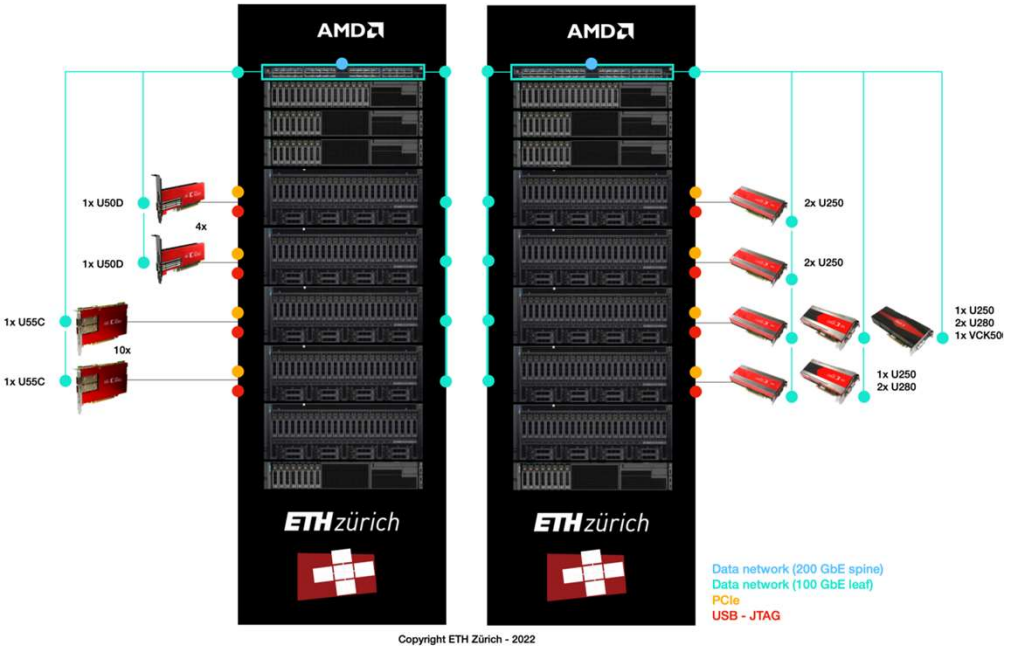
Systems@ETH zürich

Distributed Recommendation on Heterogeneous Clusters

Zhenhao He

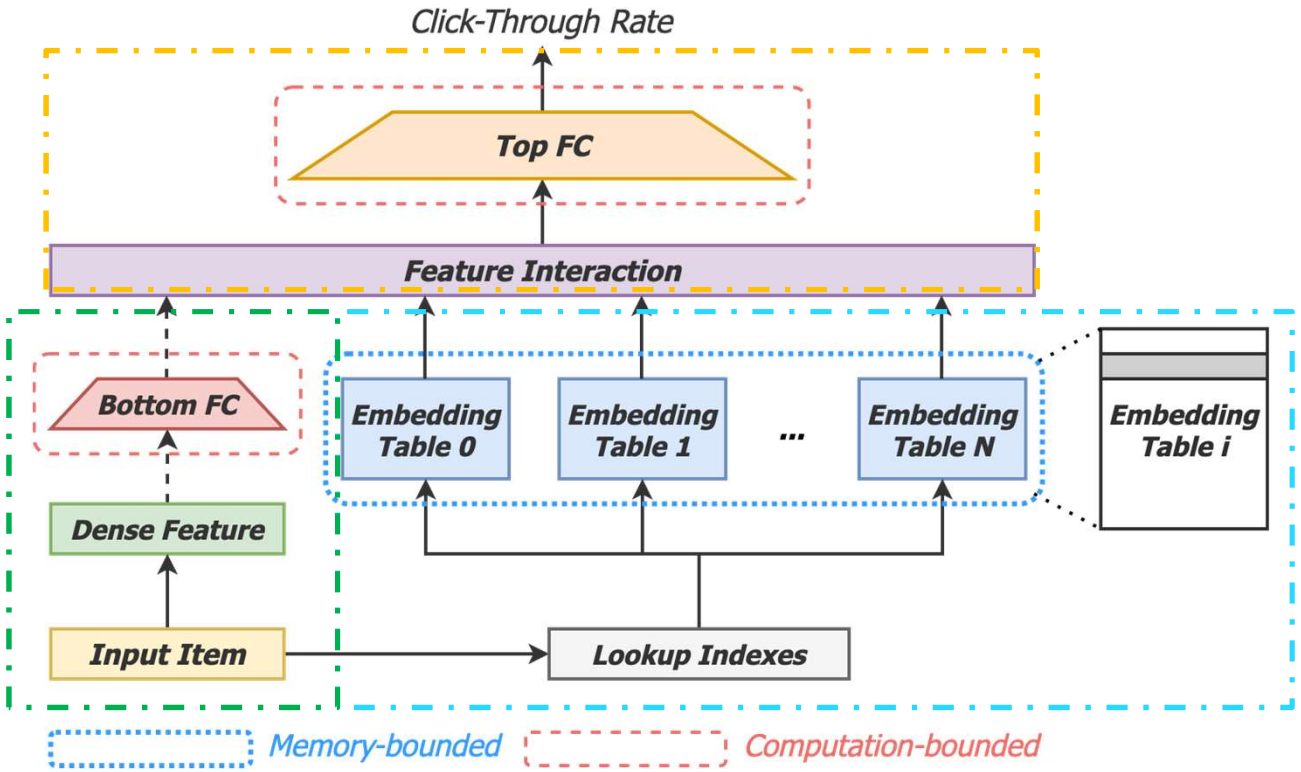


Personalized recommendation are everywhere

"Over 80% of machine learning inference cycles on Meta's datacenter fleet are devoted to recommendation filtering and ranking." [1]

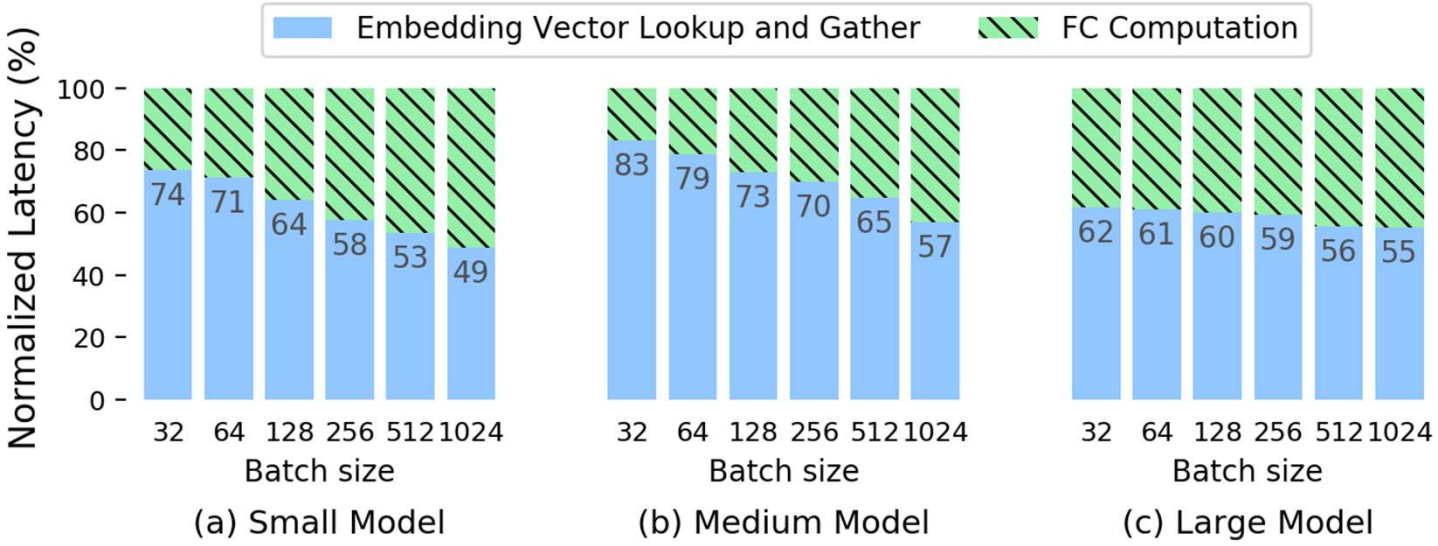
[1] Samuel Hsia et al. "Cross-Stack Workload Characterization of Deep Recommendation Systems." IISWC 2020.

Deep recommendation models involve intensive embedding table lookup operations



3

Workload profiling on Alibaba's real models

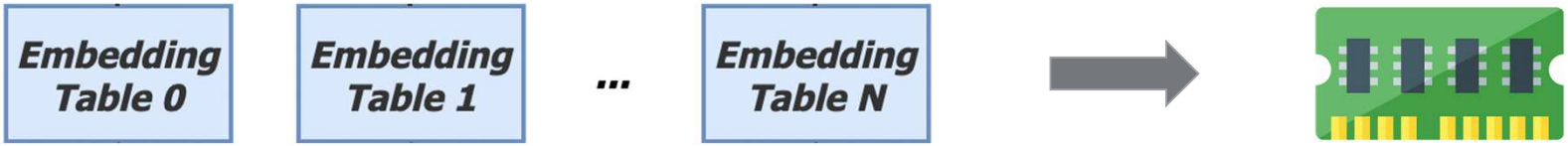


embedding lookup comprises more than half of the inference cycles

Why embedding table lookups are slow?

Many random DRAM accesses

- many embedding tables (tens to hundreds)
- each embedding vector is very short



An ideal recommendation inference system requires:

Fast embedding table lookups

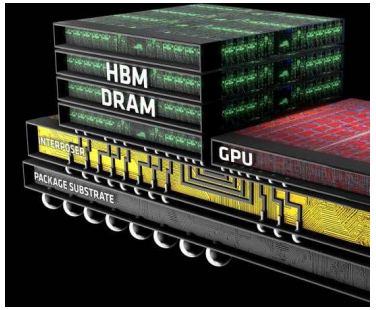
Fast DNN computation

Support different model architectures

CPUs and GPUs not ideal for recommendation inference…

Limited embedding table lookup performance

- many bank conflicts during random table lookups



Latency concern

- require batching to maximize throughput

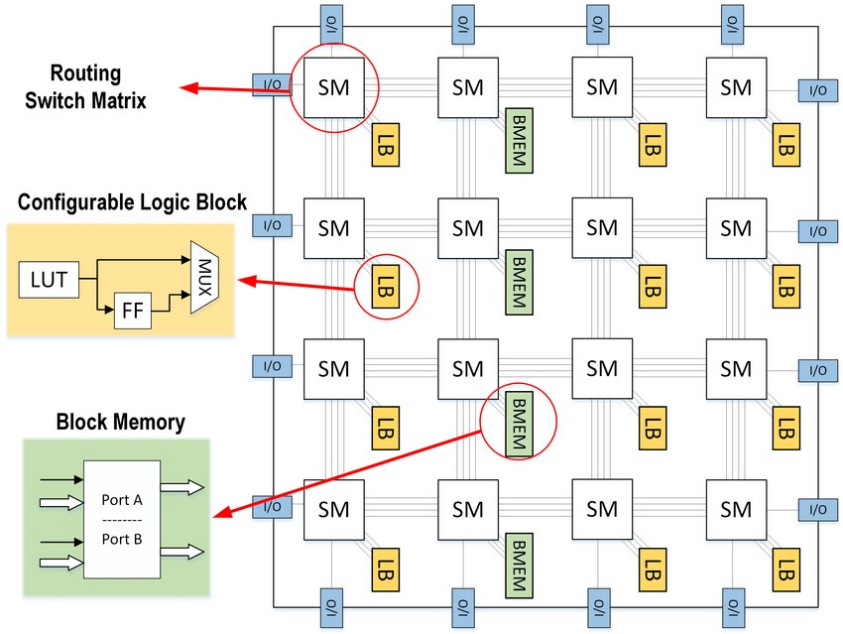
FPGA(s) for recommendation inference

FPGA memory for parallel embedding lookup

- HBM/DDR/BRAM

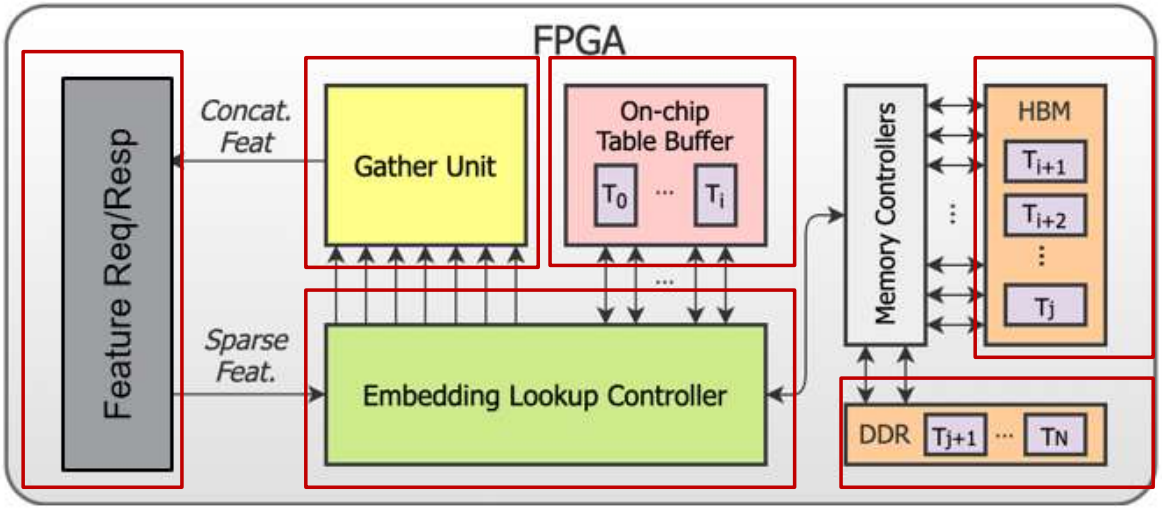
Parallel and pipelined data processing

- Embedding vector preprocessing
- DNN computation



How to design a great embedding lookup engine?

An FPGA equipped with High-bandwidth Memory (HBM) is ideal



When it fits into single machine

Embedding engine and DNN engine in single FPGA

FPGA out-perform CPU based solutions (MicroRec, MLSys'21)

- An order of magnitude faster in embedding lookup
- High speedup for total inference throughput
- Microsecond latency VS. millisecond latency

When it doesn’t fit into single machine…

Industrial embedding model can be large

- Tens of GB, hundreds of GB…
- Embedding tables distributed

DNN model parameter size can be large

- DNN computation distributed

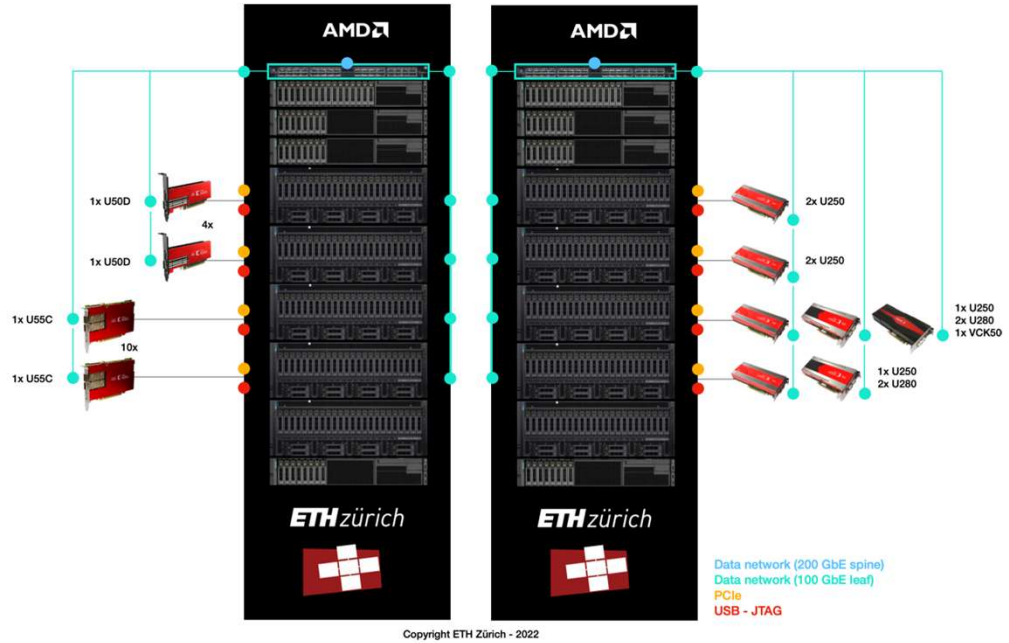
Network for partial results

- Send-recv? All-reduce?

Can we still achieve **low latency** and **high throughput**?

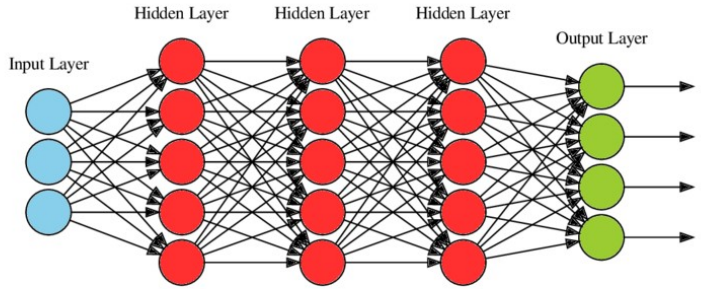
Recommender Inference on Heterogeneous Cluster

- Embedding tables and DNN computation distributed across an FPGA cluster
- FPGA networking
  - Low latency (RTT ~5us)
  - High bandwidth (100 Gbps)
- EasyNet and ACCL for networking support
  - EasyNet - TCP
  - ACCL - MPI

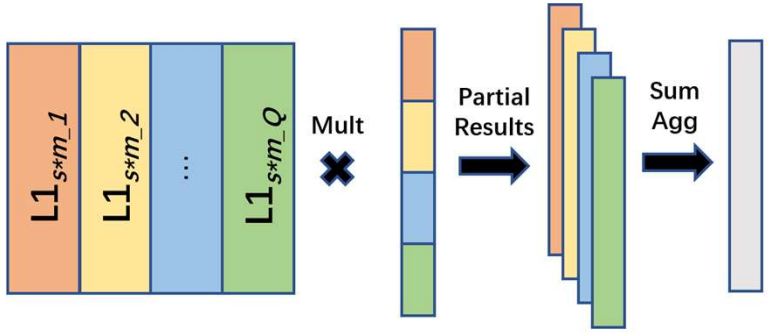


How to distribute embedding and DNN computation?

- Recommender DNN viewed as vector-matrix multiplication
  - Embedding layer as the input vector
  - Hidden layer as the input Matrix
- Distributed and parallel vector-matrix multiplication



Decomposition of Vector-Matrix Multiplication

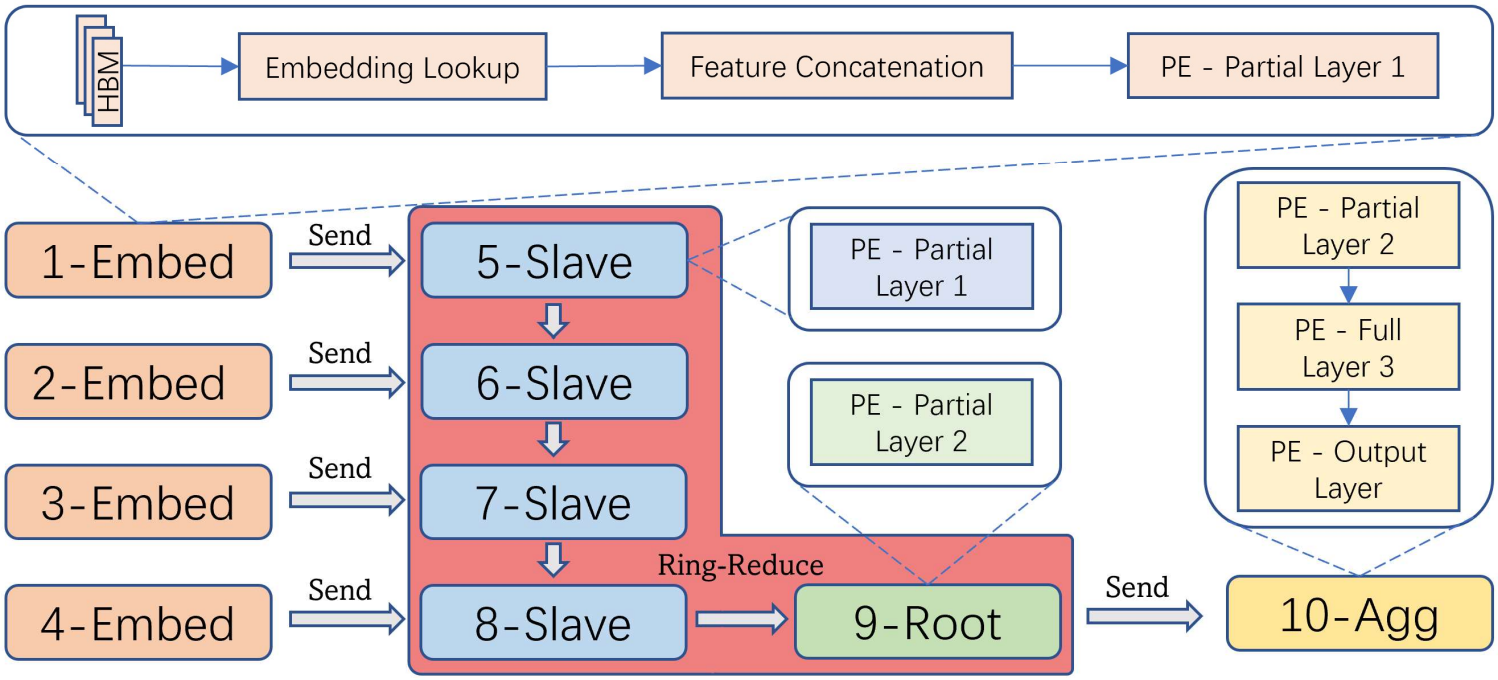


- Column decomposition of matrix
  - Each server owns partial matrix and embedding
  - Local computation of partial inputs
  - Reduction across all the server

Final decomposition considers balancing resource allocation across DNN layers

Distributed Recommendation Inference on FPGAs

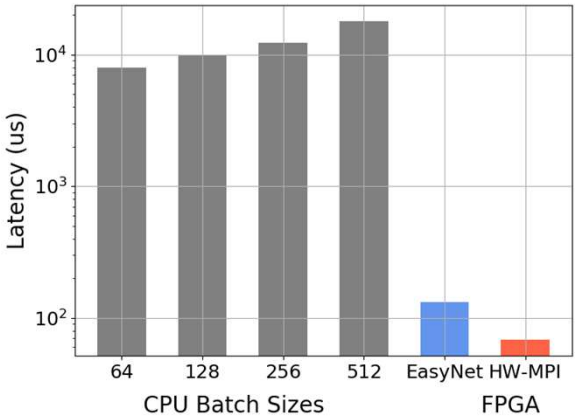
- Alibaba recommendation workload
- DNN distribution further optimized to reduce resource conflicts
- Streaming computation between layers
- Networking support with ACCL and EasyNet

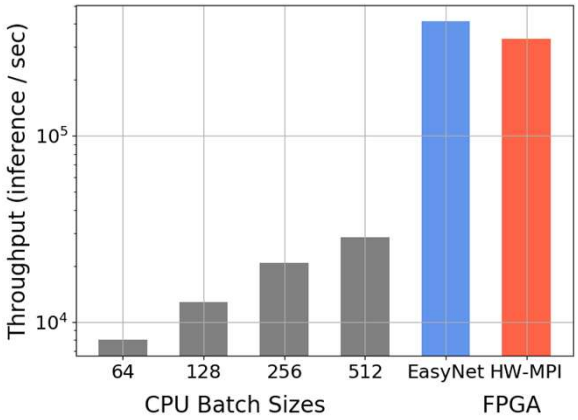


Evaluation – Ultra low latency and high throughput

Industrial model distributed across 10 FPGA nodes

CPU baseline with Intel Xeon Platinum 8259CL @ 2.50GHz, 32 vCPU, 256 GB DRAM





- FPGA significantly lower latency than CPU
- ACCL lower latency due to shorter critical path

- HW more than an order of magnitude higher throughput
- CPU throughput bounded to meet service level agreement

Take away messages

Distributed applications on top of FPGA cluster can show great performance.

Network adds minimal latency even across large number of nodes,

Due to streaming properties to overlap communication and computation.

Complicated communication pattern can be handled efficiently with ACCL and EasyNet.

17

Extension - Recommender beyond FPGA clusters

Distributed Recommender on heterogeneous cluster with CPUs, FPGAs and GPUs

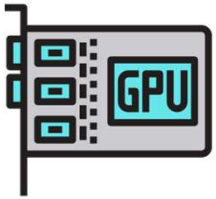
Utilize different configuration of heterogeneous devices for different recommender models

A different latency-throughput trade-off:
optimize throughput under service level agreement (~10 ms)

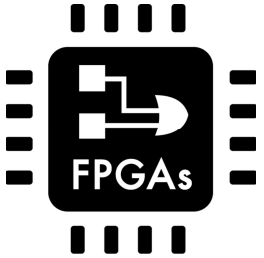
FleetRec (KDD'21)

Insight: take advantage of the strengths of different hardware

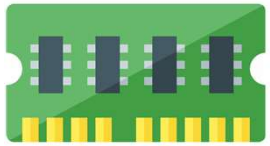
GPU for pure DNN computation

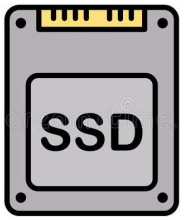


FPGA for parallel accessing small and medium embedding tables

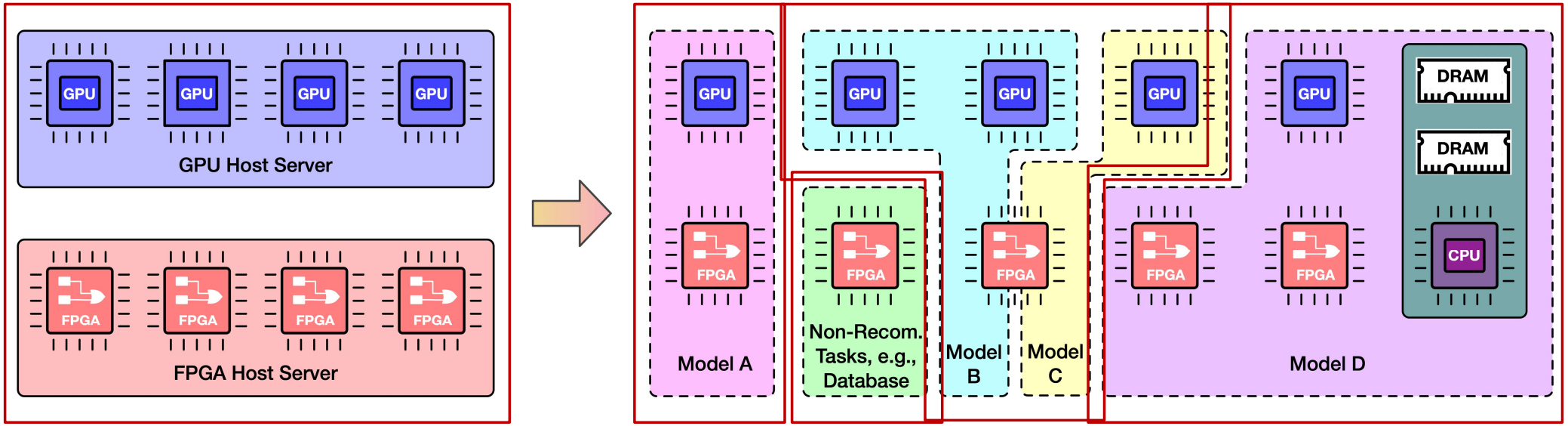


DRAM/SSD on CPU servers for few huge embedding tables





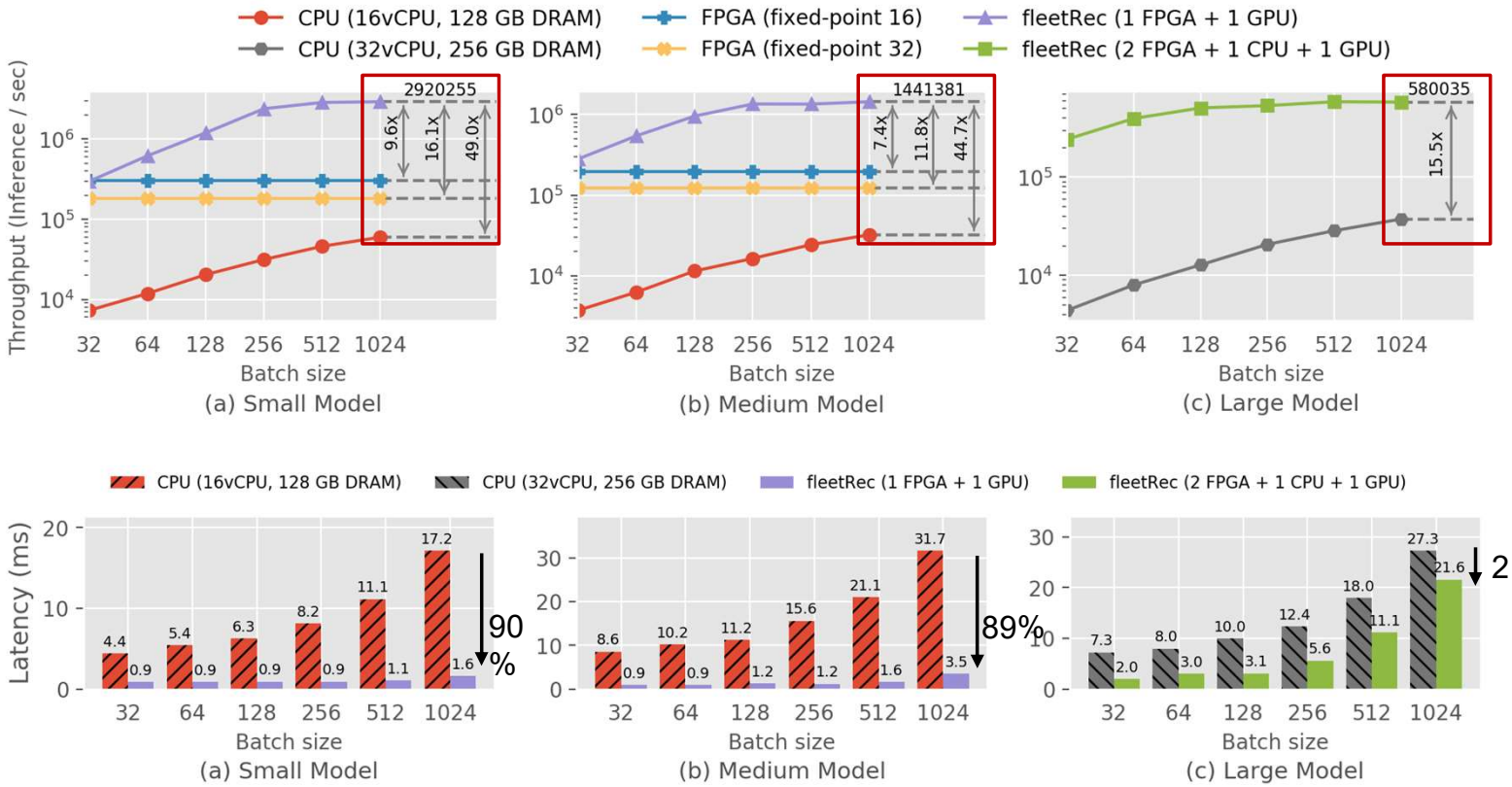
FleetRec: a high-performance recommendation inference system bridging CPUs, GPUs and FPGAs by network



FleetRec performance evaluation

FPGA baseline with MicroRec (MLSys'21)

CPU baseline with Intel Xeon Platinum 8259CL @ 2.50GHz



Discussion

- Many similar applications can be benefited by distributed computing
  - Other machine learning workloads, e.g., CNN mapped to a cluster
  - Distributed databases
- Various heterogenous devices, not limited to FPGAs
- To coordinate various devices, networking infrastructure is the enabler


Systems@ETH zürich