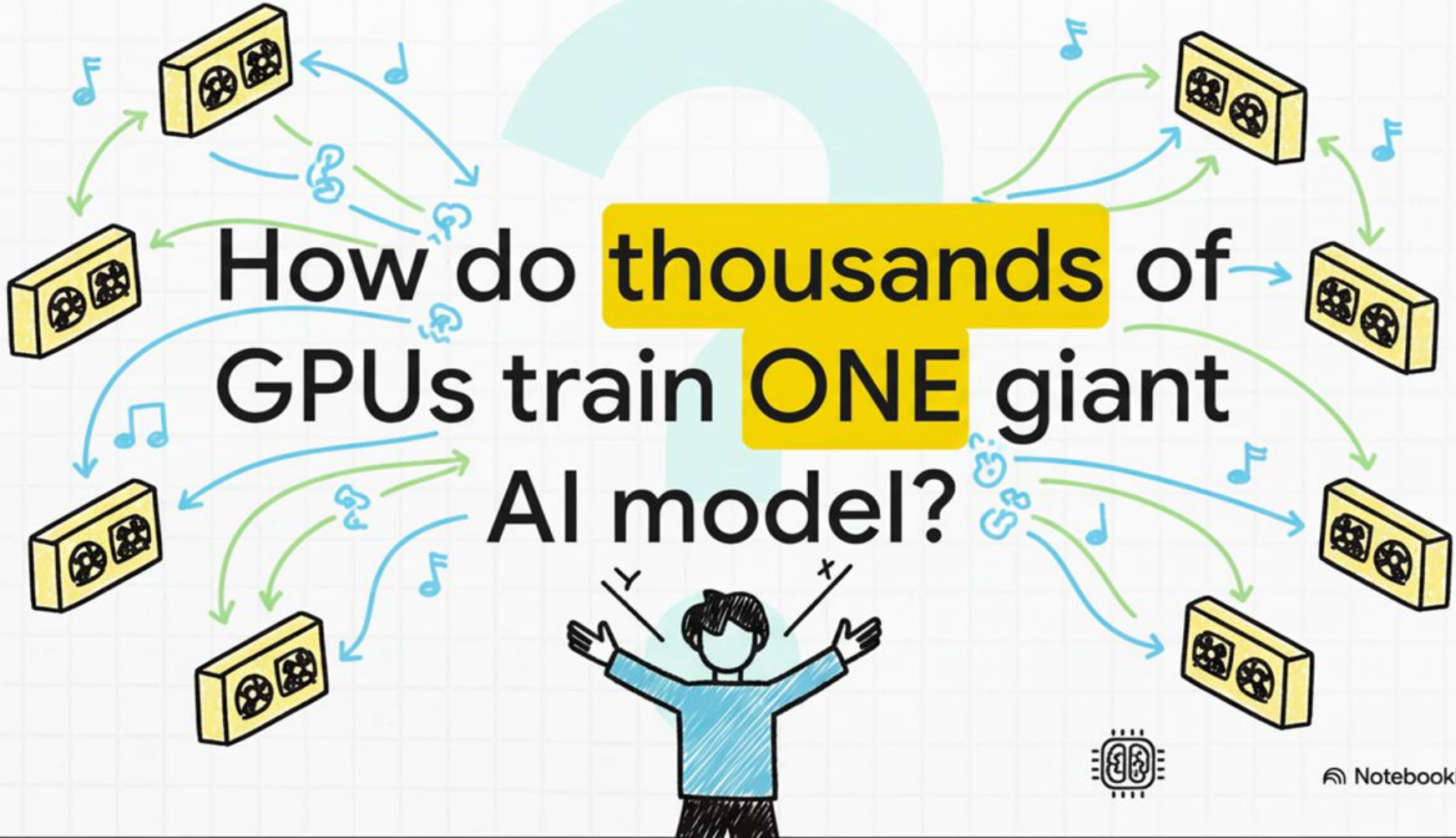
Demystifying NCCL: An In-depth Analysis of GPU Communication Protocols and Algorithms

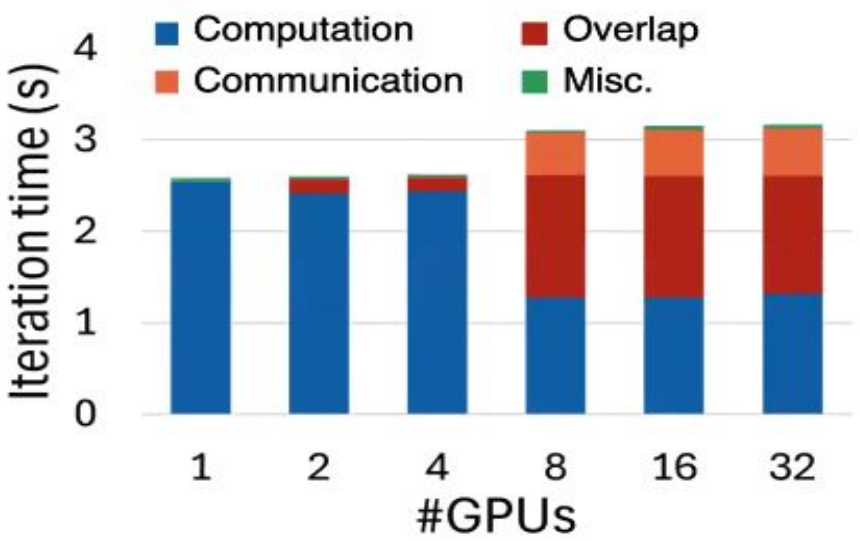
Authors: Zhiyi Hu, Siyuan Shen, Tommaso Bonato, Sylvain Jeaugey, Cedell Alexander, Eric Spada, James Dinan, Jeff Hammond, Torsten Hoefler

Slides adapted from Hot Interconnects Symposium


How do thousands of GPUs train ONE giant AI model?
NotebookLM

GPU Communication

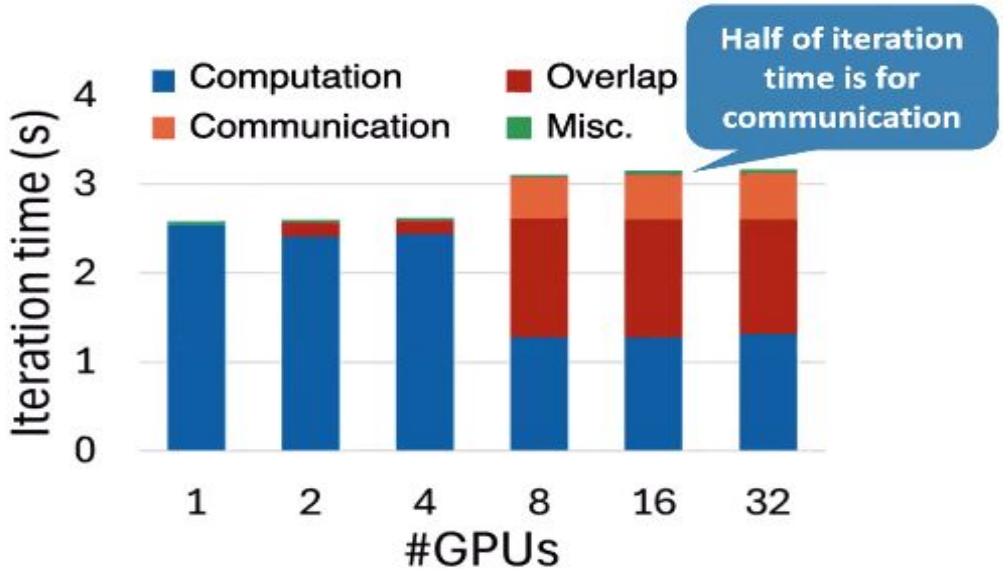
GPU Communication Bottleneck



Iteration time breakdown of the full fine-tuning method on a 1B parameters Transformer decoder model [1]

[1] N. Alnaasan et al., "Characterizing Communication in Distributed Parameter-Efficient Fine-Tuning for Large Language Models," HOTI 2024, pp. 11–19.

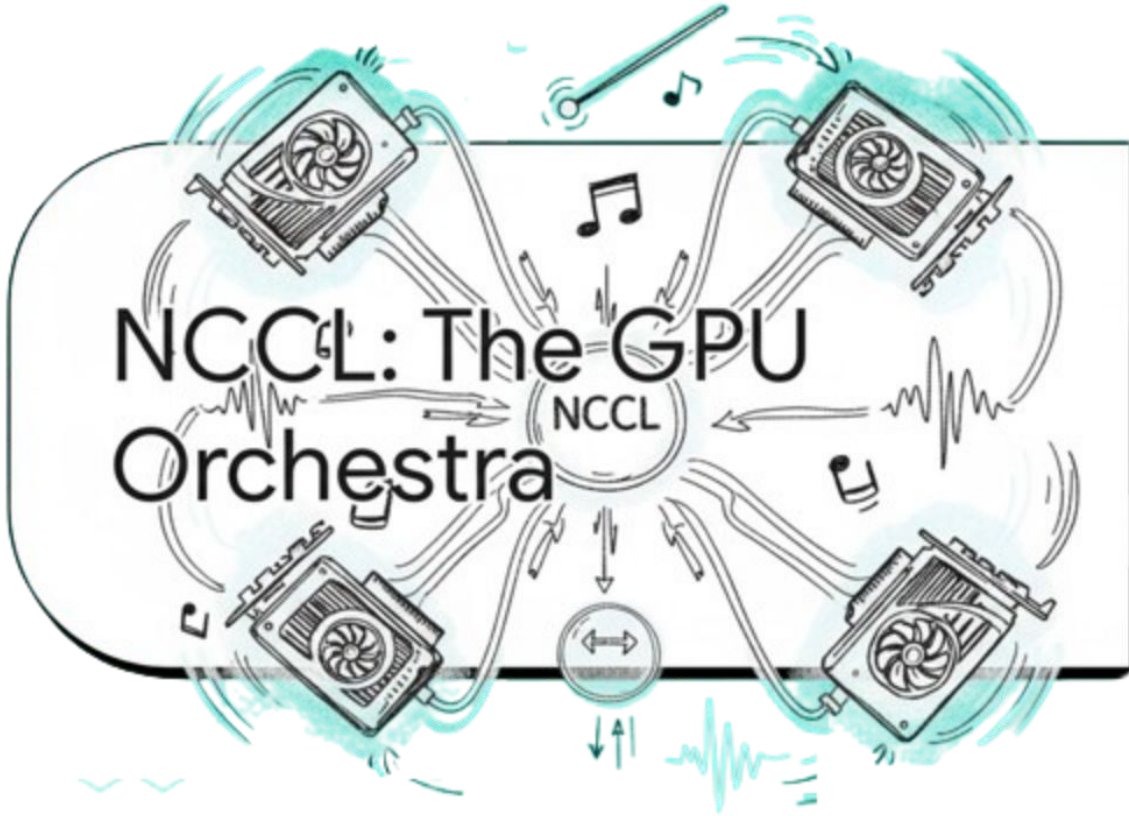
GPU Communication Bottleneck



Iteration time breakdown of the full fine-tuning method on a 1B parameters Transformer decoder model [1]

[1] N. Alnaasan et al., "Characterizing Communication in Distributed Parameter-Efficient Fine-Tuning for Large Language Models," HOTI 2024, pp. 11–19.

NCCL: The GPU Orchestra

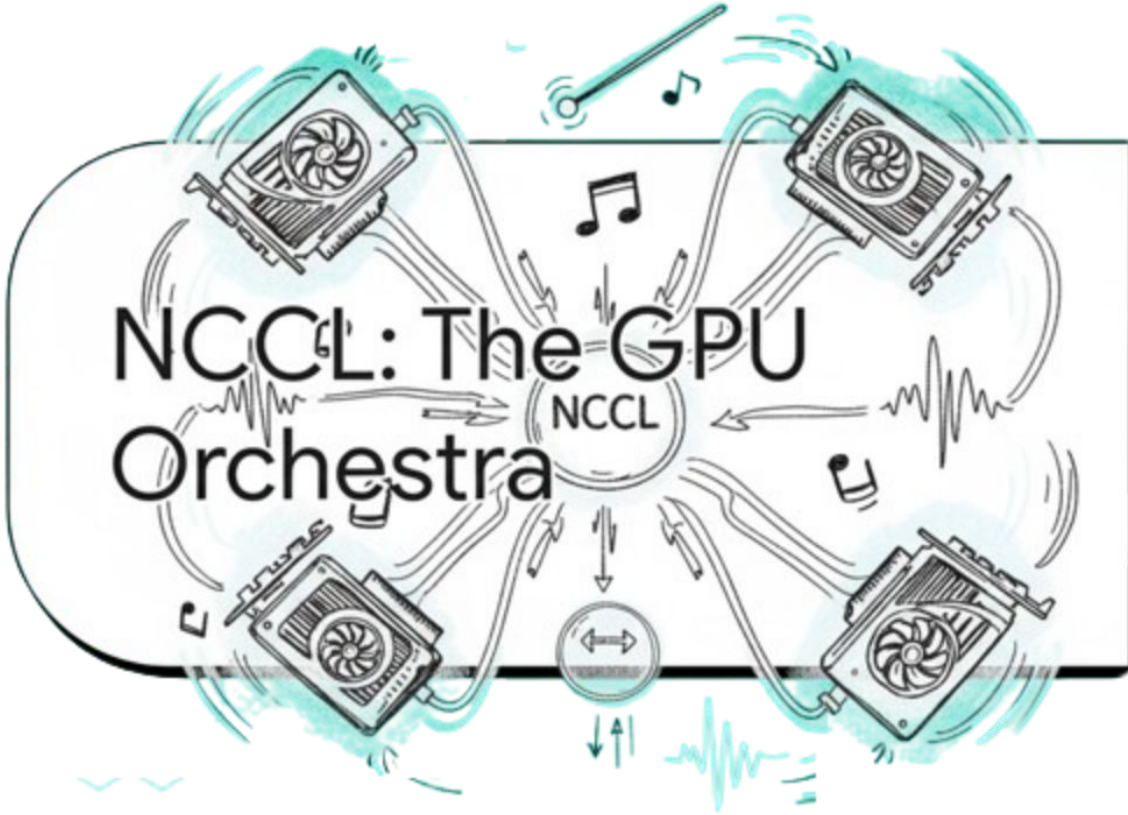


PyTorch

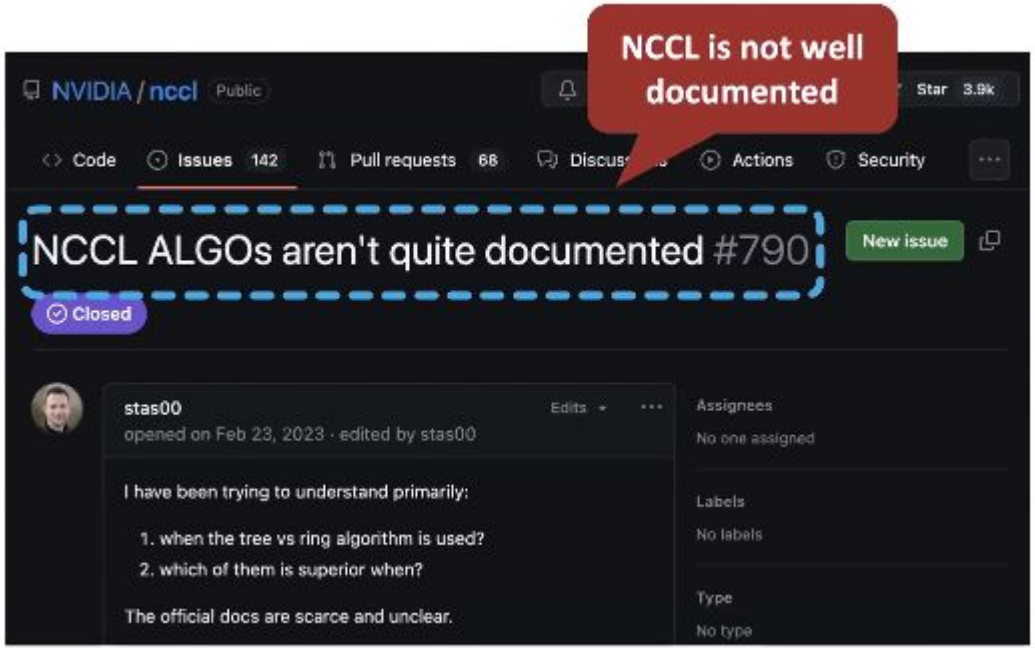
TensorFlow

NCCL

NVIDIA GPU



Motivation


NCCL is not well documented
NVIDIA / nccl
Public
Star 3.9k
Code
Issues 142
Pull requests 68
Actions
Security
NCCL ALGOs aren't quite documented #790
New issue
Closed
stas00
opened on Feb 23, 2023 · edited by stas00
Edits
I have been trying to understand primarily:
1. when the tree vs ring algorithm is used?
2. which of them is superior when?
The official docs are scarce and unclear.
Assignees
No one assigned
Labels
No labels
Type
No type

NCCL Overview - API and Execution Flow

NCCL Overview - API and Execution Flow

Communicator Creation

```
// Create communicator
ncclComm_t comm;
ncclCommInitRank(&comm, nranks, id, rank);
```

NCCL Overview - API and Execution Flow

Communicator Creation

```
// Create communicator
ncclComm_t comm;
ncclCommInitRank(&comm, nranks, id, rank);
```

Collective Communication / P2P Communication

```
// Collective communication call
ncclAllReduce(sendbuff, recvbuff, count, ncclFloat, ncclSum, comm, stream);

// Point-to-point communication call
ncclSend(sendbuff, count, ncclFloat, next_rank, comm, stream);
ncclRecv(recvbuff, count, ncclFloat, prev_rank, comm, stream);
```

NCCL Overview - API and Execution Flow


```
// Create communicator
ncclComm_t comm;
ncclCommInitRank(&comm, nranks, id, rank);

// Start group operation
ncclGroupStart();

// Collective communication call
ncclAllReduce(sendbuff, recvbuff, count, ncclFloat, ncclSum, comm, stream);

// Point-to-point communication call
ncclSend(sendbuff, count, ncclFloat, next_rank, comm, stream);
ncclRecv(recvbuff, count, ncclFloat, prev_rank, comm, stream);

// End group operation
ncclGroupEnd();
```

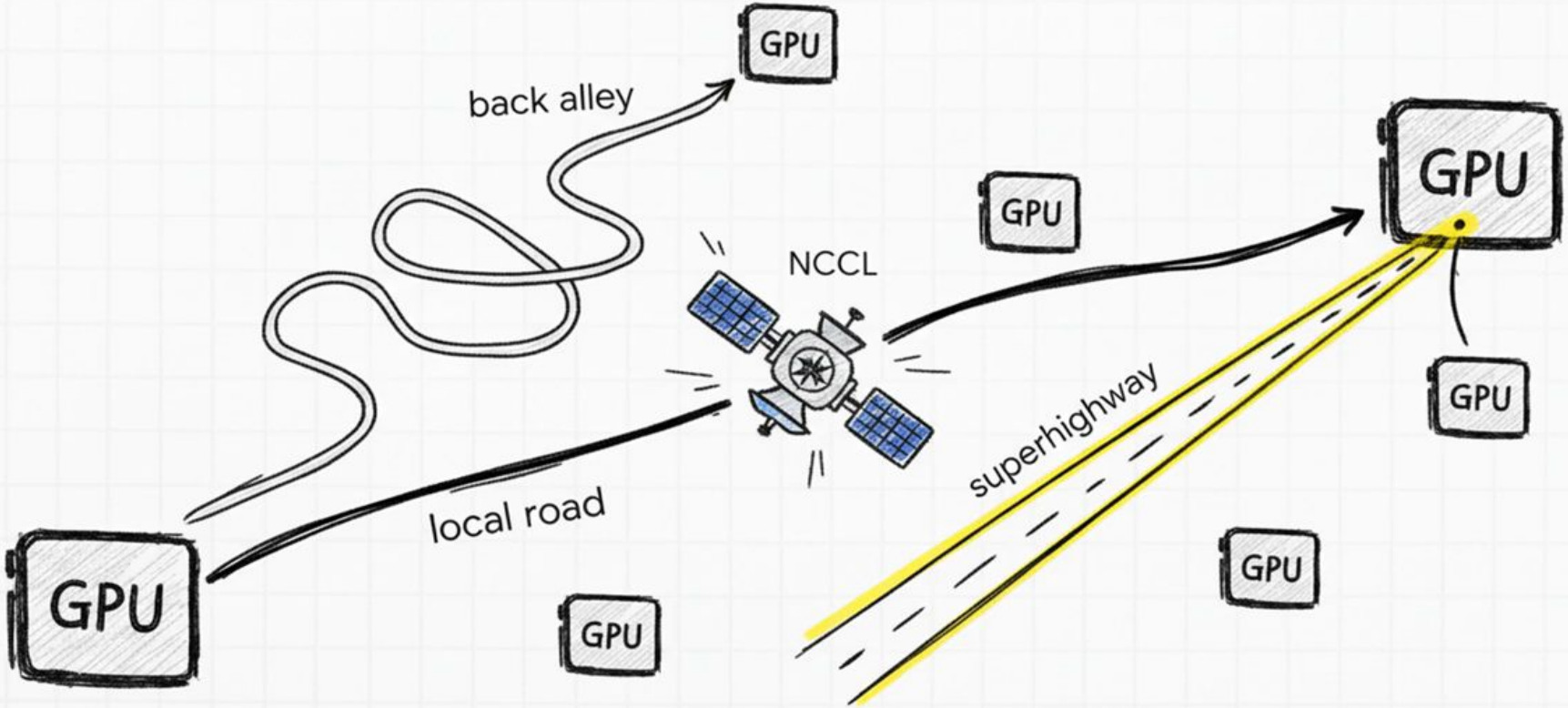
NCCL Overview - API and Execution Flow


```
// Create communicator
ncclComm_t comm;
ncclCommInitRank(&comm, nranks, id, rank);

// Start group operation
ncclGroupStart();

// Collective communication call
ncclAllReduce(sendbuff, recvbuff, count, ncclFloat, ncclSum, comm, stream);

// Point-to-point communication call
ncclSend(sendbuff, count, ncclFloat, next_rank, comm, stream);
ncclRecv(recvbuff, count, ncclFloat, prev_rank, comm, stream);

// End group operation
ncclGroupEnd();

// Destroy communicator
ncclCommDestroy(comm);
```

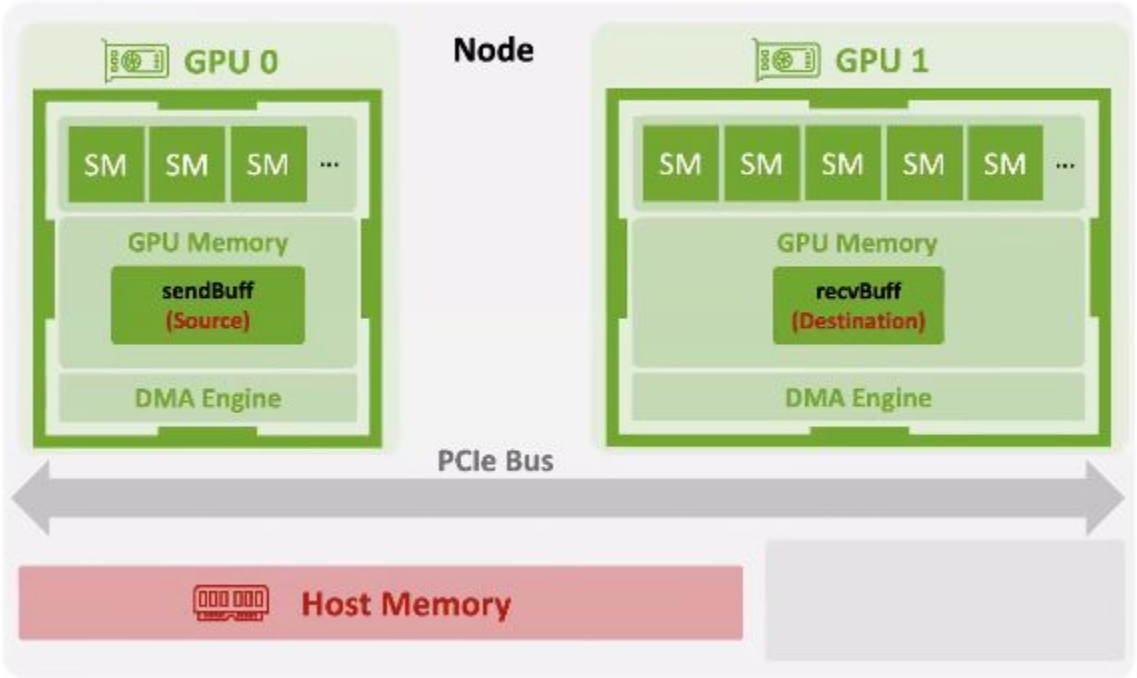
Communication Protocols

Protocol	Design Goal	Payload	Bandwidth Utilization	Latency Per-Hop
Simple	High Bandwidth	Data Chunks	Near Peak	~6μs
Low Latency (LL)	Low Latency	4B data + 4 B flag	25~50% of peak	~1μs
LL128	Best of both	120B data + 8B flag	~95% of peak	~2μs

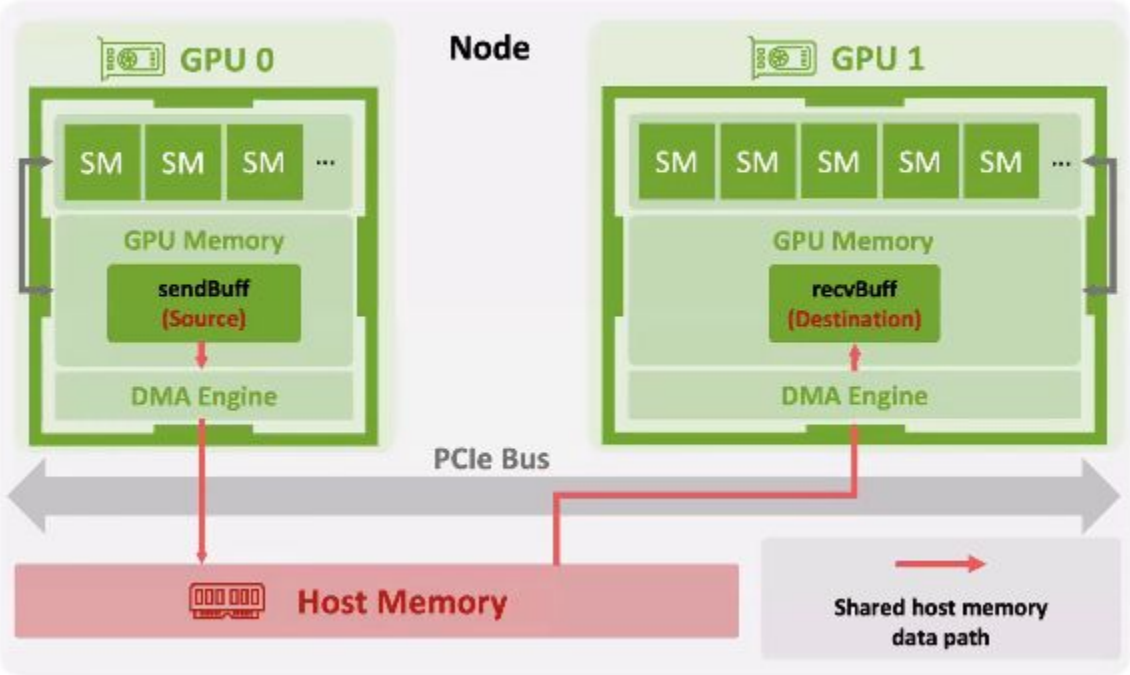
Note: LL128 requires NVLink


GPU
back alley
GPU
GPU
NCCL
GPU
superhighway
GPU
local road
GPU
GPU
GPU

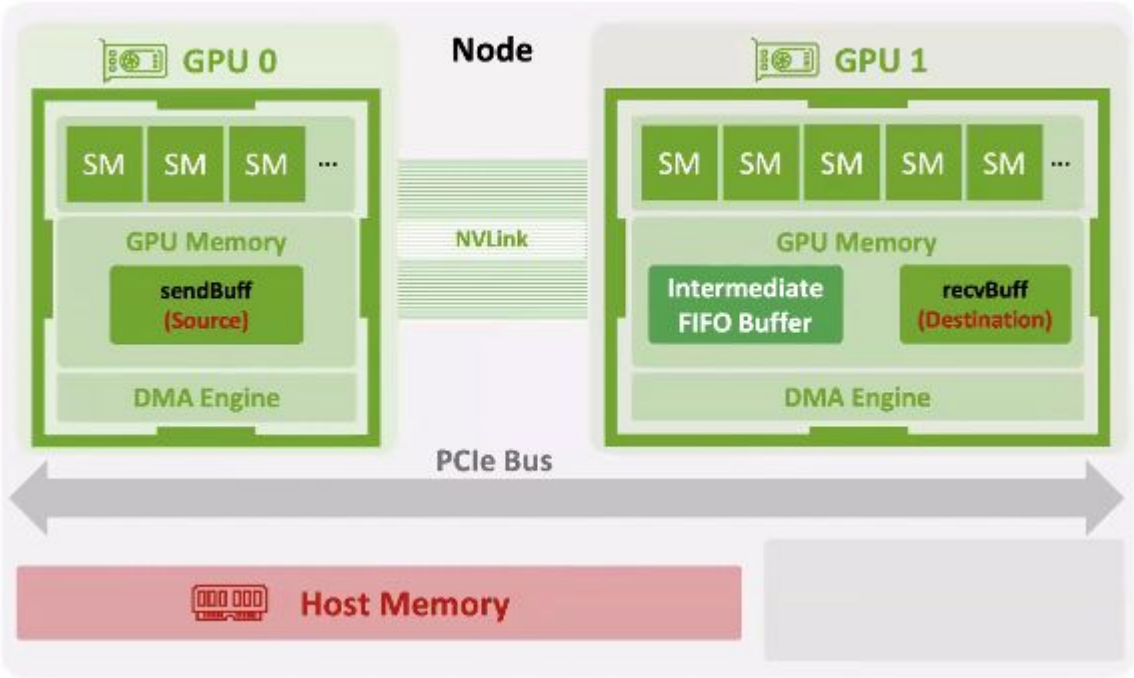
Intra-Node Communication



Intra-Node Communication

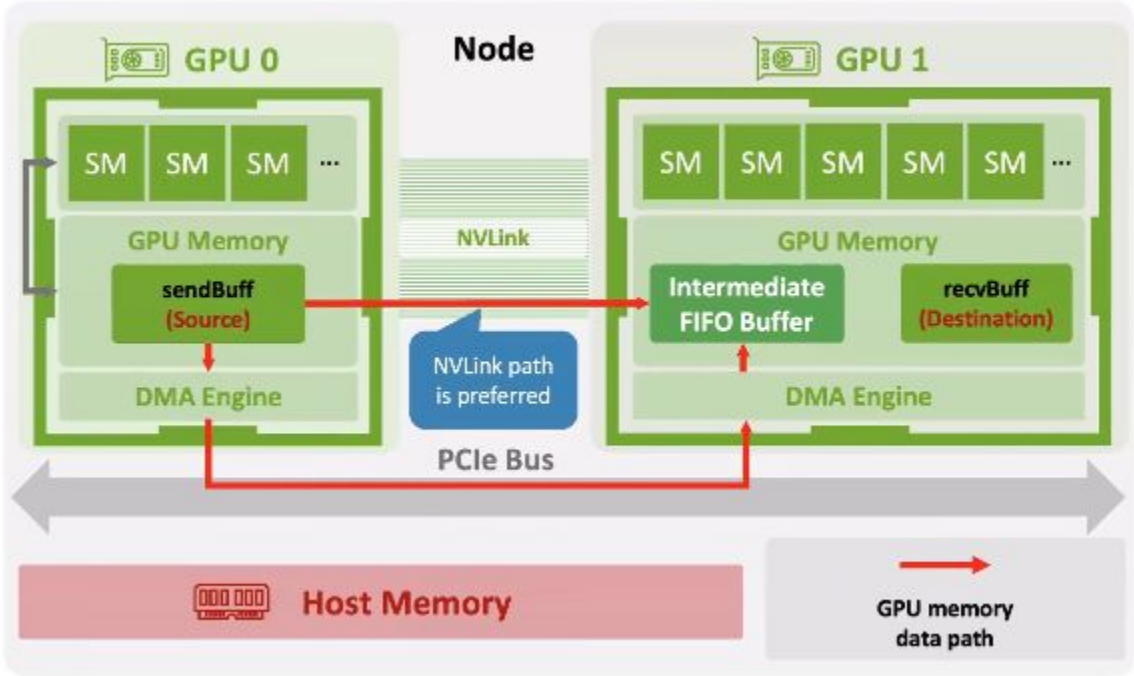


Intra-Node Communication



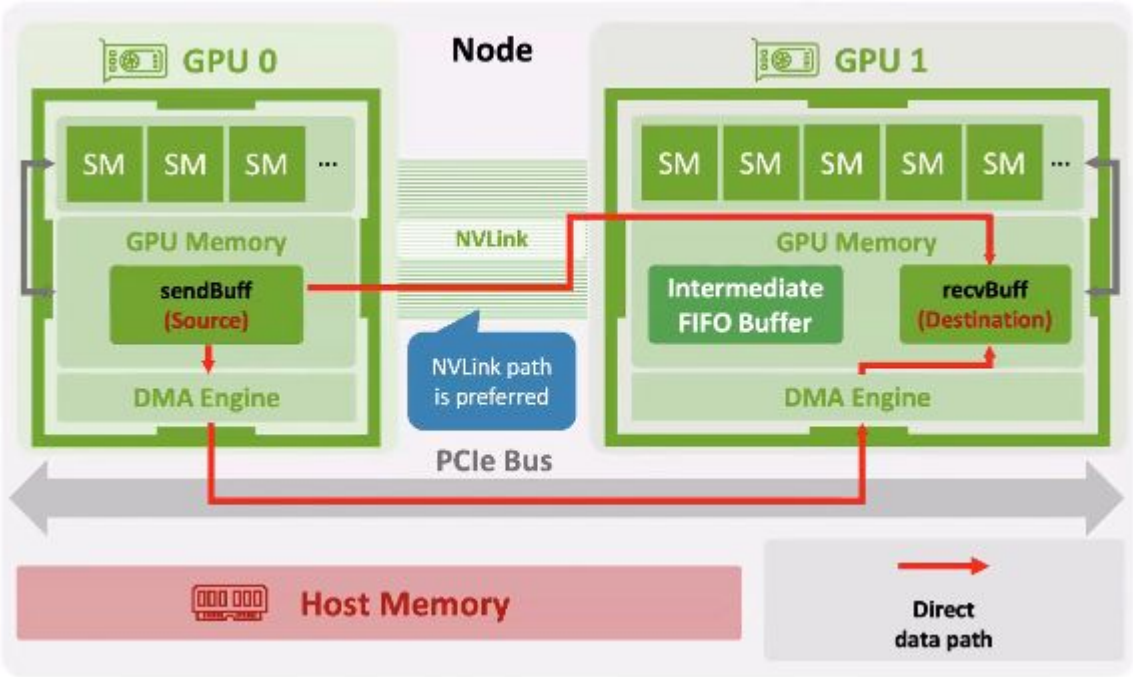
GPU Memory

Intra-Node Communication



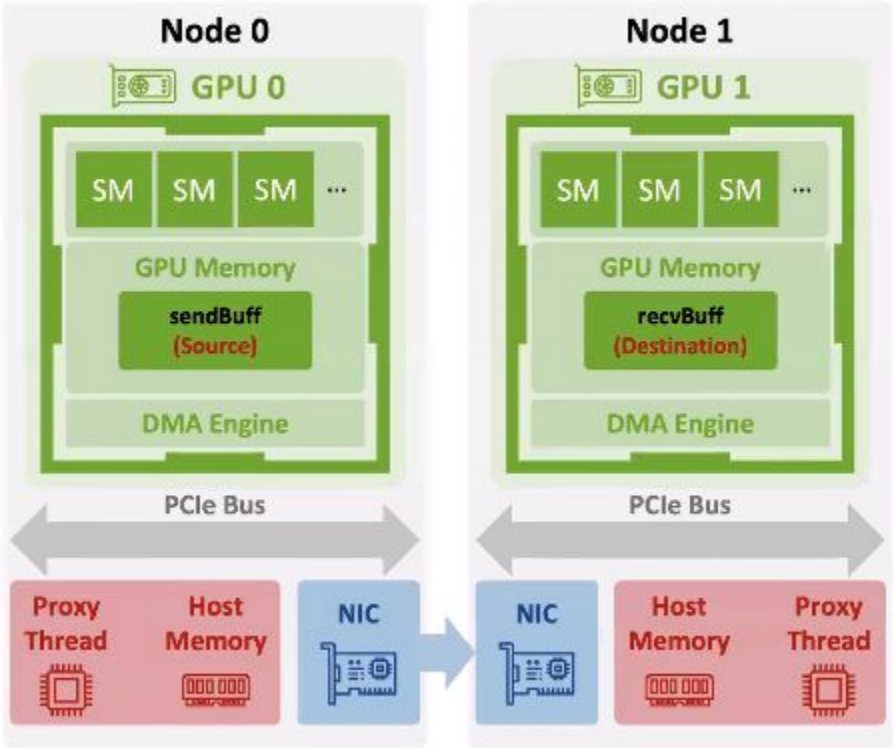
GPU Memory

Intra-Node Communication

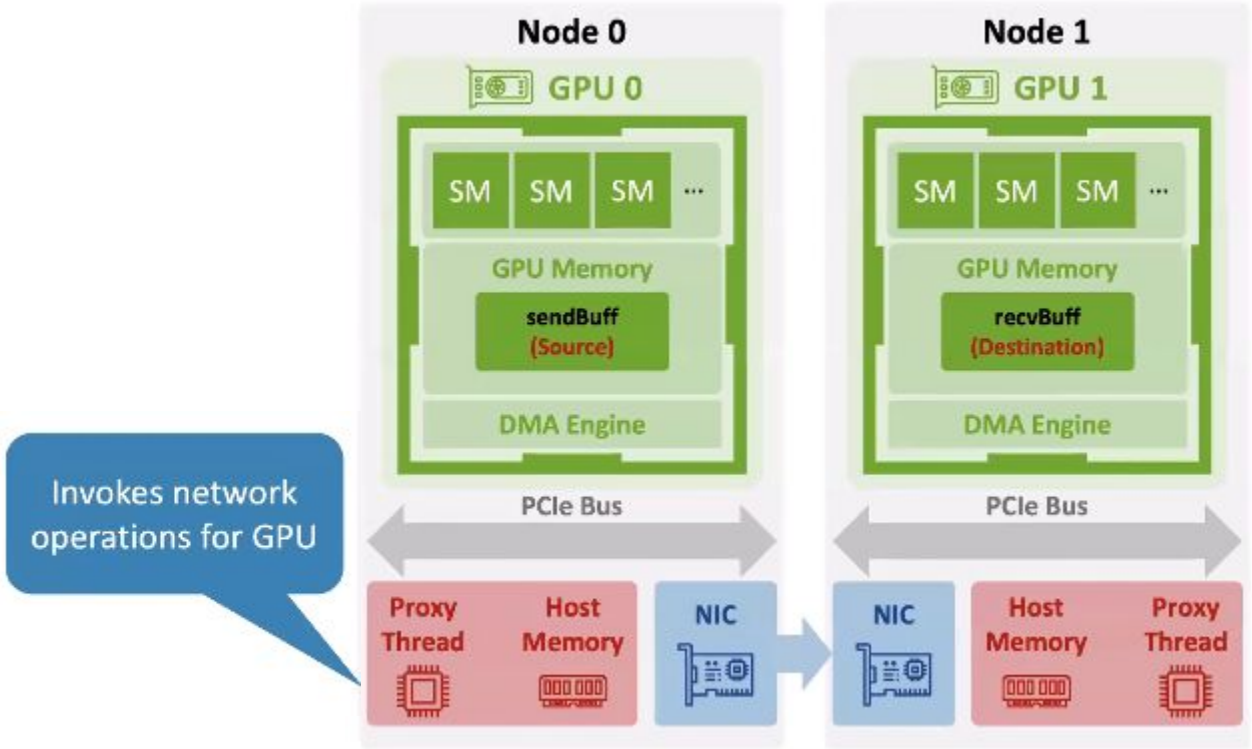


P2P_DIRECT mode enabled and GPUs belong to the same process

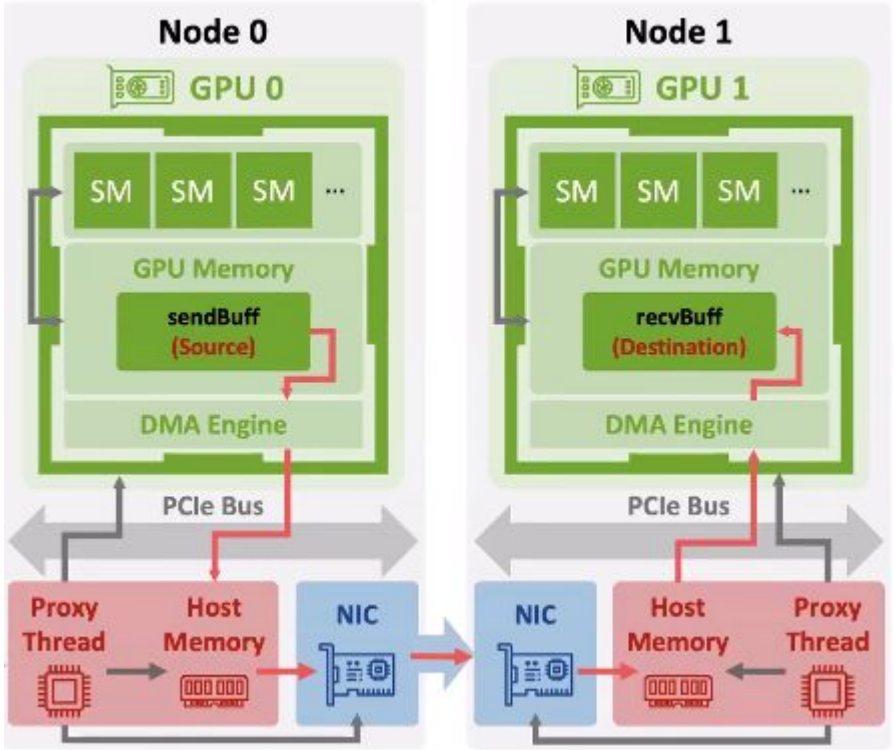
Inter-Node Communication



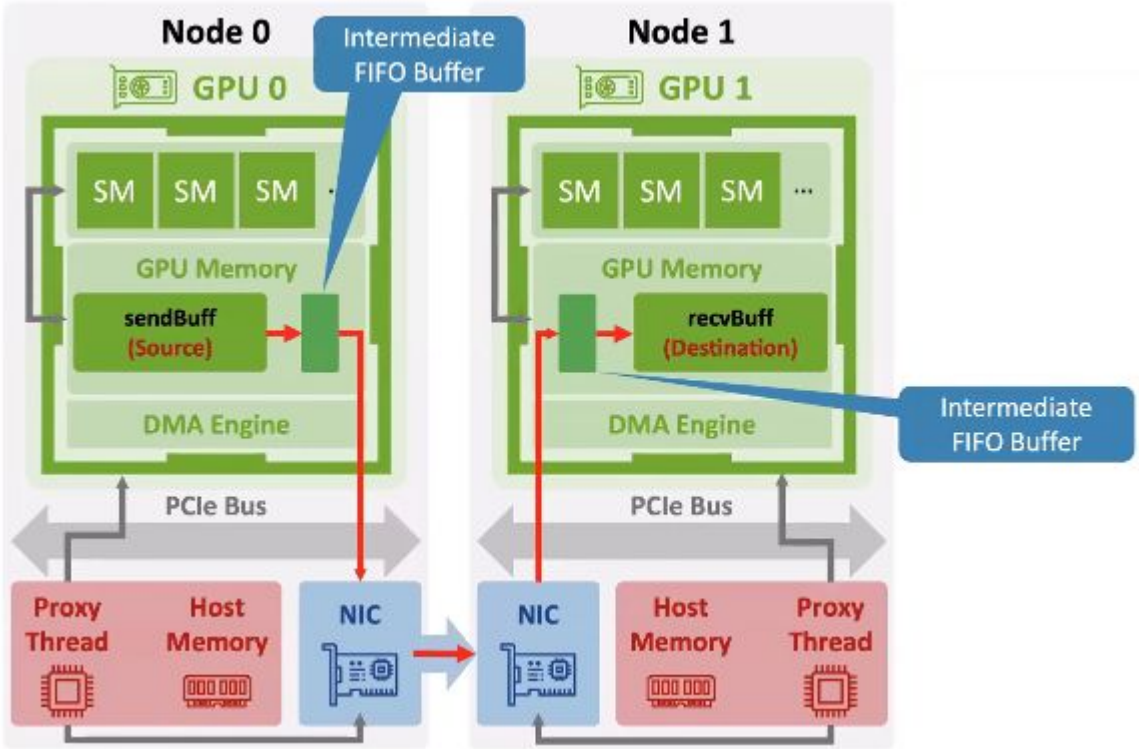
Inter-Node Communication



Inter-Node Communication

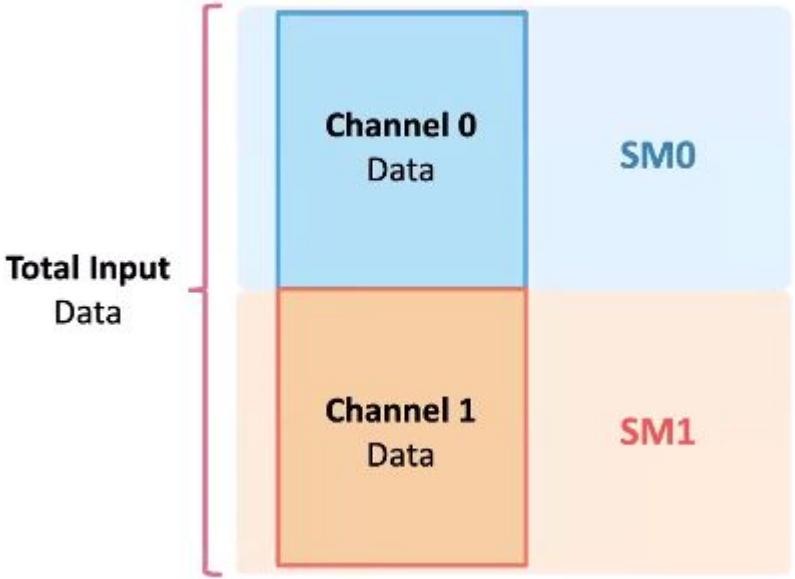


Inter-Node Communication

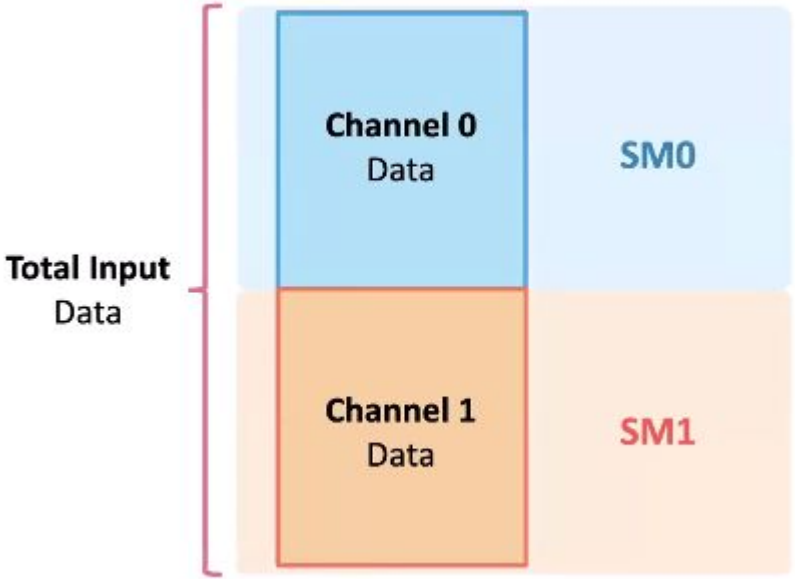


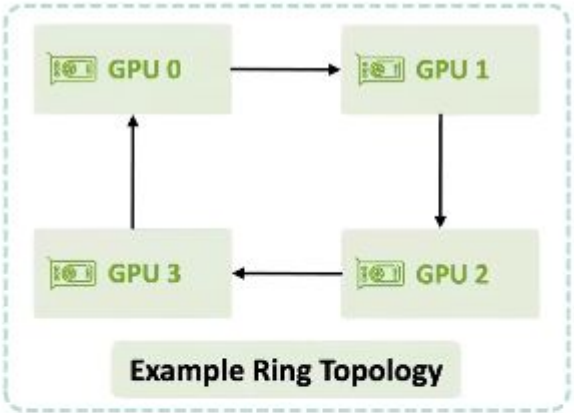
With GPU Direct RDMA

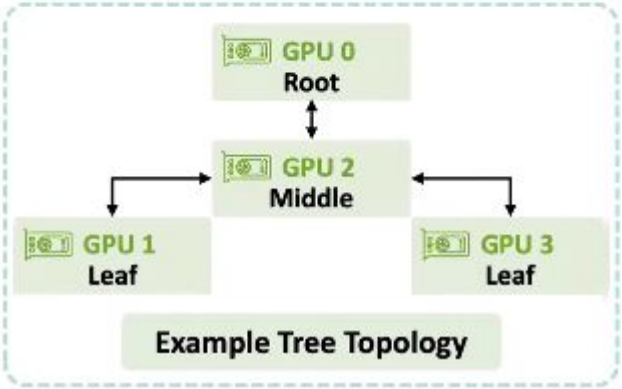
Communication Channels & Logical Topology



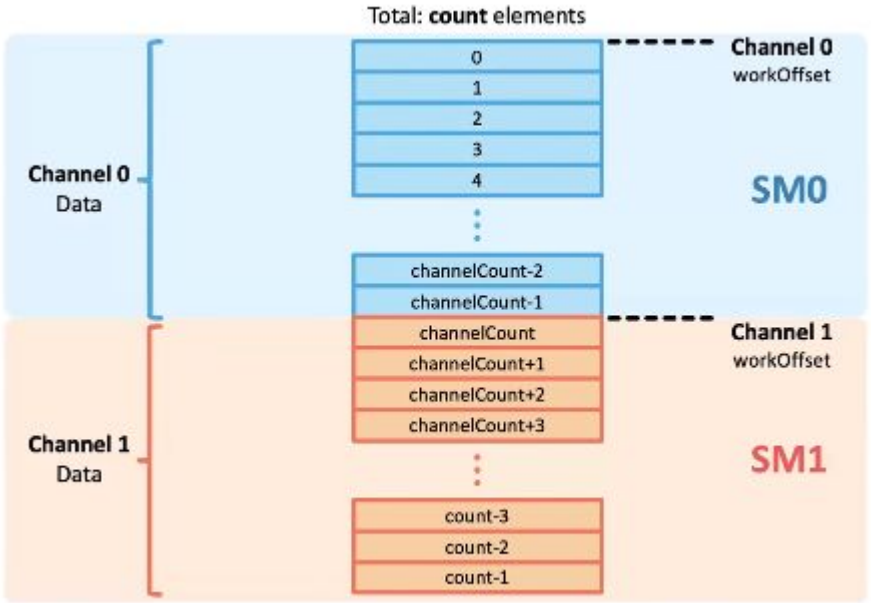
Communication Channels & Logical Topology



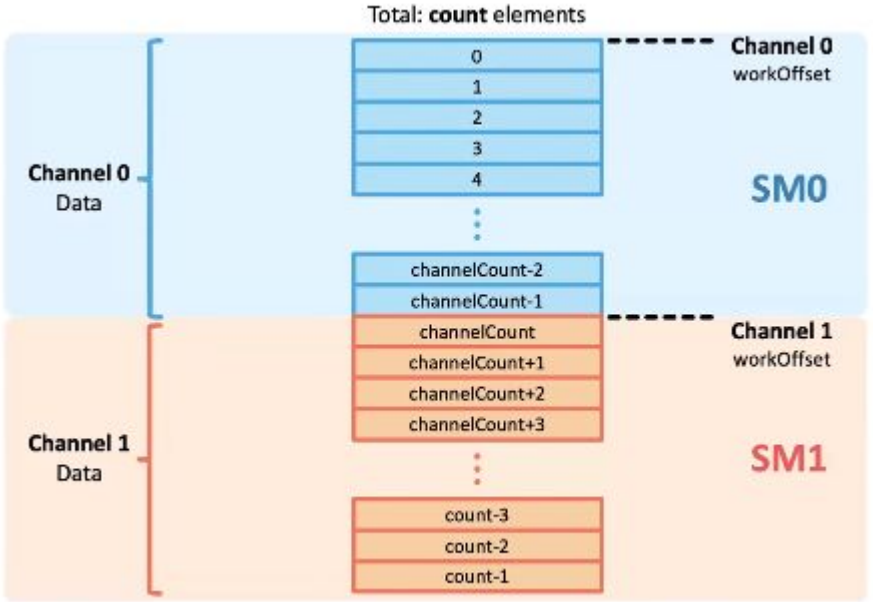




Iterative Execution and Communication Primitives



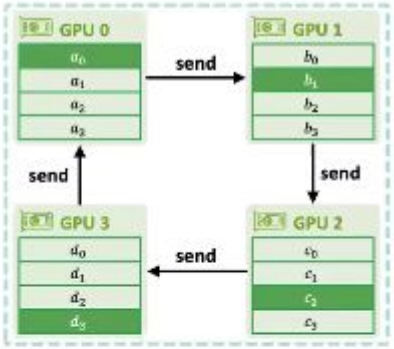
Iterative Execution and Communication Primitives

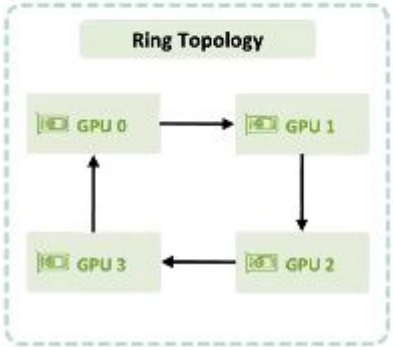


Step Index	NCCL Primitive
0	`send`
1 to $k-2$	`recvReduceSend`
$k-1$	`recvReduceCopySend`
k to $2k-3$	`recvCopySend`
$2k-2$	`recv`

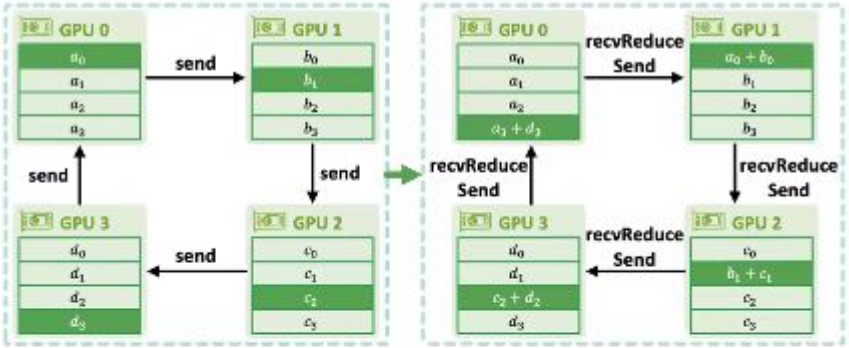
Ring AllReduce Example

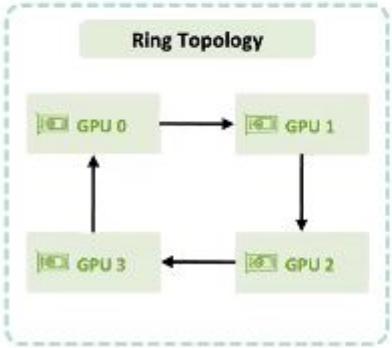
Step Index	NCCL Primitive
0	`send`
1 to $k-2$	`recvReduceSend`
$k-1$	`recvReduceCopySend`
k to $2k-3$	`recvCopySend`
$2k-2$	`recv`





Ring AllReduce Example

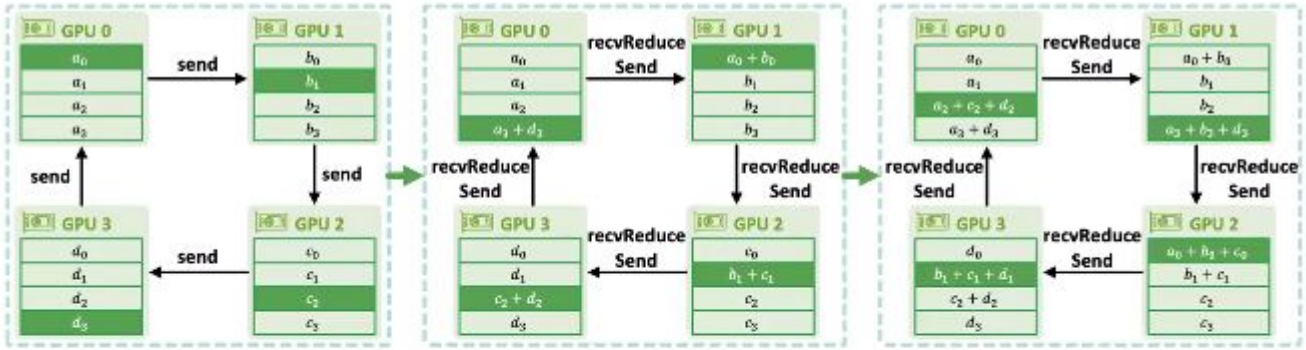


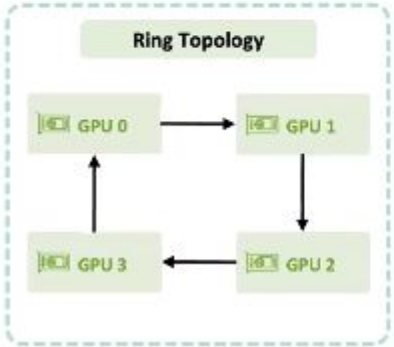


Step Index	NCCL Primitive
0	`send`
1 to $k-2$	`recvReduceSend`
$k-1$	`recvReduceCopySend`
k to $2k-3$	`recvCopySend`
$2k-2$	`recv`

Ring AllReduce Example

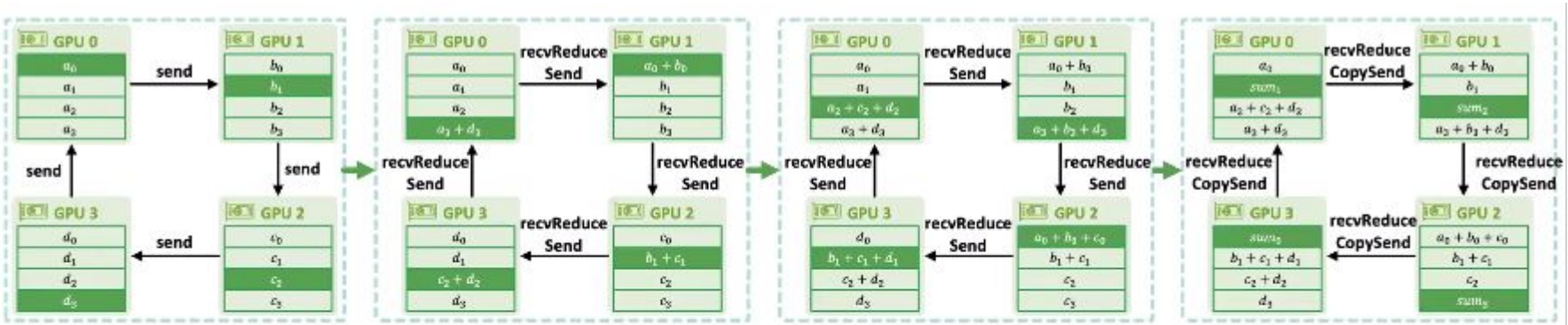
Step Index	NCCL Primitive
0	`send`
1 to $k-2$	`recvReduceSend`
$k-1$	`recvReduceCopySend`
k to $2k-3$	`recvCopySend`
$2k-2$	`recv`

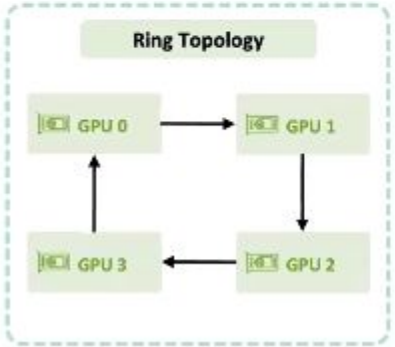




Ring AllReduce Example

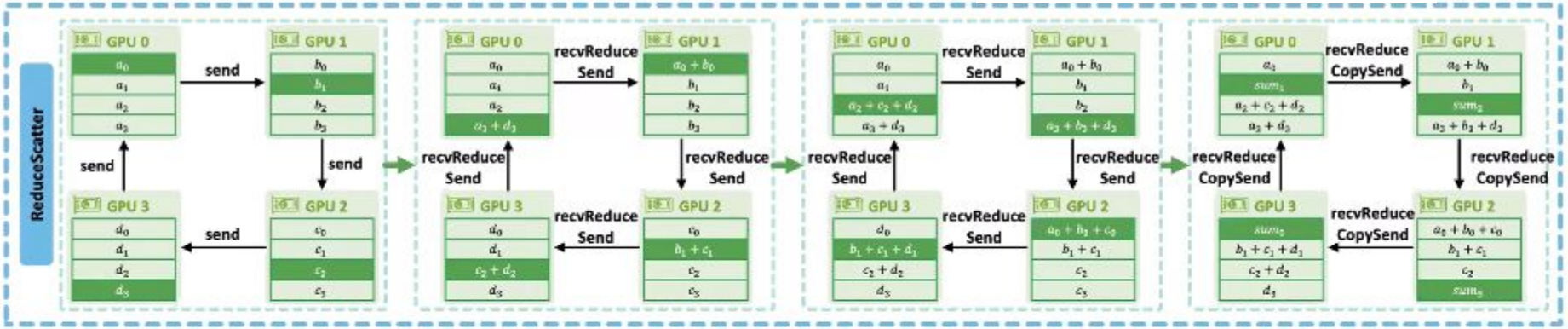
Step Index	NCCL Primitive
0	`send`
1 to $k-2$	`recvReduceSend`
$k-1$	`recvReduceCopySend`
k to $2k-3$	`recvCopySend`
$2k-2$	`recv`

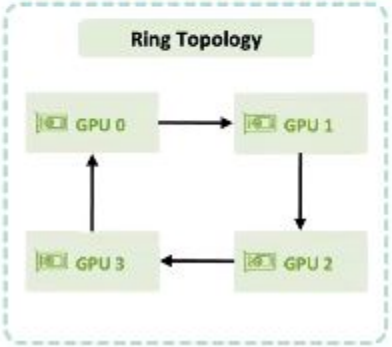




Ring AllReduce Example

Step Index	NCCL Primitive
0	`send`
1 to $k-2$	`recvReduceSend`
$k-1$	`recvReduceCopySend`
k to $2k-3$	`recvCopySend`
$2k-2$	`recv`



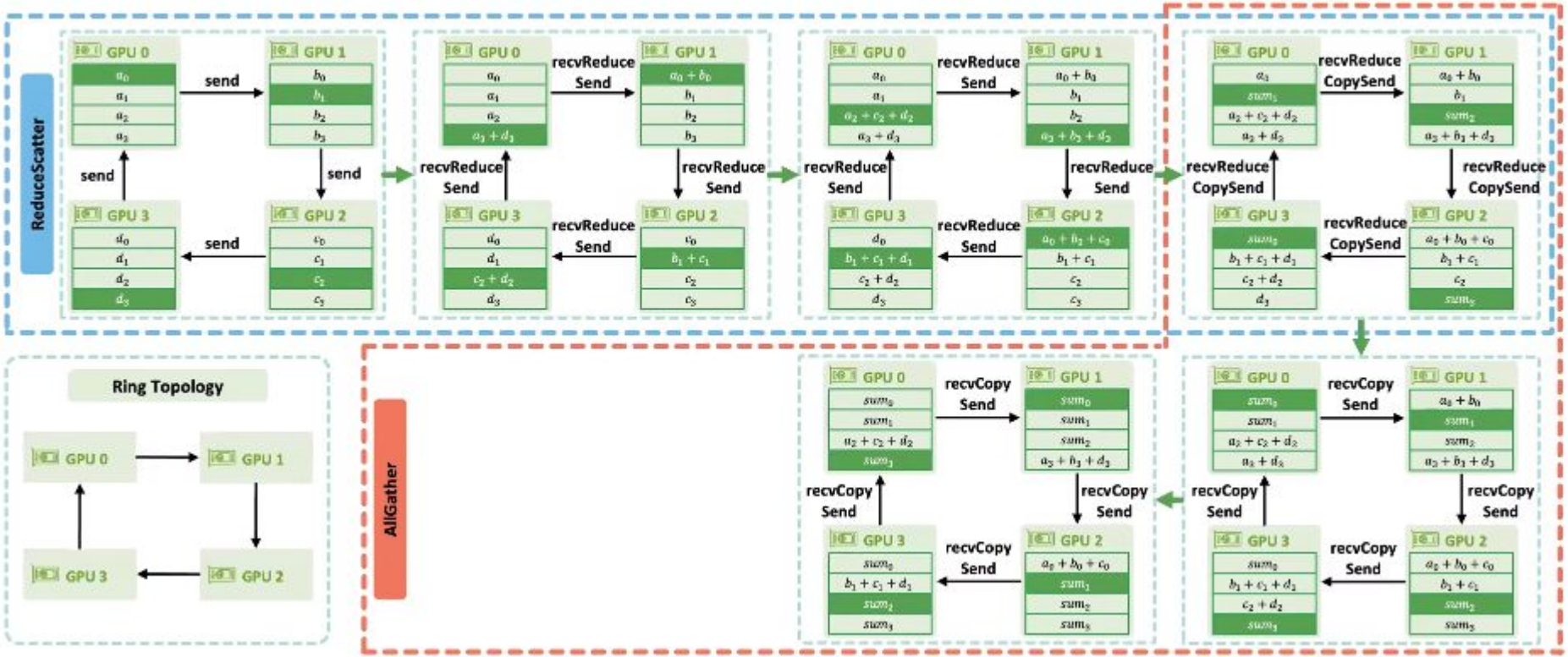


Ring AllGather Example

Step Index	NCCL Primitive
0	`send`
1 to $k-2$	`recvReduceSend`
$k-1$	`recvReduceCopySend`
k to $2k-3$	`recvCopySend`
$2k-2$	`recv`

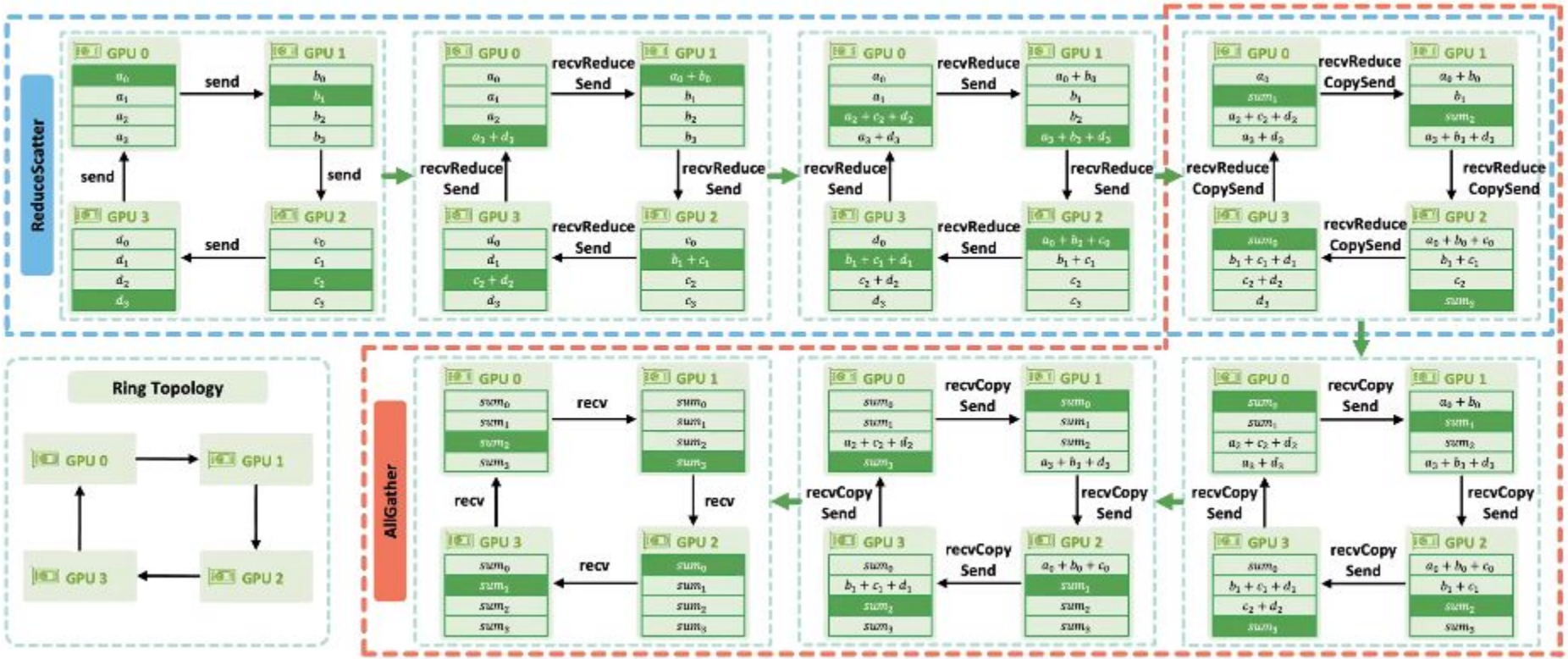
Ring AllGather Example

Step Index	NCCL Primitive
0	`send`
1 to $k-2$	`recvReduceSend`
$k-1$	`recvReduceCopySend`
k to $2k-3$	`recvCopySend`
$2k-2$	`recv`

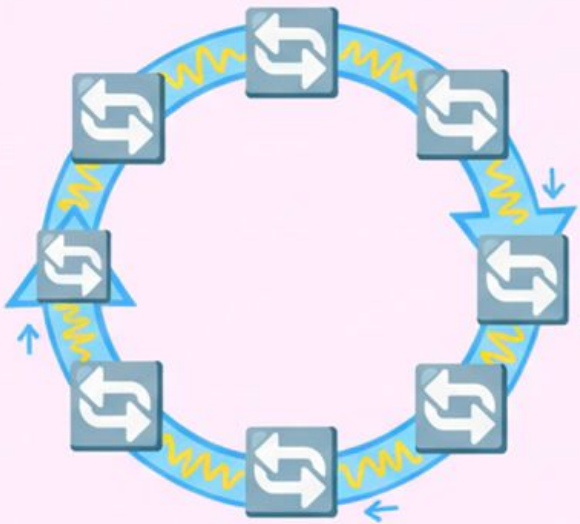


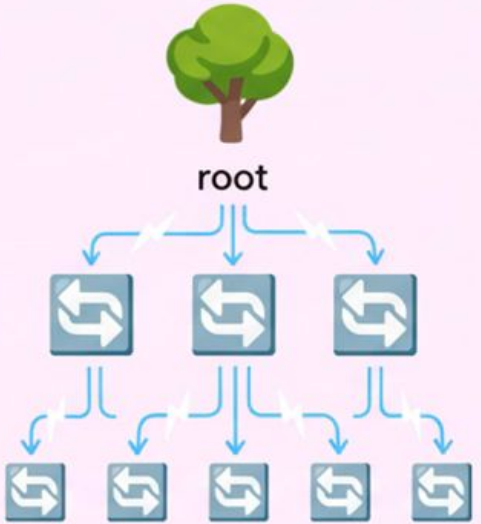
Ring AllGather Example

Step Index	NCCL Primitive
0	`send`
1 to $k-2$	`recvReduceSend`
$k-1$	`recvReduceCopySend`
k to $2k-3$	`recvCopySend`
$2k-2$	`recv`



Choosing Between the Topologies

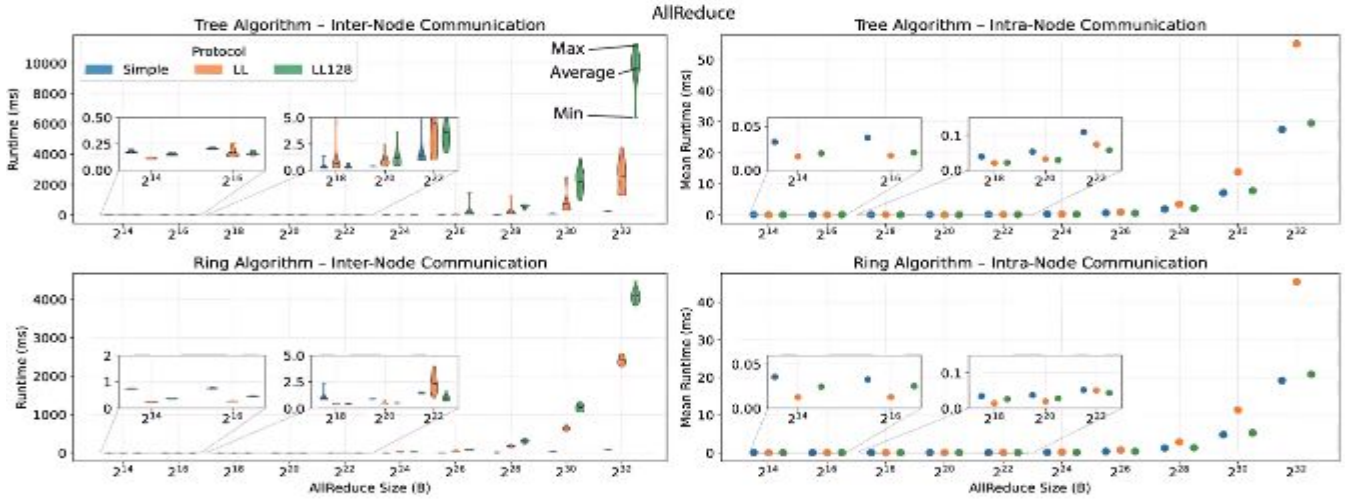




Benchmarking Results

Alps Supercomputer (CSCS)

- Grace Hopper Superchips (GH200)
- 150 GB/s intra-node communication
- 25 GB/s Cray Slingshot
- Dragonfly topology



Benchmarking results for all NCCL collectives in the paper

Impact and Outlook

ATLAHS: An Application-centric Network Simulator Toolchain for AI, HPC, and Distributed Storage

Siyuan Shen*
siyuan.shen@inf.ethz.ch
ETH Zürich
Zürich, Switzerland

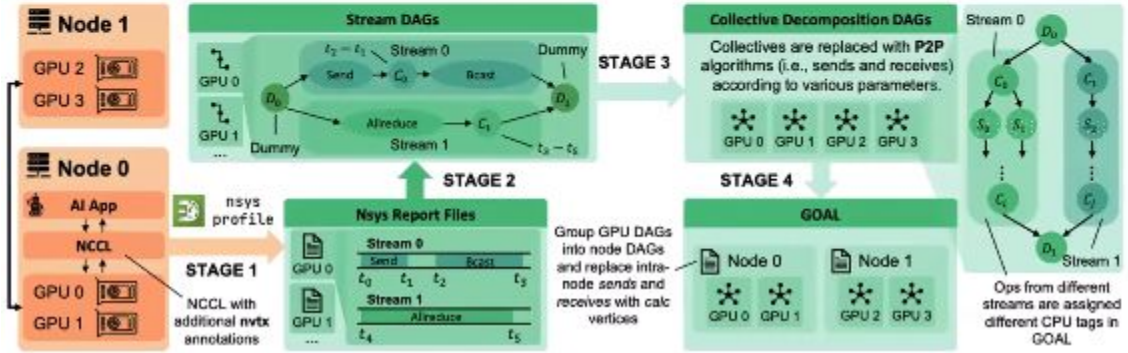
Tommaso Bonato*
tommaso.bonato@inf.ethz.ch
ETH Zürich
Zürich, Switzerland

Zhiyi Hu
zhiyihu@student.ethz.ch
ETH Zürich
Zürich, Switzerland

Pasquale Jordan
pasquale.jordan@inf.ethz.ch
ETH Zürich
Zürich, Switzerland

Tiancheng Chen
tiancheng.chen@inf.ethz.ch
ETH Zürich
Zürich, Switzerland

Torsten Hoefler
torsten.hoefler@inf.ethz.ch
ETH Zürich
Zürich, Switzerland



Questions for Discussion

- How does NCCL manage communication channels, and what is the trade-off in channel count selection?
- NCCL uses different topologies (ring vs. tree) for different operations. How might emerging network architectures (like disaggregated systems, optical interconnects, or SmartNICs) fundamentally change optimal collective communication strategies?
- NCCL's "autotuning" approach generally provides good performance. How should we think about the balance between automatic optimization and manual control in HPC/ML systems?
- The benchmarking shows significant performance differences between intra-node and inter-node communication. What does this tell us about the future of large-scale AI training systems?
- How does the derived tool, ATLAHS, address the gap of NCCL being a black box in a way that simply accessing the source code cannot?