ForestColl: Throughput-Optimal Collective Communications on Heterogeneous Network Fabrics

Liangyu Zhao[1] Saeed Maleki Yuanhong Wang[3] Zezhou Wang[2] Ziyue Yang[2]
Hossein Pourreza[2] Arvind Krishnamurthy[1]

[1]*University of Washington* [2]*Microsoft Research* [3]*Tsinghua University*

23rd USENIX Symposium on Networked Systems Design and Implementation

Table of Contents

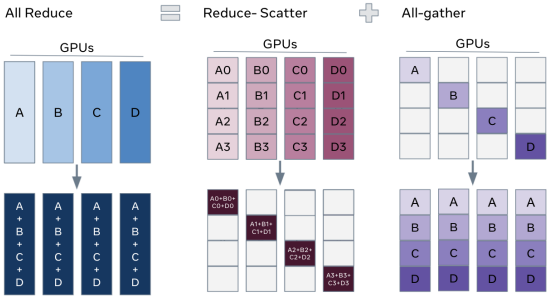
Collective Communication

Collective Communication: Coordinated communication patterns that synchronize or exchange data between multiple nodes, widely used in distributed ML training and serving.

Collective Communication

Collective Communication: Coordinated communication patterns that synchronize or exchange data between multiple nodes, widely used in distributed ML training and serving.

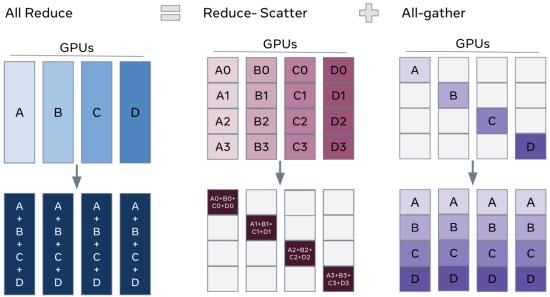
- Patterns with well-defined input and output states.



Collective Communication

Collective Communication: Coordinated communication patterns that synchronize or exchange data between multiple nodes, widely used in distributed ML training and serving.

- Patterns with well-defined input and output states.
- An efficient allgather algorithm can be transformed into other collective operations.



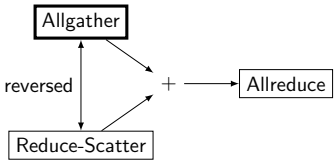


Figure: Relations between collective operations.

Challenges

We aim to generate high-throughput communication schedules for any given network topology.

Challenges

We aim to generate high-throughput communication schedules for any given network topology.

- **Diversity & Heterogeneity:** Today's ML network topologies are highly *diverse* across hardware platforms and *heterogeneous* within individual networks.

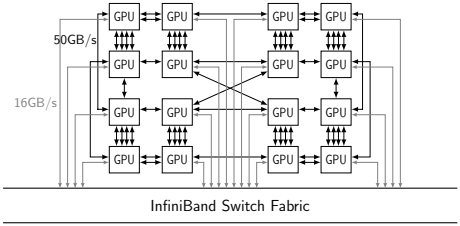


Figure: AMD MI250 Topology

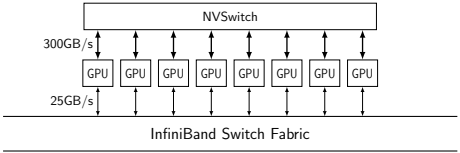


Figure: NVIDIA DGX A100 Topology

Challenges

We aim to generate high-throughput communication schedules for any given network topology.

- **Diversity & Heterogeneity:** Today's ML network topologies are highly *diverse* across hardware platforms and *heterogeneous* within individual networks.
- **Scalability:** Optimizing aggregation and multicast traffic requires strict data dependency, often resulting in NP-hard discrete optimization.

# of nodes	4	9	16	25	36
SCCL [PPoPP '21]	0.61s	1.00s	60s	3286s	$>10^4$s
TACCL [NSDI '23]	0.45s	67.8s	1801s	1802s	n/a

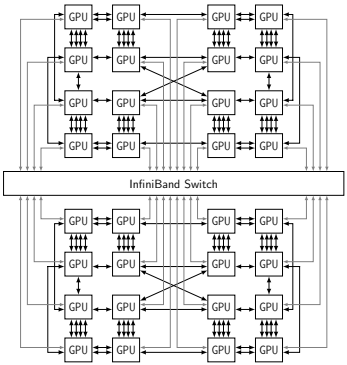
Table: Generation Time on 2D Torus ($n \times n$)

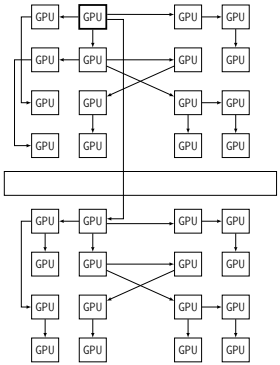
Table of Contents

ForestColl

ForestColl: construct spanning trees (forest☺) with k trees rooted at each node/GPU.

- In allgather, every tree *simultaneously* broadcasts $1/k$ of its root's data shard.

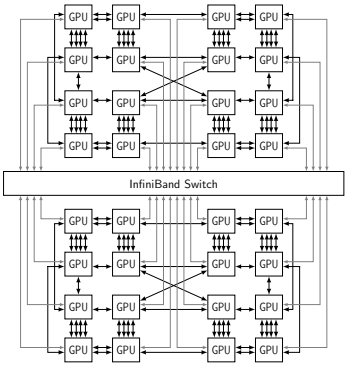


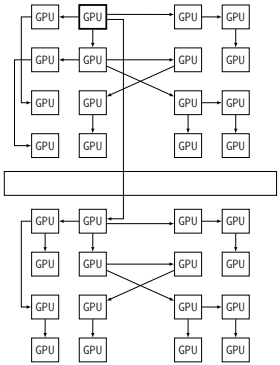


ForestColl

ForestColl: construct spanning trees (forest☺) with k trees rooted at each node/GPU.

- In allgather, every tree *simultaneously* broadcasts $1/k$ of its root's data shard.
- **Performance:** The trees achieve mathematically **maximum throughput** during broadcast.

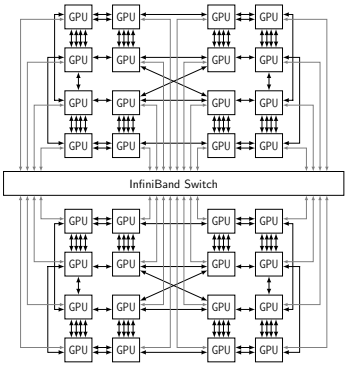


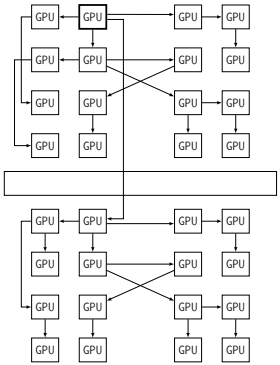


ForestColl

ForestColl: construct spanning trees (forest☺) with k trees rooted at each node/GPU.

- In allgather, every tree *simultaneously* broadcasts $1/k$ of its root's data shard.
- **Performance:** The trees achieve mathematically **maximum throughput** during broadcast.
- **Scalability:** Schedule generation is in **polynomial time**.





Previous Work

	SCCL [PPoPP '21]	TACCL [NSDI '23]	BFB [NSDI '25]	Blink [MLSys '20]	TE-CCL [SIGCOMM '24]	SyCCL [SIGCOMM '25]	**ForestColl** [NSDI '26]
Switch Network							
Optimal Schedule							
Scalable Runtime							

Previous schedule generation methods either

Previous Work

	SCCL [PPoPP '21]	TACCL [NSDI '23]	BFB [NSDI '25]	Blink [MLSys '20]	TE-CCL [SIGCOMM '24]	SyCCL [SIGCOMM '25]	**ForestColl** [NSDI '26]
Switch Network	×	✓	×	×	✓	✓	✓
Optimal Schedule							
Scalable Runtime							

Previous schedule generation methods either

- focus on switch-free direct-connect networks only;

Previous Work

	SCCL [PPoPP '21]	TACCL [NSDI '23]	BFB [NSDI '25]	Blink [MLSys '20]	TE-CCL [SIGCOMM '24]	SyCCL [SIGCOMM '25]	**ForestColl** [NSDI '26]
Switch Network	×	✓	×	×	✓	✓	✓
Optimal Schedule	✓	×	×	×	×	×	✓
Scalable Runtime							

Previous schedule generation methods either

- focus on switch-free direct-connect networks only;
- lack theoretical performance guarantees for generated schedules;

Previous Work

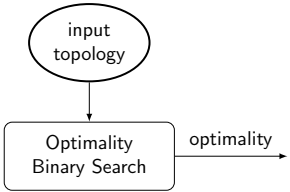
	SCCL [PPoPP '21]	TACCL [NSDI '23]	BFB [NSDI '25]	Blink [MLSys '20]	TE-CCL [SIGCOMM '24]	SyCCL [SIGCOMM '25]	**ForestColl** [NSDI '26]
Switch Network	×	✓	×	×	✓	✓	✓
Optimal Schedule	✓	×	×	×	×	×	✓
Scalable Runtime	×	×	✓	✓	×	×	✓

Previous schedule generation methods either

- focus on switch-free direct-connect networks only;
- lack theoretical performance guarantees for generated schedules;
- rely on NP-hard optimization methods.

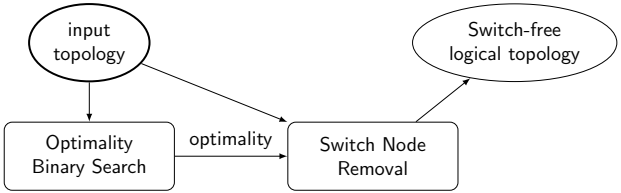
Algorithm Overview

1. **Optimality Binary Search:** Compute the optimal throughput of a given network topology.



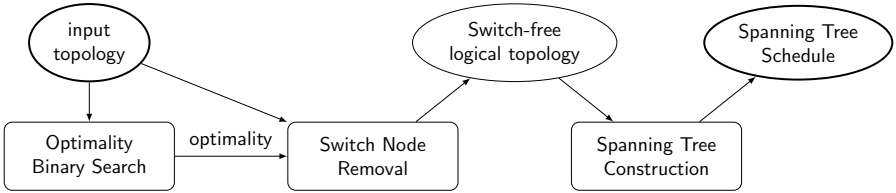
Algorithm Overview

1. **Optimality Binary Search:** Compute the optimal throughput of a given network topology.
2. **Switch Node Removal:** Derive a switch-free logical topology.



Algorithm Overview

1. **Optimality Binary Search:** Compute the optimal throughput of a given network topology.
2. **Switch Node Removal:** Derive a switch-free logical topology.
3. **Spanning Tree Construction:** Construct spanning trees on the switch-free topology as a schedule.



ForestColl Optimality

Q: What is the optimal allgather throughput given a network topology?

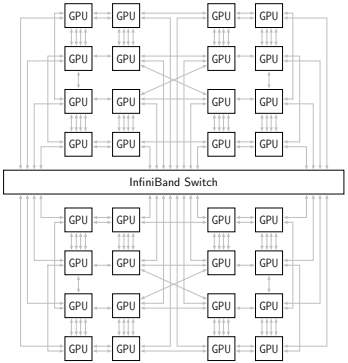


Figure: 2-Box AMD MI250

ForestColl Optimality

Q: What is the optimal allgather throughput given a network topology?

- Previous works often look at **single-node lower bound**:

$$\frac{\text{data recv by a node}}{\text{node bandwidth}} = \frac{\text{shard size} \times (\text{num of GPUs} - 1)}{\text{node bandwidth}} = \frac{M}{B} \cdot \frac{N-1}{N}$$

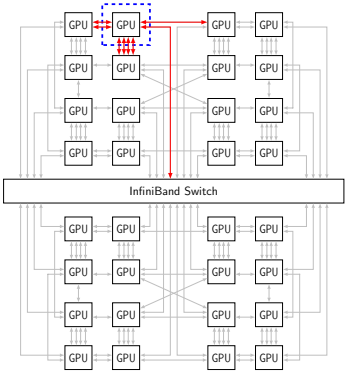


Figure: 2-Box AMD MI250

ForestColl Optimality

Q: What is the optimal allgather throughput given a network topology?

- Previous works often look at **single-node lower bound**:

$$\frac{\text{data recv by a node}}{\text{node bandwidth}} = \frac{\text{shard size} \times (\text{num of GPUs} - 1)}{\text{node bandwidth}} = \frac{M}{B} \cdot \frac{N-1}{N}$$

- **Problem:** What if the throughput is not bottlenecked by the bandwidth of a single node?

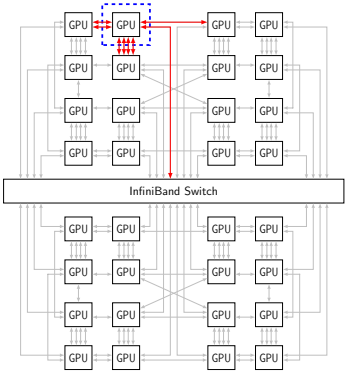


Figure: 2-Box AMD MI250

ForestColl Optimality

Q: What is the optimal allgather throughput given a network topology?

- Consider an arbitrary network cut S.

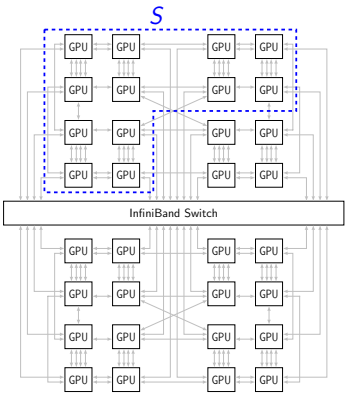


Figure: 2-Box AMD MI250

ForestColl Optimality

Q: What is the optimal allgather throughput given a network topology?

- Consider an arbitrary network cut S.
- Cut S implies an allgather time lower bound:

$$\frac{\text{min data exiting } S}{\text{available bandwidth}} = \frac{\text{shard size} \times \text{num of GPUs in } S}{\text{exiting bandwidth of } S}$$

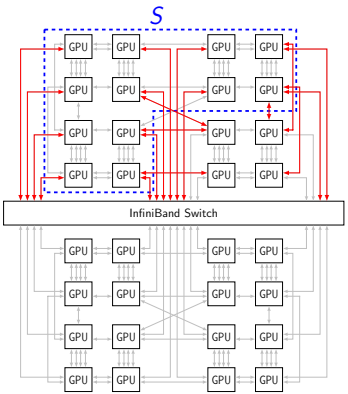


Figure: 2-Box AMD MI250

ForestColl Optimality

Q: What is the optimal allgather throughput given a network topology?

- Consider an arbitrary network cut S.
- Cut S implies an allgather time lower bound:

$$\frac{\text{min data exiting } S}{\text{available bandwidth}} = \frac{\text{shard size} \times \text{num of GPUs in } S}{\text{exiting bandwidth of } S}$$

- The optimal allgather throughput is determined by a **bottleneck cut** S^*, where

$$\text{shard size} \times \frac{\text{num of GPUs in } S^*}{\text{exiting bandwidth of } S^*}$$

is maximized across all possible network cuts.

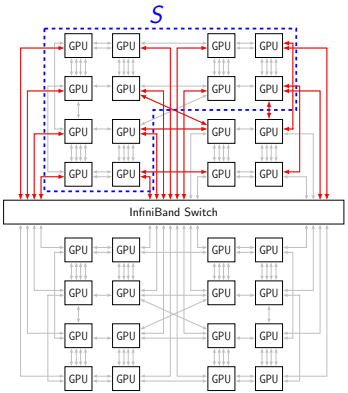


Figure: 2-Box AMD MI250

ForestColl Optimality

Q: What is the optimal allgather throughput given a network topology?

- Consider an arbitrary network cut S.
- Cut S implies an allgather time lower bound:

$$\frac{\text{min data exiting } S}{\text{available bandwidth}} = \frac{\text{shard size} \times \text{num of GPUs in } S}{\text{exiting bandwidth of } S}$$

- The optimal allgather throughput is determined by a **bottleneck cut** S^*, where

$$\text{shard size} \times \frac{\text{num of GPUs in } S^*}{\text{exiting bandwidth of } S^*} = \boxed{\frac{M}{N} \max_{S \subset V, S \supseteq V_c} \frac{|S \cap V_c|}{B^+(S)}}$$

is maximized across all possible network cuts.

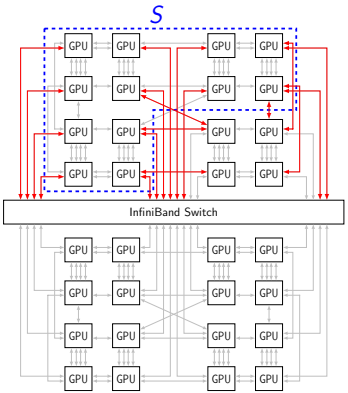


Figure: 2-Box AMD MI250

ForestColl Optimality

Q: What is the optimal allgather throughput given a network topology?

- Consider an arbitrary network cut S.
- Cut S implies an allgather time lower bound:

$$\frac{\text{min data exiting } S}{\text{available bandwidth}} = \frac{\text{shard size} \times \text{num of GPUs in } S}{\text{exiting bandwidth of } S}$$

- The optimal allgather throughput is determined by a **bottleneck cut** S^*, where

$$\text{shard size} \times \frac{\text{num of GPUs in } S^*}{\text{exiting bandwidth of } S^*} = \boxed{\frac{M}{N} \max_{S \subset V, S \not\supseteq V_c} \frac{|S \cap V_c|}{B^+(S)}}$$

is maximized across all possible network cuts.

Different from min cut, which only looks at cut bandwidth!

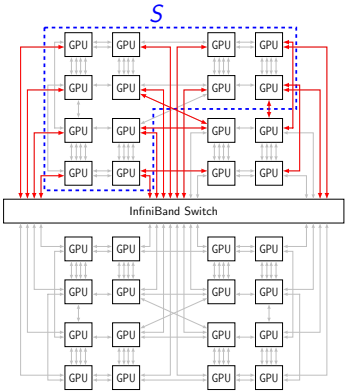


Figure: 2-Box AMD MI250

ForestColl Optimality

Q: What is the optimal allgather throughput given a network topology?

- Consider an arbitrary network cut S.
- Cut S implies an allgather time lower bound:

$$\frac{\text{min data exiting } S}{\text{available bandwidth}} = \frac{\text{shard size} \times \text{num of GPUs in } S}{\text{exiting bandwidth of } S}$$

- The optimal allgather throughput is determined by a **bottleneck cut** S^*, where

$$\text{shard size} \times \frac{\text{num of GPUs in } S^*}{\text{exiting bandwidth of } S^*} = \boxed{\frac{M}{N} \max_{S \subset V, S \not\supseteq V_c} \frac{|S \cap V_c|}{B^+(S)}}$$

is maximized across all possible network cuts.

1. ForestColl can efficiently compute the above optimality.
2. ForestColl's spanning trees achieve the optimality.

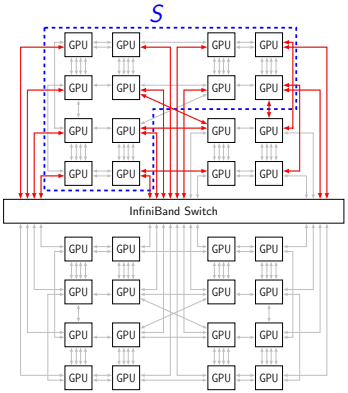
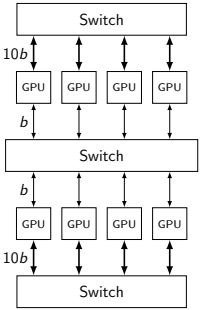


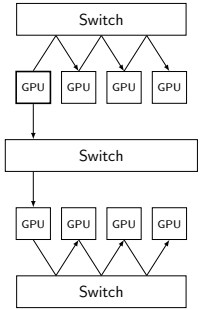
Figure: 2-Box AMD MI250

Spanning Tree Packing

Spanning Tree Packing: Construct as many trees as possible under link capacity constraints.

- **Intuition:** Each tree is one broadcast flow. Maximize num of trees = maximize throughput.

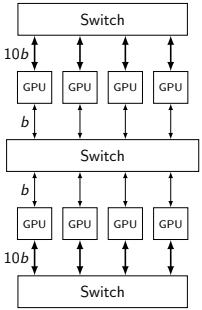


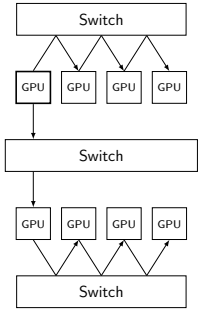


Spanning Tree Packing

Spanning Tree Packing: Construct as many trees as possible under link capacity constraints.

- **Intuition:** Each tree is one broadcast flow. Maximize num of trees = maximize throughput.
- **Problem:** Allgather broadcast is not a spanning tree in a switch topology.

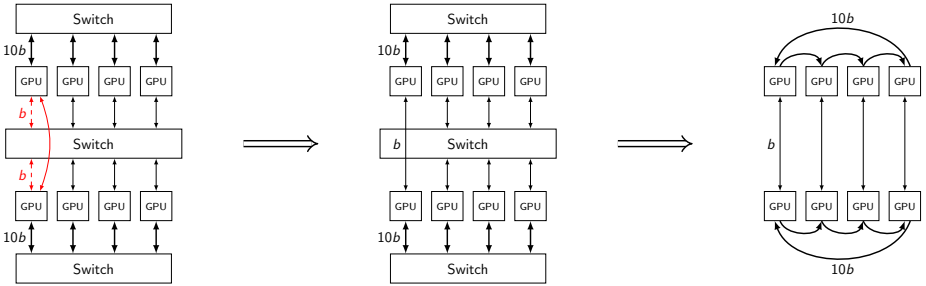




Switch Node Removal

Switch Node Removal: Convert the input topology into a switch-free logical topology.

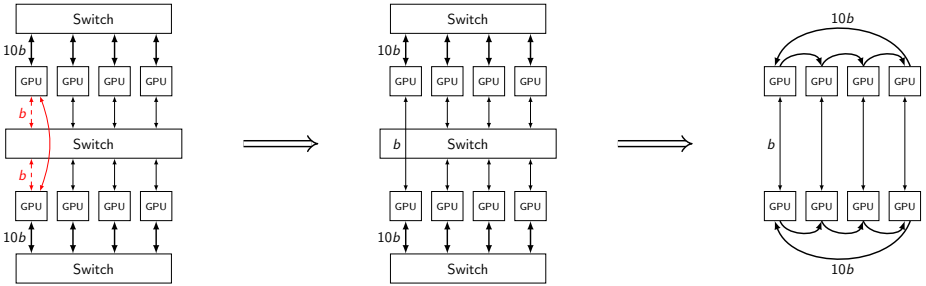
- **Edge Splitting:** Iteratively replace switch links with direct links bypassing the switch.



Switch Node Removal

Switch Node Removal: Convert the input topology into a switch-free logical topology.

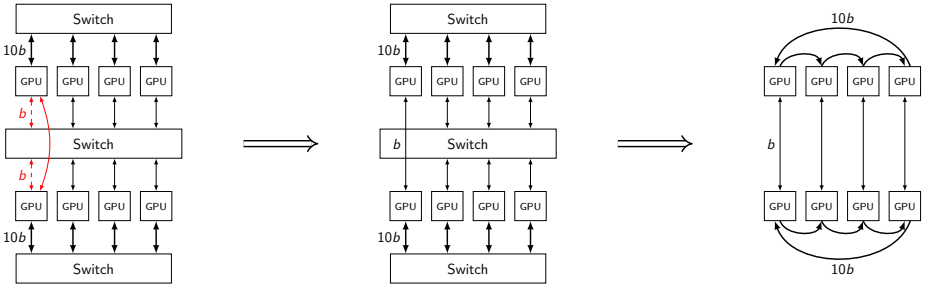
- **Edge Splitting:** Iteratively replace switch links with direct links bypassing the switch.
- **Equivalence:** schedule for logical topology $\Longrightarrow$ schedule for input topology with same throughput.



Switch Node Removal

Switch Node Removal: Convert the input topology into a switch-free logical topology.

- **Edge Splitting:** Iteratively replace switch links with direct links bypassing the switch.
- **Equivalence:** schedule for logical topology $\Longrightarrow$ schedule for input topology with same throughput.
- **Zero Optimality Loss:** optimality of logical topology = optimality of input topology.



Switch Node Removal

Switch Node Removal: Convert the input topology into a switch-free logical topology.

- **Edge Splitting:** Iteratively replace switch links with direct links bypassing the switch.
- **Equivalence:** schedule for logical topology $\Longrightarrow$ schedule for input topology with same throughput.
- **Zero Optimality Loss:** optimality of logical topology = optimality of input topology.
  - Guarantee no decrease in bottleneck cut bandwidth during edge splitting.

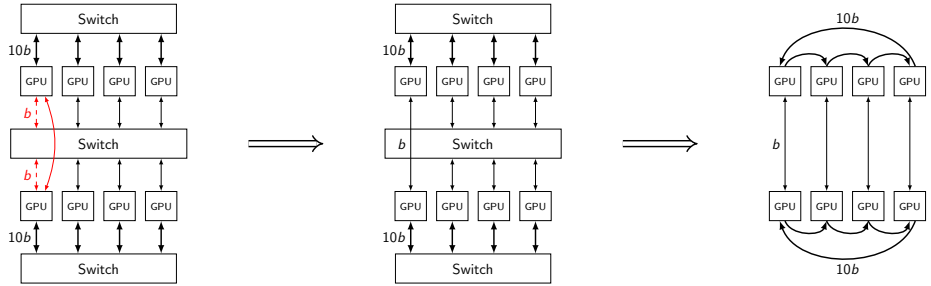


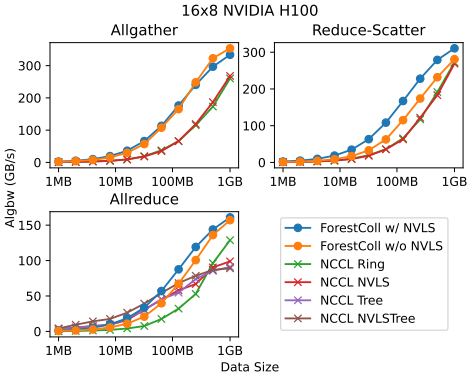
Table of Contents

Experiments

Implement ForestColl's schedule with MSCCL++:

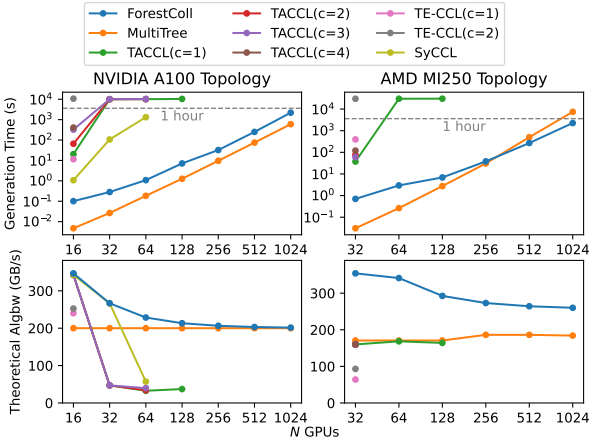
- Cross-box: parallel rings.
- Intra-box: ring / NVLink SHARP.
- Overlap cross- and intra-box communication.

16x8 H100: ForestColl delivers 32%, 14%, and 25% higher throughput than NCCL at 1GB for allgather, reduce-scatter, and allreduce, respectively.



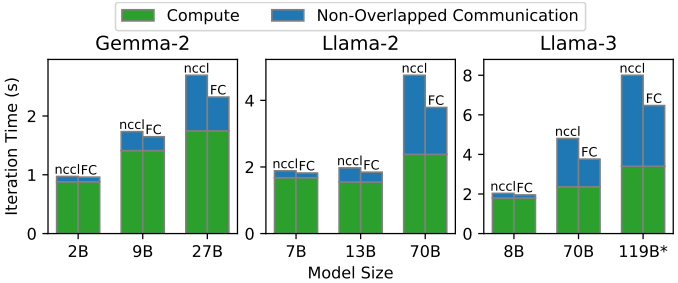
Schedule Generation

- **Generation Speed:** order-of-magnitude faster than NP-hard solutions.
- **Schedule Quality:** ForestColl is always throughput-optimal.



ML Training Evaluation

FSDP Training: ForestColl reduces training iteration times by 14% for Gemma 27B and 20% for Llama 70B and 119B* compared to NCCL.



Llama-3-119B* is our reduced version of Llama-3-405B, with fewer hidden layers.

Summary

ForestColl is a schedule generation algorithm for collective communications that

- provides **provably optimal** schedule;
- works on **any network topology** (direct-connect or switch topology);
- runs in **polynomial time** (scalable to large number of nodes);
- outperforms state-of-the-art solutions in collective communication performance, ML training, and schedule generation speed.

ForestColl: Throughput-Optimal Collective Communications on Heterogeneous Network Fabrics
Contact: liangyurain@gmail.com
NSDI '26

Ring vs ForestColl

Q: Why not just use rings?

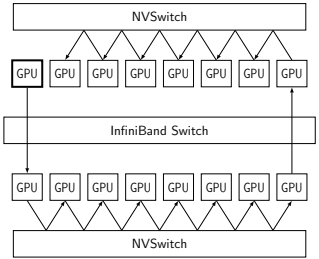


Figure: Ring

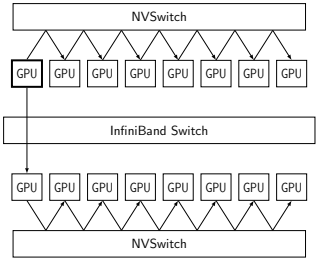


Figure: ForestColl

Ring vs ForestColl

Q: Why not just use rings?

- **Bottleneck:** *inter-box* bandwidth is significantly less than *intra-box* bandwidth.

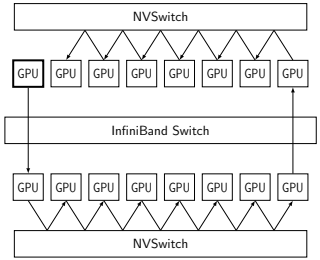


Figure: Ring

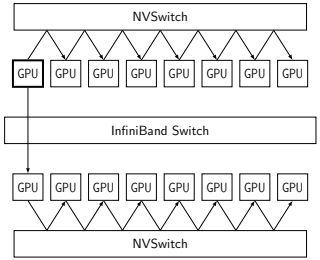


Figure: ForestColl

Ring vs ForestColl

Q: Why not just use rings?

- **Bottleneck:** *inter-box* bandwidth is significantly less than *intra-box* bandwidth.
- Rings often overuse inter-box bandwidth, even though data could be sent intra-box.

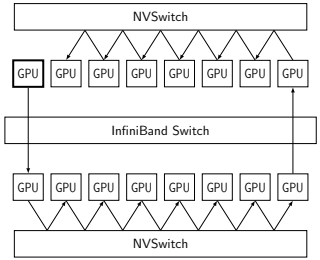


Figure: Ring

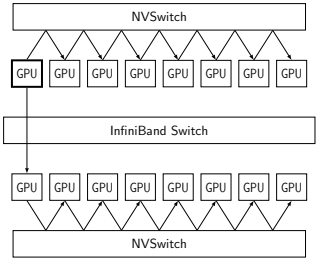


Figure: ForestColl

Ring vs ForestColl

Q: Why not just use rings?

- **Bottleneck:** *inter-box* bandwidth is significantly less than *intra-box* bandwidth.
- Rings often overuse inter-box bandwidth, even though data could be sent intra-box.

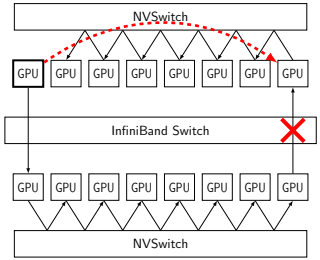


Figure: Ring

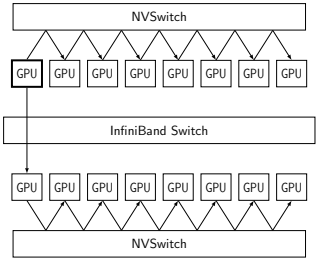


Figure: ForestColl

Ring vs ForestColl

Q: Why not just use rings?

- **Bottleneck:** *inter-box* bandwidth is significantly less than *intra-box* bandwidth.
- Rings often overuse inter-box bandwidth, even though data could be sent intra-box.
  - When all GPUs broadcast simultaneously, ring allgather generates nearly 2x the amount of inter-box traffic compared to ForestColl.

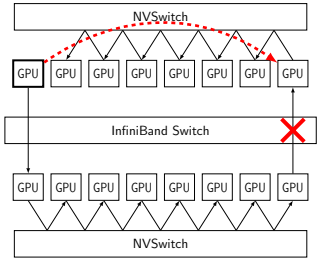


Figure: Ring

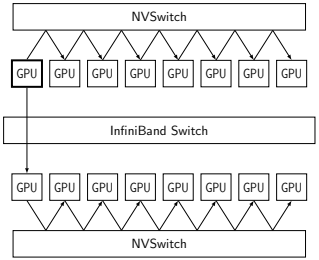


Figure: ForestColl