDIMM
- No APIs to control NMP
  - In emulation, NMP is mapped to CUDA's streaming multiprocessor (SM)

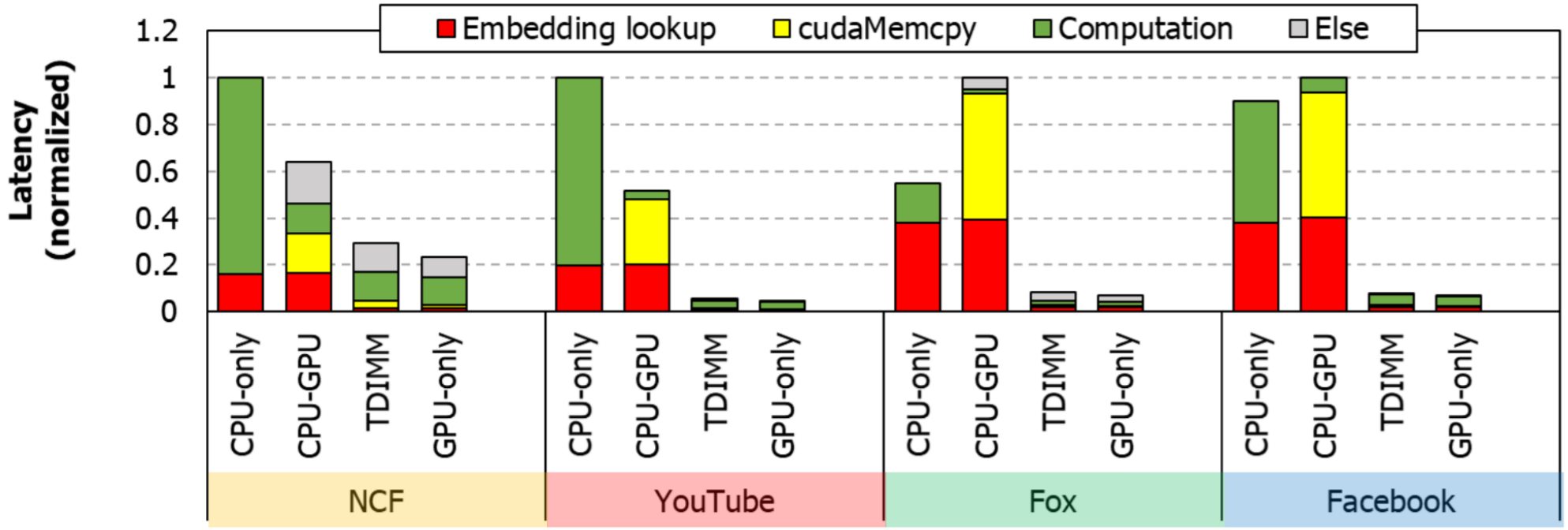


TensorDIMM: Evaluation

Latency breakdown

- Four system design point
  - CPU-only, Hybrid CPU-GPU, TensorDIMM, GPU-only (Oracle)
- TensorDIMM helps reduce both embedding/MLP latency
  - Overall 6~9x speedup compared to baseline

Network	Lookup tables	Max reduction	FC/MLP layers
NCF	4	2	4
YouTube	2	50	4
Fox	2	50	1
Facebook	8	25	6



Conclusion

- **More memory-centric**
  - Graphccore's IPU
    - Large on-chip memory

COLOSSUS MK2

the worlds most complex processor

59.4Bn transistors, TSMC 7nm @ 823mm^2

250TFlops AI-Float | 900MB In-Processor-Memory

1472 independent processor cores

8832 separate parallel threads

>8x step-up in system performance vs Mk1

GC200 - IPU

Reference

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

[2] In-Datacenter Performance Analysis of a Tensor Processing Unit, *ISCA*, 2017

감사합니다