An Extensible Software Transport Layer for GPU Networking

Yang Zhou

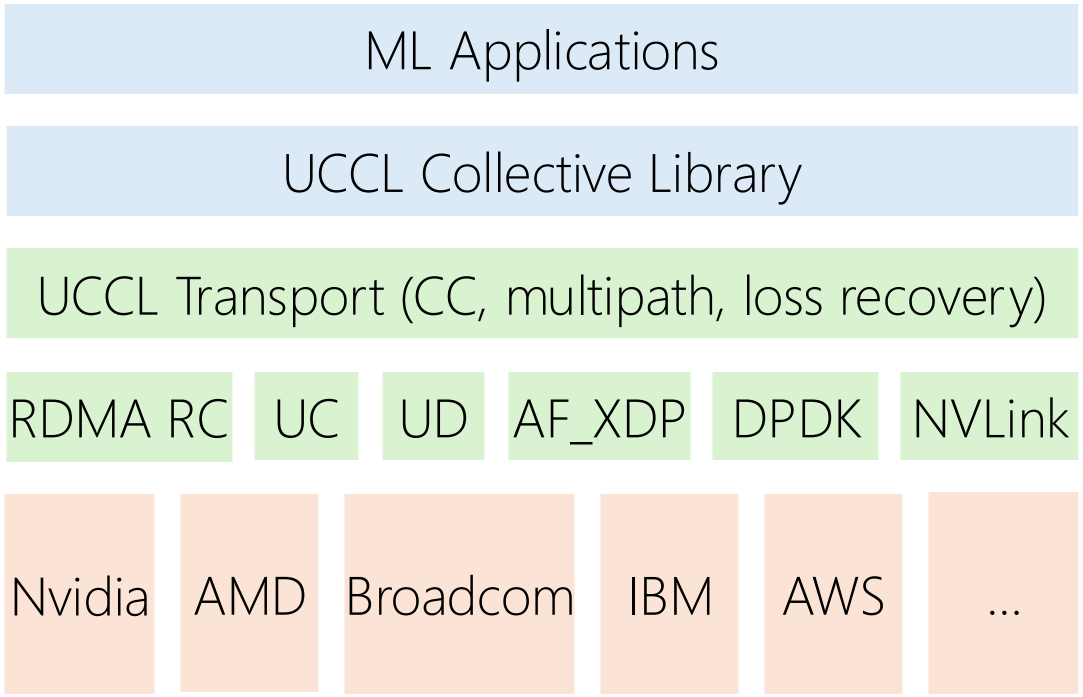
with: Zhongjie Chen, Ziming Mao, ChonLam Lao, Shuo Yang, Pravein Govindan Kannan, Jiaqi Gao, Yilong Zhao, Yongji Wu, Kaichao You, Fengyuan Ren, Zhiying Xu, Costin Raiciu, Ion Stoica

tinyurl.com/uccl-paper ⋄ github.com/uccl-project/uccl

June 2025

About UCCL Project

Building the fastest collective communication library (CCL)



ML Applications

UCCL Collective Library

UCCL Transport (CC, multipath, loss recovery)

RDMA RC | UC | UD | AF_XDP | DPDK | NVLink

Nvidia | AMD | Broadcom | IBM | AWS | ...

→ Rearchitecting CCL layer (keep NCCL API)

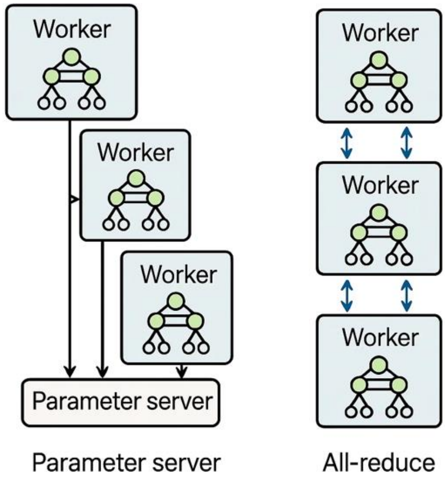
→ Rearchitecting transport layer (this talk)

- Up to 3.3× higher tput over NCCL on AWS

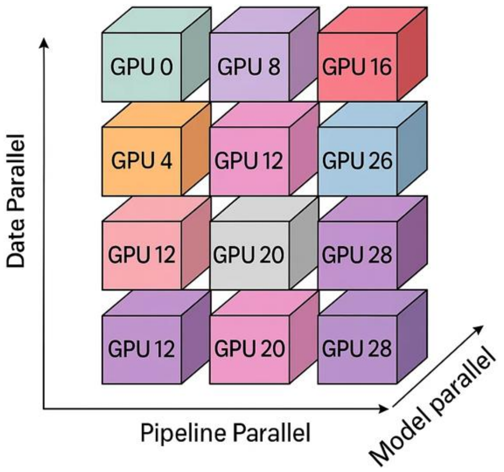
→ Supporting heterogeneous vendors

Open and collaborative platform: github.com/uccl-project/uccl

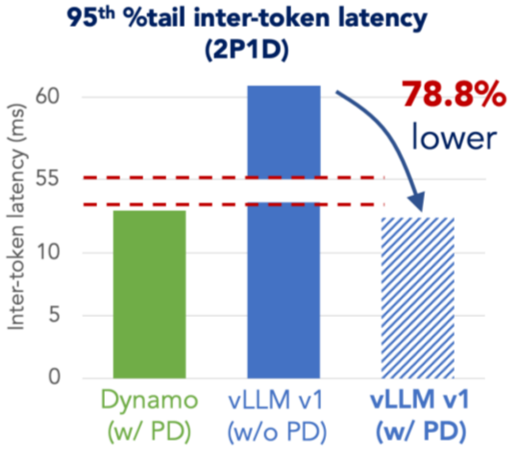
Fast Evolving ML Workloads



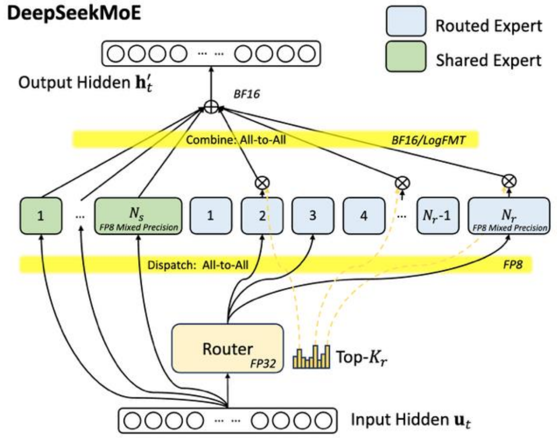
DNN training: parameter servers, allreduce



LLM training: allreduce, allgather, and reduce-scatter



PD disaggregation: P2P transfer



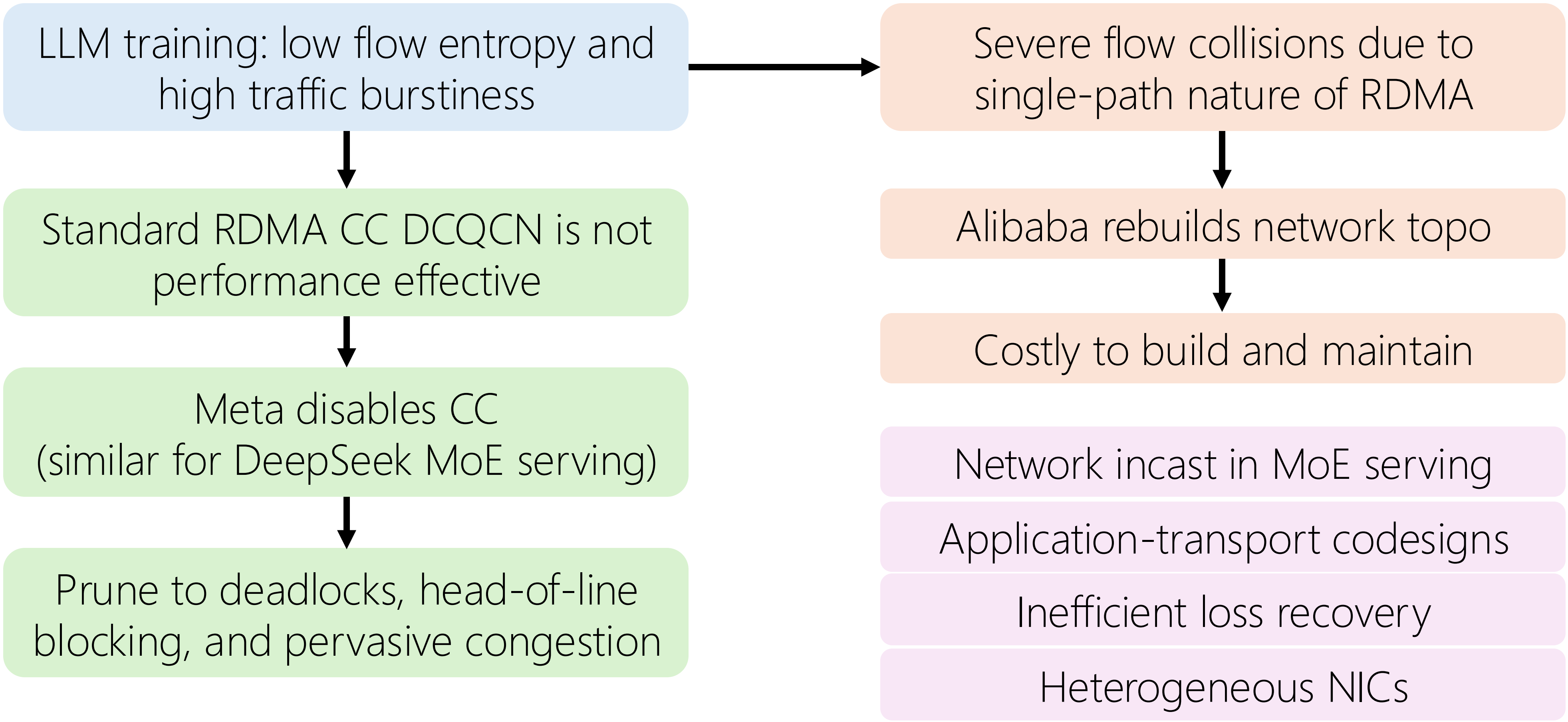
DeepSeek-V3 MoE: all-to-all like

~2015 ~2020 2024 2025 Time

Slowly Evolving Networking

Host transport on RDMA NICs is hard to adapt to better suit ML workloads

- Hardware changes are time-consuming



LLM training: low flow entropy and high traffic burstiness

Standard RDMA CC DCQCN is not performance effective

Meta disables CC (similar for DeepSeek MoE serving)

Prune to deadlocks, head-of-line blocking, and pervasive congestion

Severe flow collisions due to single-path nature of RDMA

Alibaba rebuilds network topo

Costly to build and maintain

Network incast in MoE serving

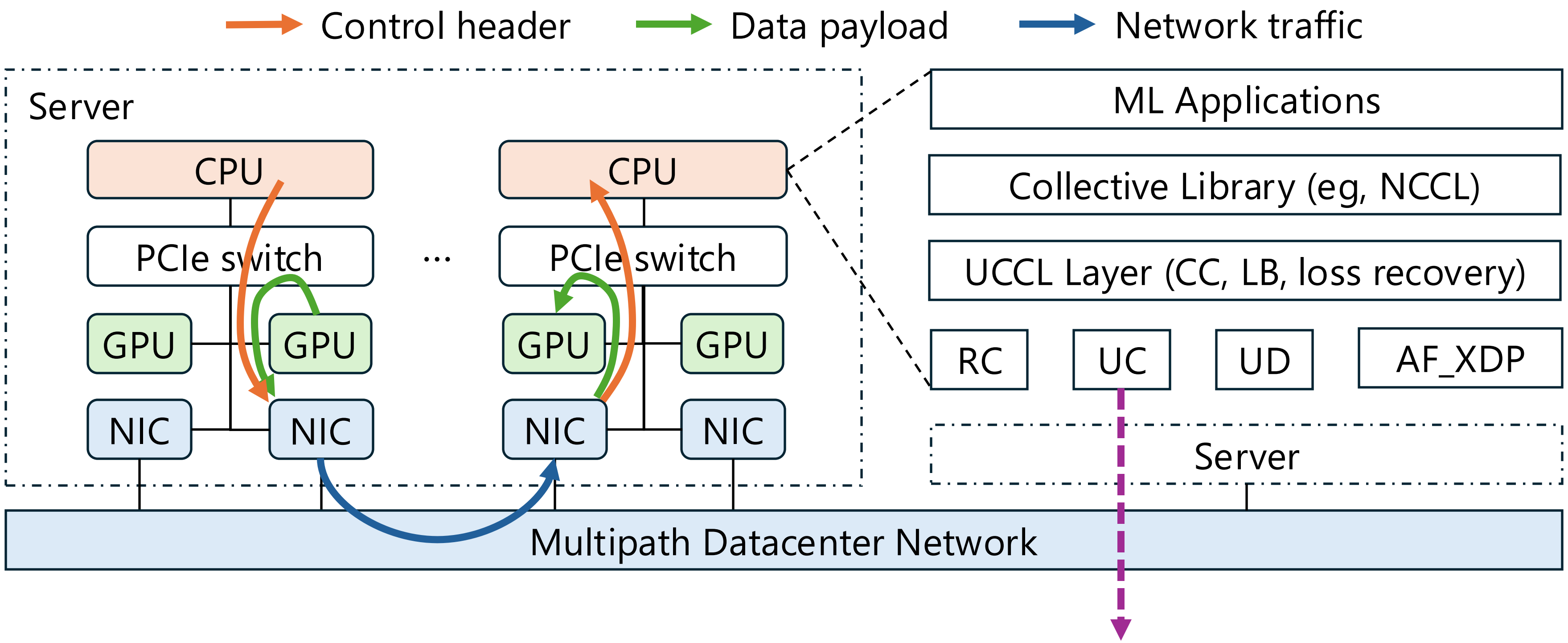
Application-transport codesigns

Inefficient loss recovery

Heterogeneous NICs

Overarching Problem: Network Extensibility

UCCL approach: a software-only extensible transport for GPU networking

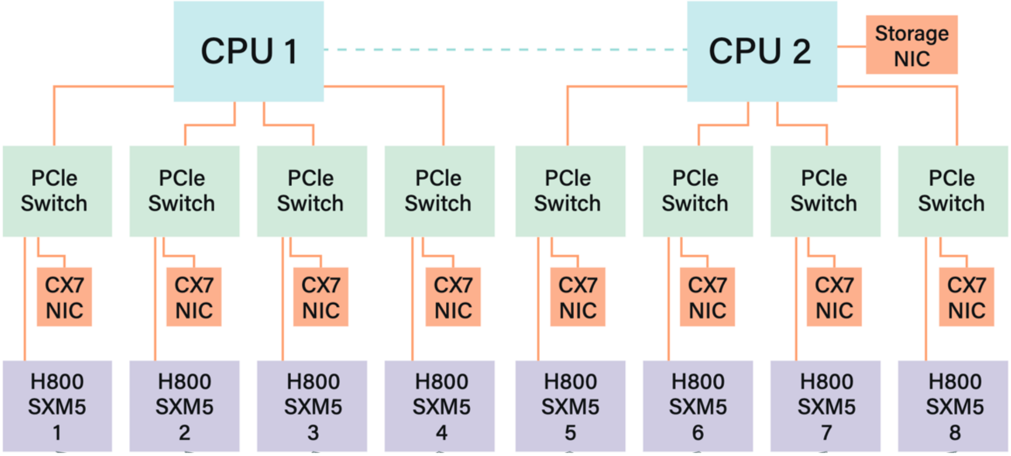


RC/UC/UD Queue Pair (QP)
= RDMA connection

UCCL Key Challenges

- How to **decouple** the data and control paths for existing RDMA NICs?
  - Eg, Nvidia NICs, Broadcom NICs, AWS EFA NICs

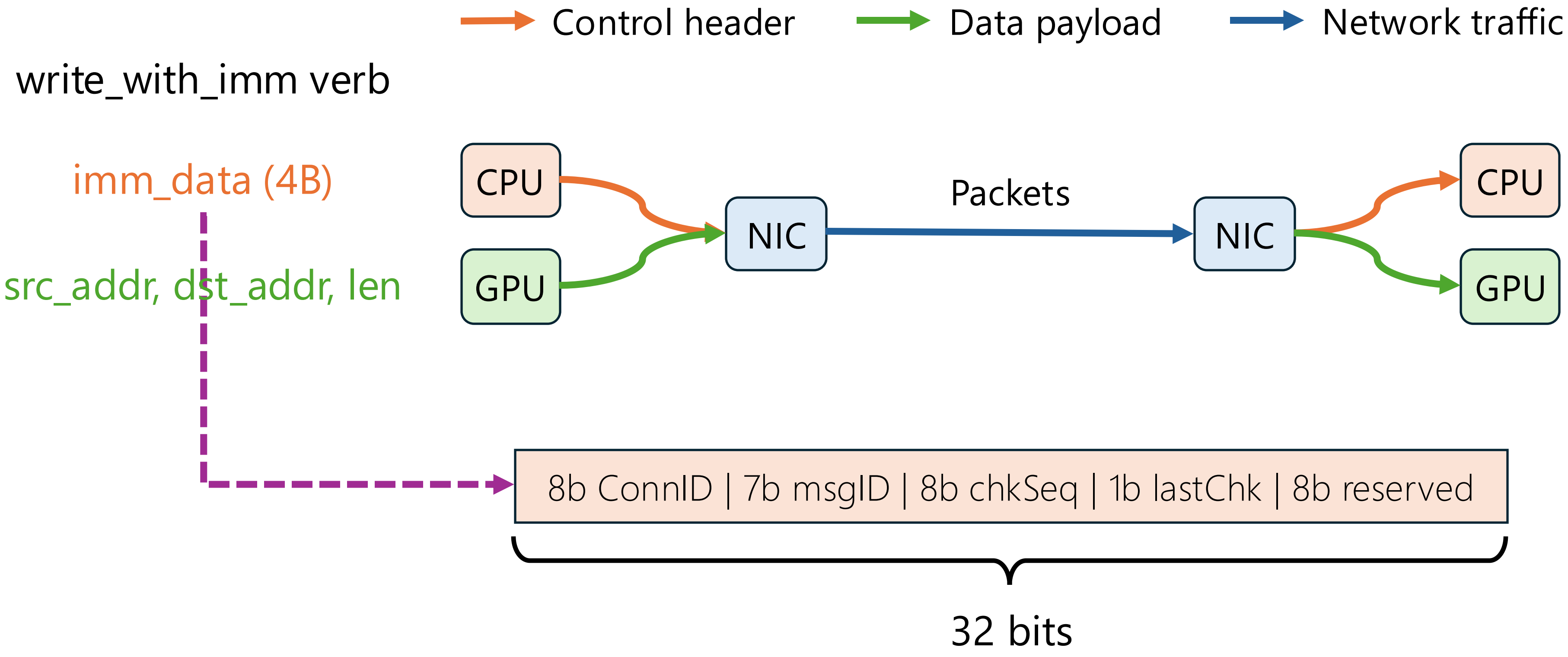
- How to achieve hardware-level **performance** for software control path?
  - Eg, 3.2 Tbps inter-server bandwidth



Technique #1: Decoupling Control & Data Path

Leveraging UC/RC QPs + RDMA write with immediate

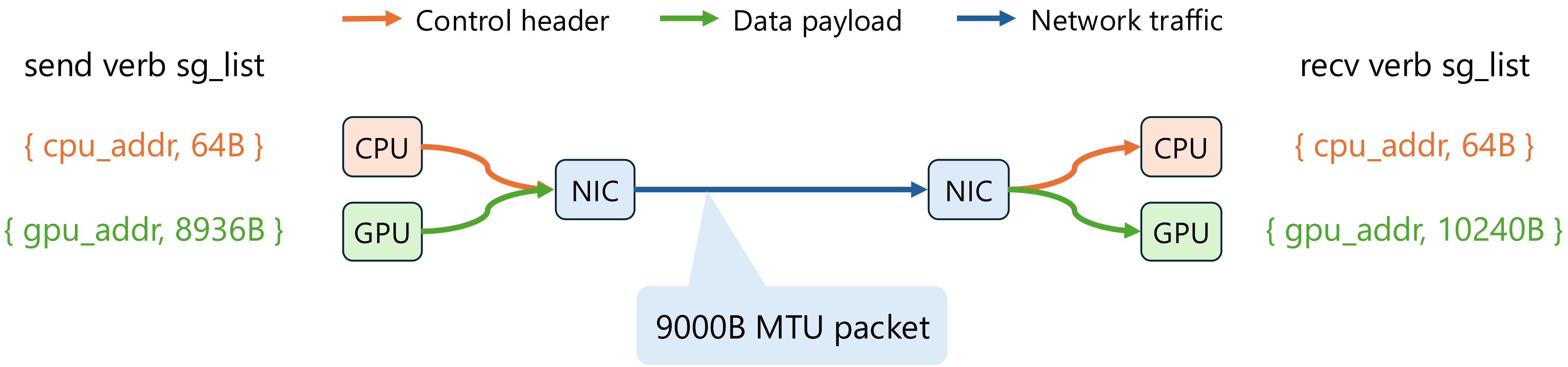
- Eg, for Nvidia and Broadcom NICs (that support UC or allow disabling RC's CC logic)



Technique #1: Decoupling Control & Data Path

Leveraging UD QPs + send/recv with scatter-gather list

- Eg, for AWS EFA NICs (that cannot disable RC's CC logic)



→ Control header

→ Data payload

→ Network traffic

send verb sg_list

{ cpu_addr, 64B }

{ gpu_addr, 8936B }

CPU

GPU

NIC

9000B MTU packet

NIC

CPU

GPU

recv verb sg_list

{ cpu_addr, 64B }

{ gpu_addr, 10240B }

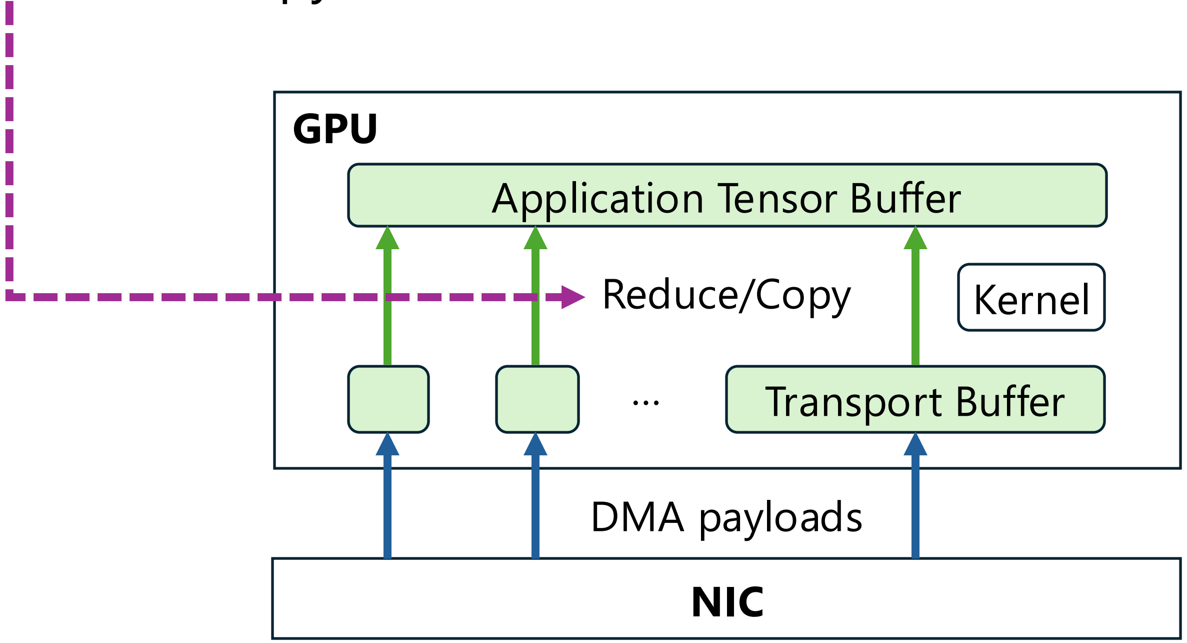
Technique #1: Decoupling Control & Data Path

Leveraging UD QPs + send/recv with scatter-gather list

- Eg, for AWS EFA NICs (that cannot disable RC's CC logic)

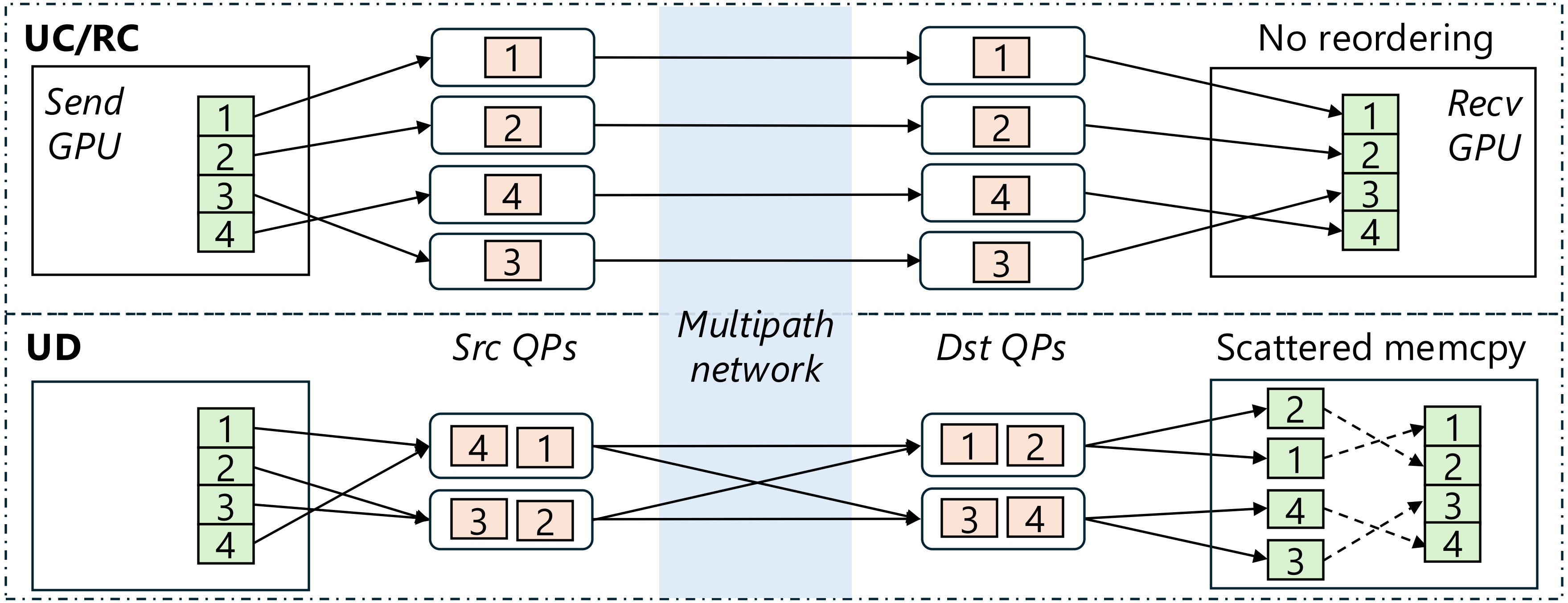
Handling out-of-order packet delivery

- Fusing scattered memcpy at the receiver GPU



Technique #1: Decoupling Control & Data Path

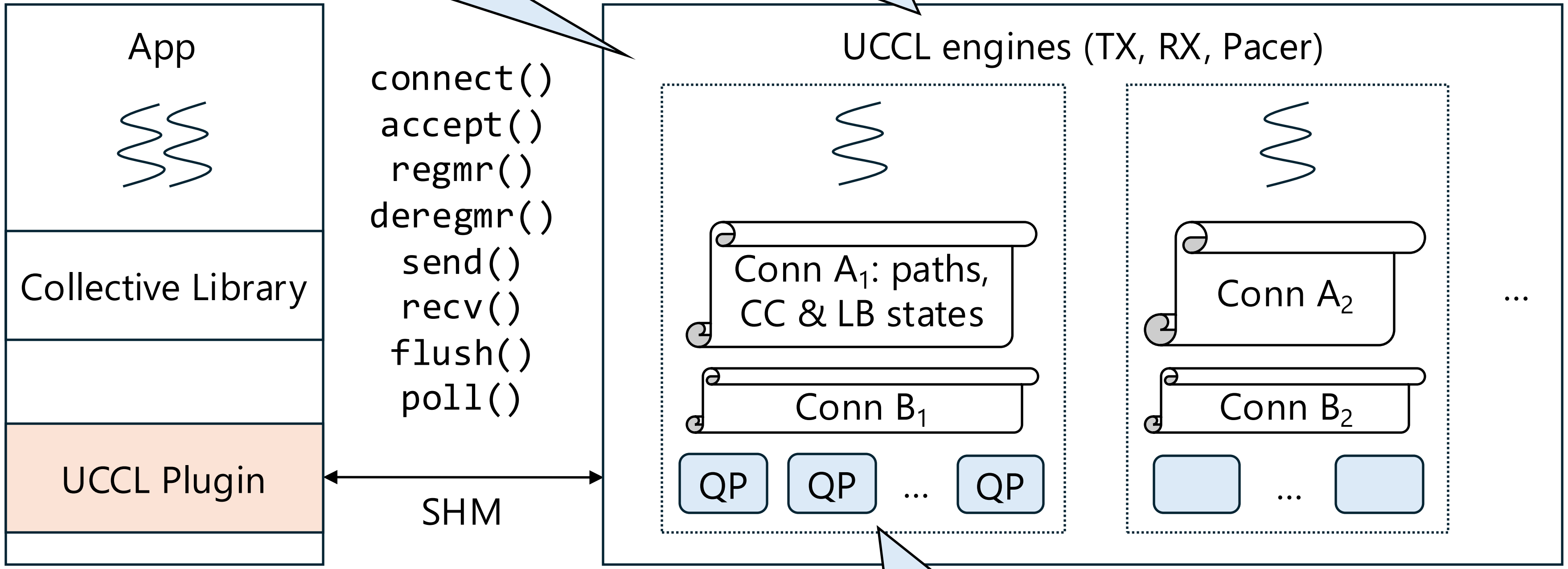
Multipathing with packet spraying



Technique #2: Efficient Software Transport

Control coalescing at 32KB

Run-to-completion execution



Connection splitting

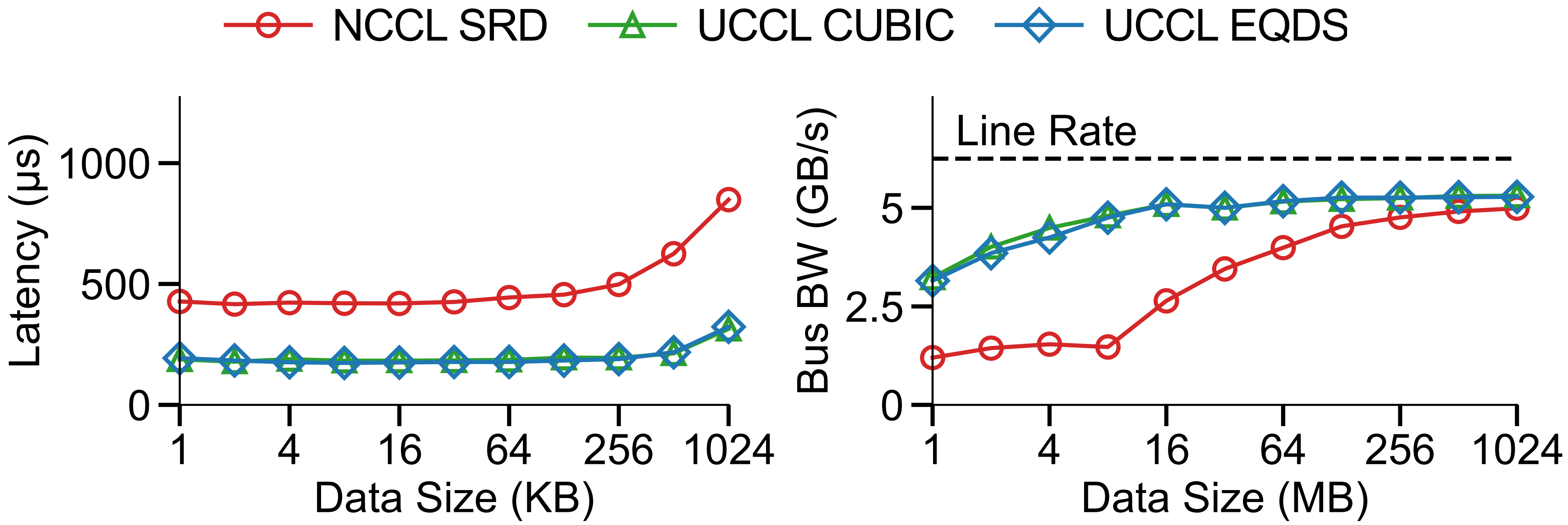
chained verbs posting, small message bypassing multipath

Implementation & Feature Support

- 27k LoC in C++
  - Drop-in replacement for NCCL applications
  - Packet spraying with 256 paths
  - Latency-based CC, receiver-driven CC
  - Efficient loss recovery by selective repeat
- Support both Nvidia and AMD GPUs
  - Future: AWS Trainium
- Support a variety of NIC vendors:
  - RDMA: Nvidia, Broadcom, AWS EFA
  - Non-RDMA: Nvidia, AWS ENA, IBM VirtIO

Evaluation: 4 AWS p4d all-to-all

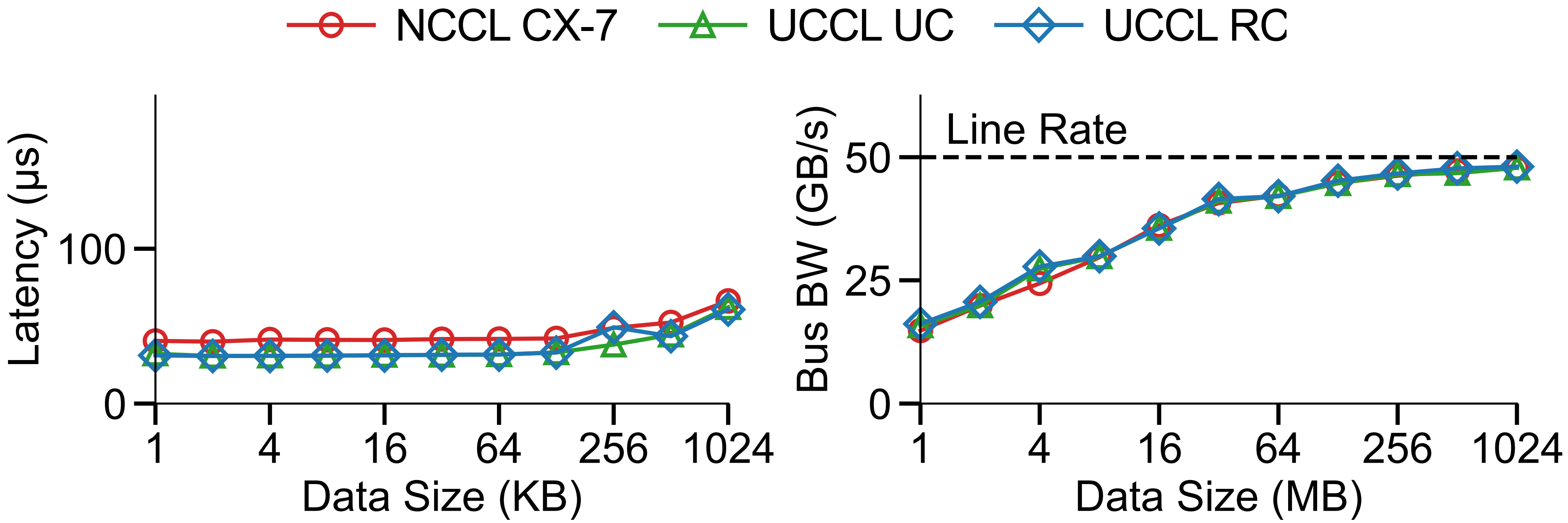
- 4×100G EFA NICs per node, Fattree over Ethernet
  - NVLink disabled to emulate larger testbed



UCCL achieves up to 3.2× higher performance over NCCL on AWS

Evaluation: 2 HGX all-to-all same-rack IB

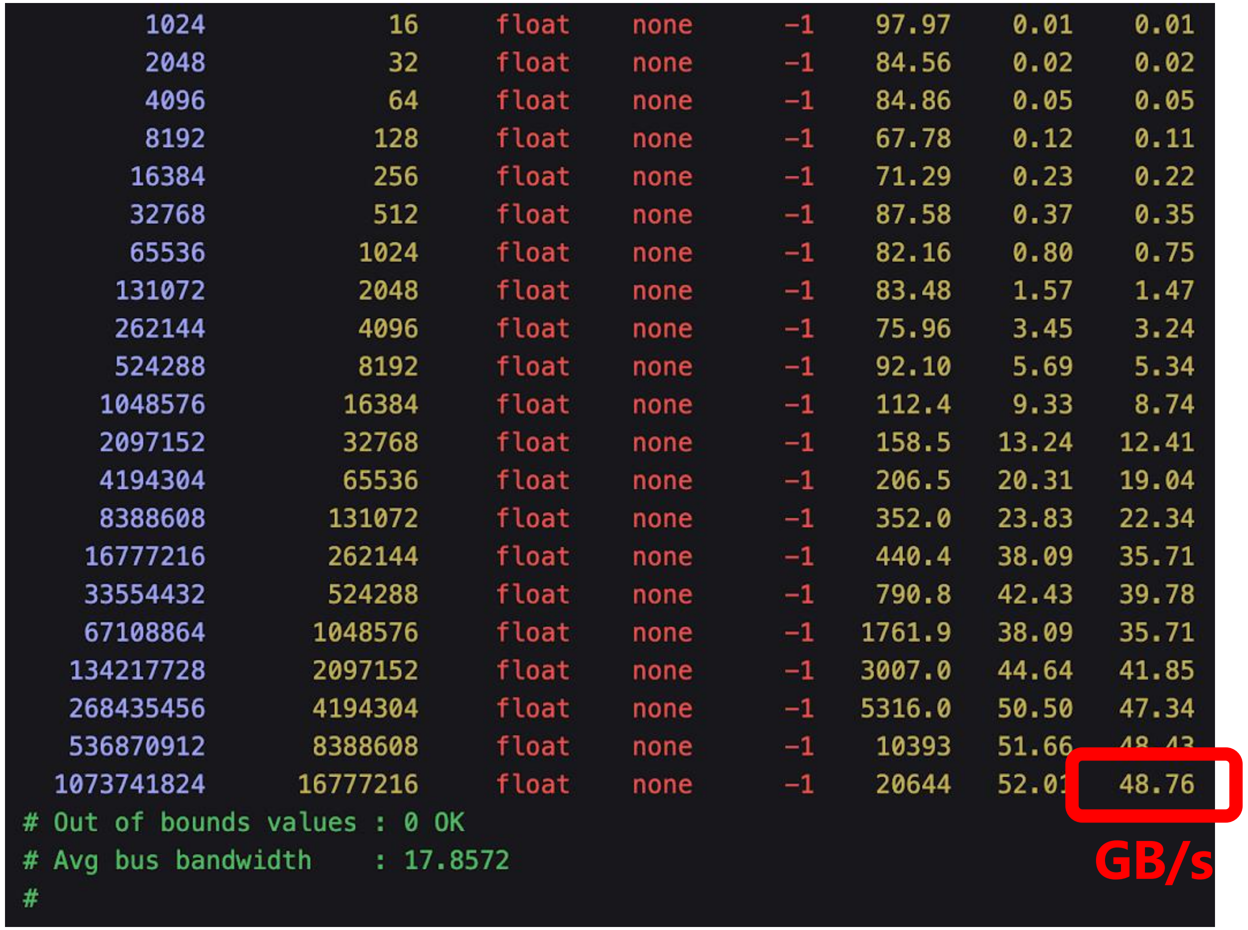
- 8×400G Nvidia CX-7 NICs per node, same rack over InfiniBand
  - NVLink disabled to emulate larger testbed



UCCL matches NCCL performances on ASIC-based NICs under same rack

Evaluation: 2 HGX all-to-all cross-rack RoCE

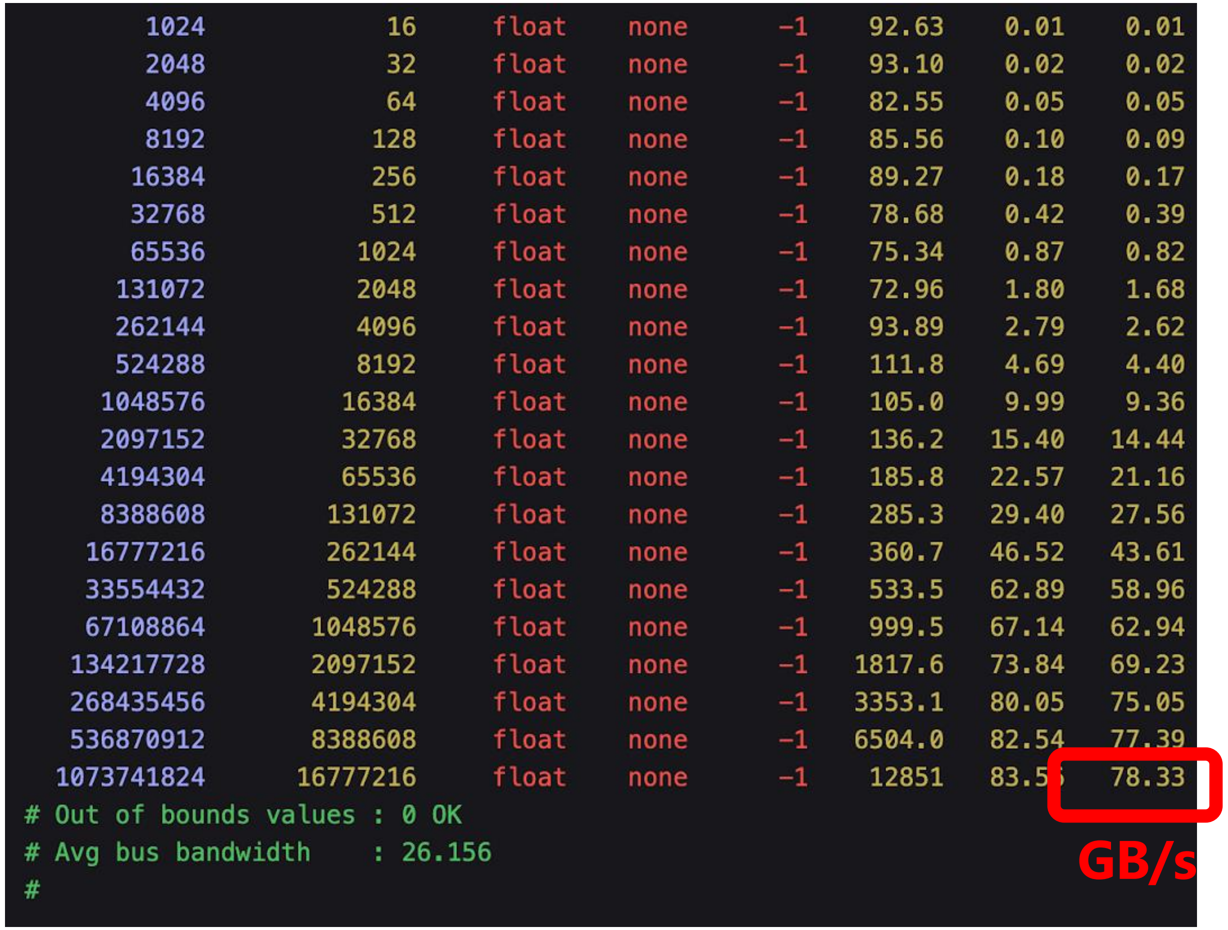
NCCL with RC QP scaling of 4



```
        1024            16   float    none    -1   97.97   0.01   0.01
        2048            32   float    none    -1   84.56   0.02   0.02
        4096            64   float    none    -1   84.86   0.05   0.05
        8192           128   float    none    -1   67.78   0.12   0.11
       16384           256   float    none    -1   71.29   0.23   0.22
       32768           512   float    none    -1   87.58   0.37   0.35
       65536          1024   float    none    -1   82.16   0.80   0.75
      131072          2048   float    none    -1   83.48   1.57   1.47
      262144          4096   float    none    -1   75.96   3.45   3.24
      524288          8192   float    none    -1   92.10   5.69   5.34
     1048576         16384   float    none    -1   112.4   9.33   8.74
     2097152         32768   float    none    -1   158.5  13.24  12.41
     4194304         65536   float    none    -1   206.5  20.31  19.04
     8388608        131072   float    none    -1   352.0  23.83  22.34
    16777216        262144   float    none    -1   440.4  38.09  35.71
    33554432        524288   float    none    -1   790.8  42.43  39.78
    67108864       1048576   float    none    -1  1761.9  38.09  35.71
   134217728       2097152   float    none    -1  3007.0  44.64  41.85
   268435456       4194304   float    none    -1  5316.0  50.50  47.34
   536870912       8388608   float    none    -1   10393  51.66  48.43
  1073741824      16777216   float    none    -1   20644  52.01  48.76
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 17.8572
#
```

GB/s

UCCL with 256 paths (UC QPs)



```
        1024            16   float    none    -1   92.63   0.01   0.01
        2048            32   float    none    -1   93.10   0.02   0.02
        4096            64   float    none    -1   82.55   0.05   0.05
        8192           128   float    none    -1   85.56   0.10   0.09
       16384           256   float    none    -1   89.27   0.18   0.17
       32768           512   float    none    -1   78.68   0.42   0.39
       65536          1024   float    none    -1   75.34   0.87   0.82
      131072          2048   float    none    -1   72.96   1.80   1.68
      262144          4096   float    none    -1   93.89   2.79   2.62
      524288          8192   float    none    -1   111.8   4.69   4.40
     1048576         16384   float    none    -1   105.0   9.99   9.36
     2097152         32768   float    none    -1   136.2  15.40  14.44
     4194304         65536   float    none    -1   185.8  22.57  21.16
     8388608        131072   float    none    -1   285.3  29.40  27.56
    16777216        262144   float    none    -1   360.7  46.52  43.61
    33554432        524288   float    none    -1   533.5  62.89  58.96
    67108864       1048576   float    none    -1   999.5  67.14  62.94
   134217728       2097152   float    none    -1  1817.6  73.84  69.23
   268435456       4194304   float    none    -1  3353.1  80.05  75.05
   536870912       8388608   float    none    -1  6504.0  82.54  77.39
  1073741824      16777216   float    none    -1   12851  83.56  78.33
# Out of bounds values : 0 OK
# Avg bus bandwidth    : 26.156
#
```

GB/s

UCCL outperforms NCCL by 1.6× on ASIC-based NICs across racks

Dev Plan

- Dynamic membership with GPU servers joining and exiting
- GPU-initiated P2P communication (eg, IBGDA)
  - For MoE all-to-all and PD disaggregation
  - Generic to NIC vendors like AWS EFA and Broadcom, and GPU vendors like AMD
- Rearchitecting NCCL to unleash network hardware capability
  - Scalable and efficient CPU proxy
  - Low-cost async collectives with compute-communication ordering guarantee
  - Device kernels in vendor-agnostic Triton language

- We would like to hear about your feature needs!

Conclusion

UCCL: building the fastest collective communication library

- Network transport layer, CCL layer, heterogeneous vendors, and more
- Open and collaborative platform---talk with us in the poster session

tinyurl.com/uccl-paper

github.com/uccl-project/uccl

Thank you!

yangzhou.rpc@gmail.com/berkeley.edu