Software-Hardware Co-design of Heterogeneous SmartNIC System for Recommendation Model Inference and Training

Anqi Guo (University of Rochester, Boston University)
Yuchen Hao (Meta Platforms)
Chunshu Wu (Boston University)
Pouya Haghi (Boston University)
Zhenyu Pan (University of Rochester)
Min Si (Meta Platforms)
Dingwen Tao (Indiana University)
Ang Li (Pacific Northwest National Laboratory)
Martin Herbordt (Boston University)
Tong Geng (University of Rochester)
Personalized recommendation is everywhere

“35% of purchases on Amazon and 75% of videos on Netflix are powered by recommendation algorithms”
— McKinsey & Co
AI inference cycles in Facebook’s datacenter

Recommendation service account for over 80% of all AI inference cycles in Meta’s datacenter.

Meta’s datacenters perform 200+ trillion inferences every day.

Deep learning based recommendation model has evolved as single largest AI application in Meta.

Ref: The Architectural Implications of Facebook’s DNN-based Personalized Recommendation
Deep Learning Recommendation Model (DLRM)

Continuous (dense) features:
- Age
- Time of day

Categorical (sparse) features:
- User purchase history
- Book’s genre

Dense DNNs

Embedding Table

User

Visited
- Inkheart
- Moby Dick
- Hunger Games

Embedding aggregation

Item (Book)

Lookup

Genre

Sparse & Dense Integration

Sum

Concat

Ref: The Architectural Implications of Facebook’s DNN-based Personalized Recommendation
Large-scale distributed system for DLRM

Embedding tables can be Gigabytes to Terabytes

Exceed GPU’s HBM size

Requires large distributed system

Data parallel + Model parallel

Scalability Issue
Scalability Limits the development of DLRM

The growth of GPU’s HBM cannot keep up with the ever-growing DLRM size

System grows even larger

Even Worse Scalability Issue!

Communication Bottleneck:
• All-to-All communication

Memory Bandwidth Challenge:
• Large amount and frequent embedding access in GPU’s HBM

Computation Efficiency Challenge:
• DLRM’s Irregular computation and data reformatting

SmartNIC offers an opportunity
Regular Network Interface Card (NIC)

NIC in distributed system for communication as device

Any packet from the end host to the network and vice versa goes through the NIC

- The Physical Layer (L1)
- (Part of) the data link layer (L2)

http://www.cs.cornell.edu/~mt822/docs/smartNICs_lecture.pdf

- The host CPU generates, processes and delivers packets based on application request
- Packets are sent or received to the NIC over PCIe
- NIC transforms packets to bits and send them over the link

However, only communication devices
Smart Network Interface Card (SmartNIC)

SmartNIC = Regular NIC (Communication) + Computation Capability

SmartNICs are evolving with powerful valuable computation resources and heterogeneity

NIC + Arm
NIC + ASIC
NIC + FPGA
NIC + FPGA + Arm
NIC + SoC
SmartNIC offers an opportunity

mitigate network communication challenges in scale out data centers

\[ \downarrow \quad \text{computation support} \]

\[ \downarrow \quad \text{Capability of coupling communication and computation} \]

**However,** Simply adding SmartNICs to a distributed system only addresses point-to-point communication latency.

**How** to leverages SmartNIC resources?

overcome the critical challenges: communication bottleneck, memory bandwidth pressure, improving computational efficiency.
A software-hardware co-design of a heterogeneous SmartNIC system for Deep Learning Recommendation Model (DLRM)

A set of SmartNIC designs:
1. Cache systems:
   - local cache
   - remote cache
2. SmartNIC computation kernels
3. Graph Algorithm

- Exploits the locality of DLRM to reduce data movement
- Relieves memory access intensity
- Improves GPUs’ computation efficiency.

1+1 > 2
A software-hardware co-design of a heterogeneous SmartNIC system for Deep Learning Recommendation Model (DLRM)

A set of SmartNIC designs:
- Cache systems:
  - local cache
  - remote cache

SmartNIC computation kernels

Graph Algorithm
DLRM Data Power Law Distribution

A small fraction of embeddings results in most of the access

SmartNIC Design:
- Cache System (Buffer Local, Remote embedding)
  - Reduce communication workload
  - Relief memory bandwidth pressure
- Graph Algorithm
  - Clustering similar samples
Outline

• Graph Algorithm

• SmartNIC Cache System
  • Local Cache
  • Remote Cache

• Computation Kernels on SmartNIC
Graph Algorithm:

Grouping similar samples exploring data locality improve the overall system performance further.

Incidence Matrix | Hyper Graph | A Mini Batch of Similar Samples
Refresh Batch:

Issue: Samples with less common sparse features will always left

Solution:
• Downgrade factor
  Track similarity of generated mini batches
• Timing counter:
  Track samples waited too long for service

Batch Refresh with New Samples
Outline

• Graph Algorithm

• **SmartNIC Cache System**
  • Local Cache
  • Remote Cache

• Computation Kernels on SmartNIC
Local Cache on SmartNIC

Local Cache buffers local node embedding tables’ popular embedding vectors

Save local GPU HBM Bandwidth
Outline

• Graph Algorithm

• **SmartNIC Cache System**
  • Local Cache
  • **Remote Cache**

• Computation Kernels on SmartNIC
Remote Cache on SmartNIC

Remote Cache buffers embedding tables’ popular embedding vectors from remote node

Save Communications Workloads
Outline

• Graph Algorithm

• SmartNIC Cache System
  • Local Cache
  • Remote Cache

• Computation Kernels on SmartNIC
Computation Kernel on SmartNIC

Irregular computation:
- Data reshape
- Matrix flattening
- Matrix transposing

Gradients reduction:
- Local gradients reduction
- Global gradients reduction

Improves GPU computation efficiency
Evaluation

Forward Propagation

Graph algorithm + Remote cache

Backward Propagation

Local cache + SmartNIC computation

2X speedup on inference and 1.5X speedup on training.
Our Heterogeneous smartNIC system improves DLRM system scalability with higher training throughput and lower inference latency over GPU cluster.
Thank you!
Any questions and ideas are welcomed!

Contact: Anqi Guo anqiquo@bu.edu