Communication Algorithm-Architecture Co-Design for Distributed Deep Learning

Jiayi Huang Pritam Majumder Sungkeun Kim
Abdullah Muzahid Ki Hwan Yum EJ Kim

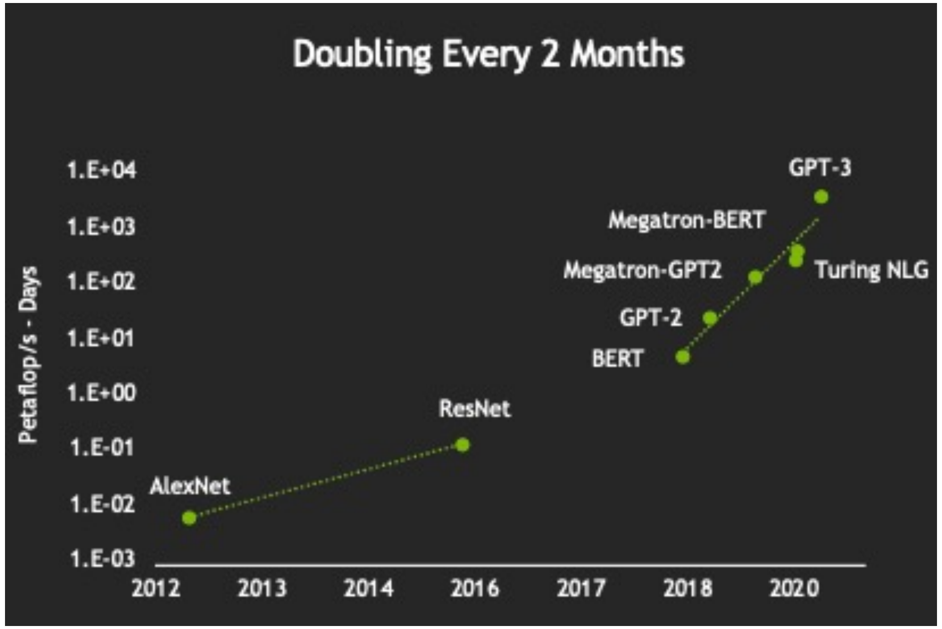
UC Santa Barbara (work done at TAMU) Texas A&M University

Increasing Demand for Distributed Training

- Dataset and model complexity is exploding

Increasing Demand for Distributed Training

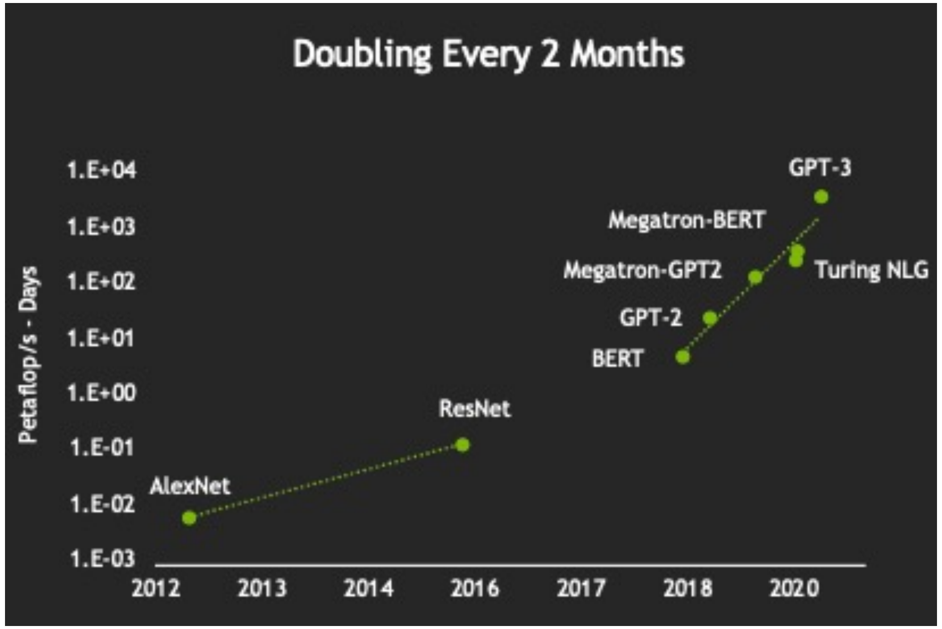
- Dataset and model complexity is exploding

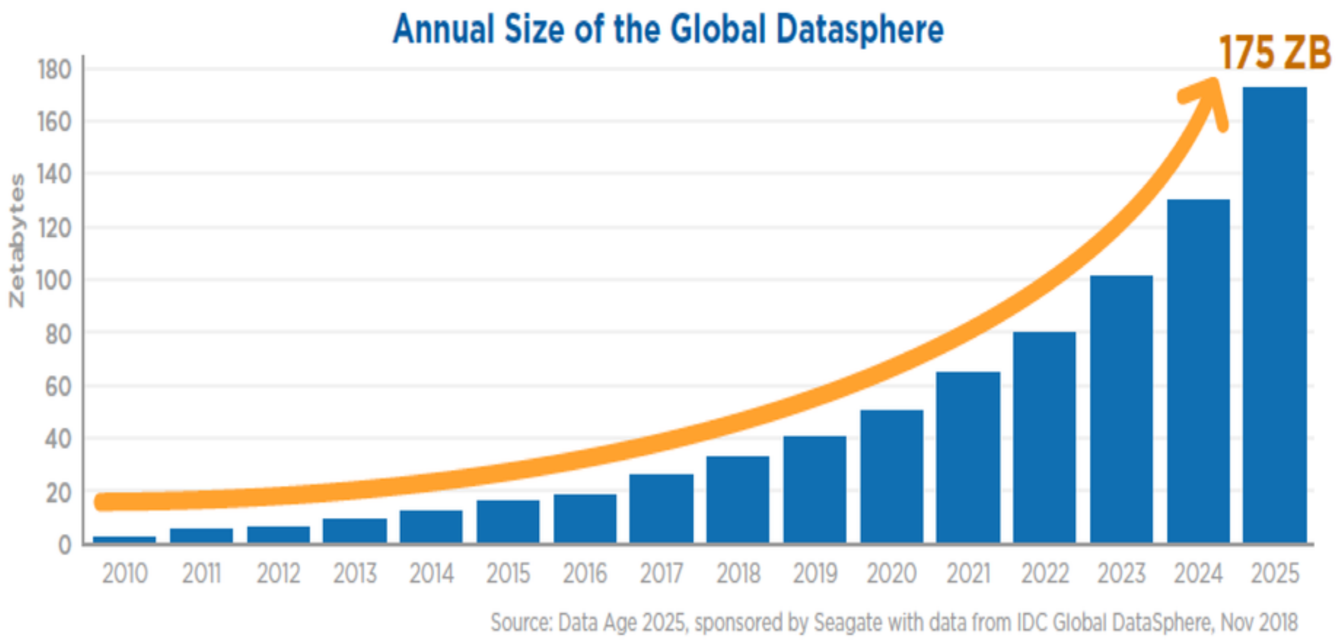


Source: Dally, Logarithmic Numbers and Asynchronous Accumulators, The Future of DL Chips
Chips & Compiler Symposium at MLSys'21

Increasing Demand for Distributed Training

- Dataset and model complexity is exploding

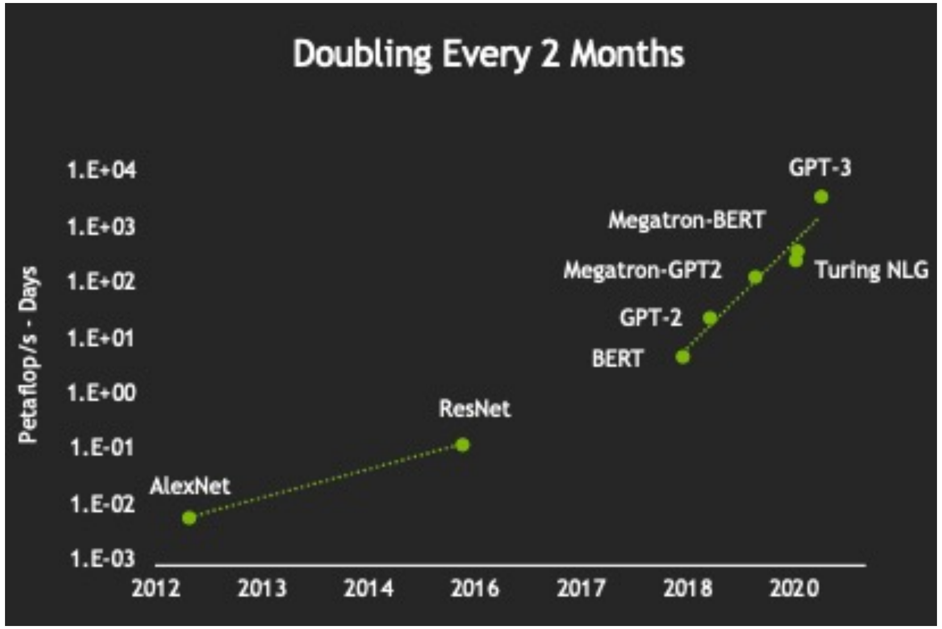


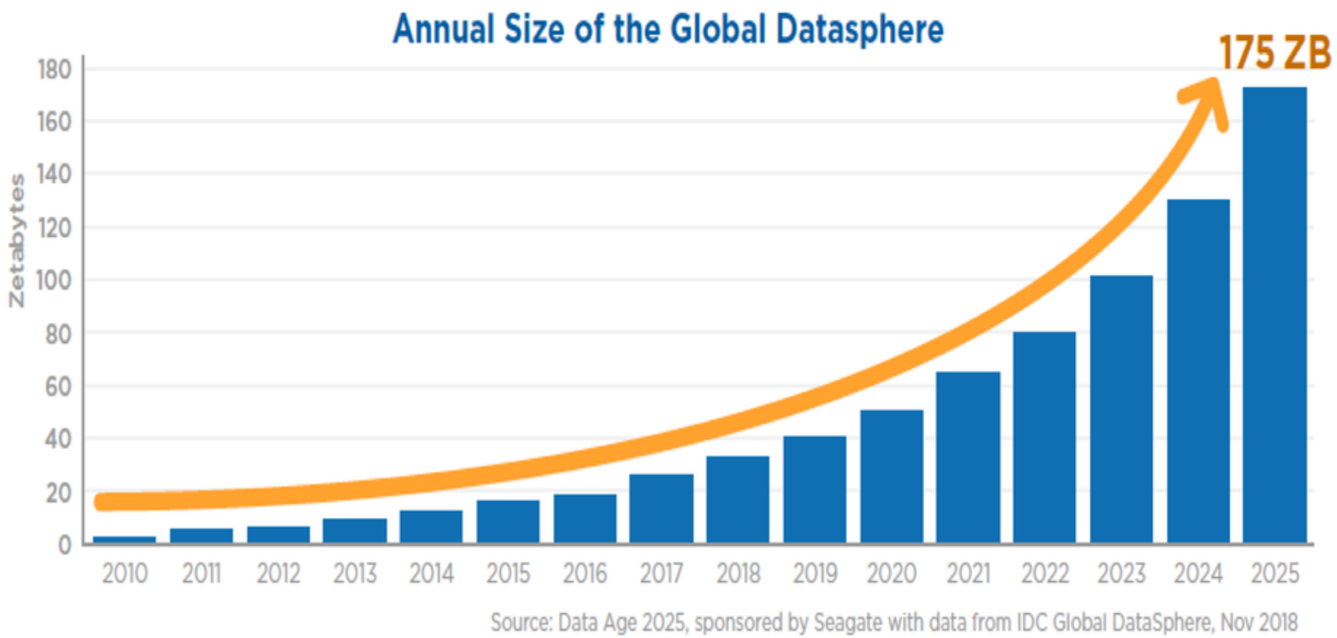


Source: Dally, Logarithmic Numbers and Asynchronous Accumulators, The Future of DL Chips
Chips & Compiler Symposium at MLSys'21

Increasing Demand for Distributed Training

- Dataset and model complexity is exploding





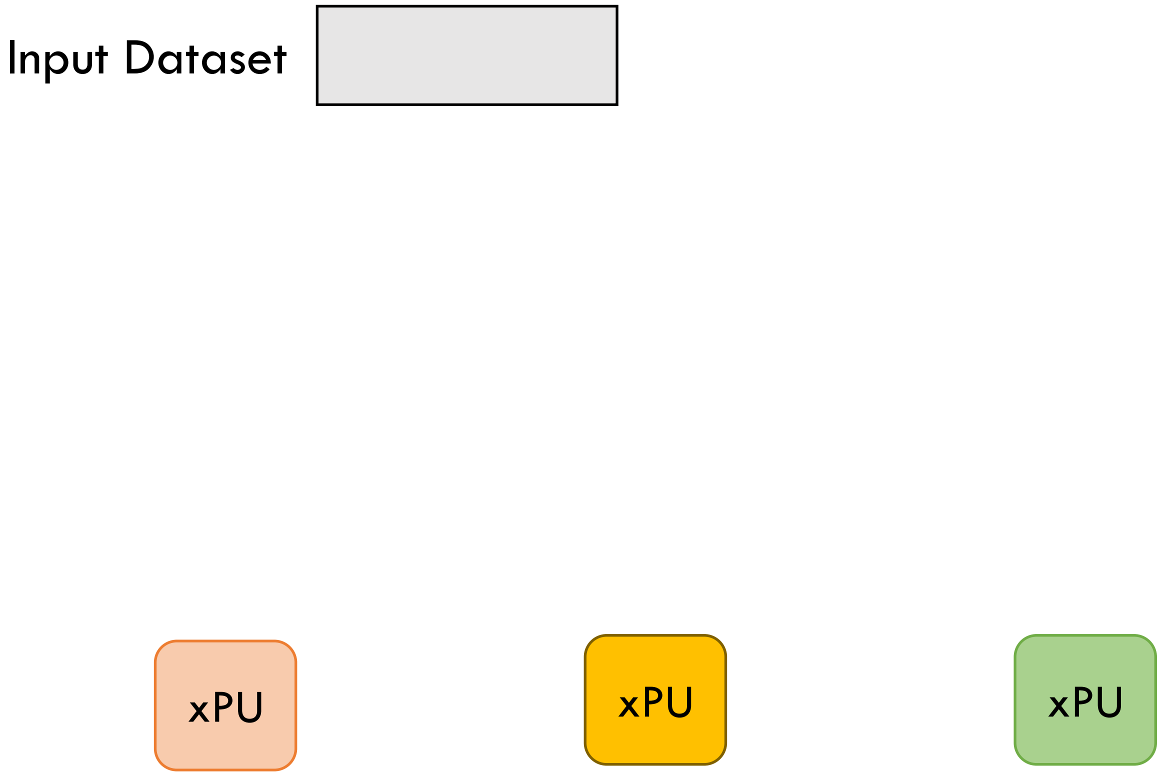
Source: Dally, Logarithmic Numbers and Asynchronous Accumulators, The Future of DL Chips
Chips & Compiler Symposium at MLSys'21

- GPT-3 – trained on part of an 10,000-GPU cluster* [Brown+ 2020]

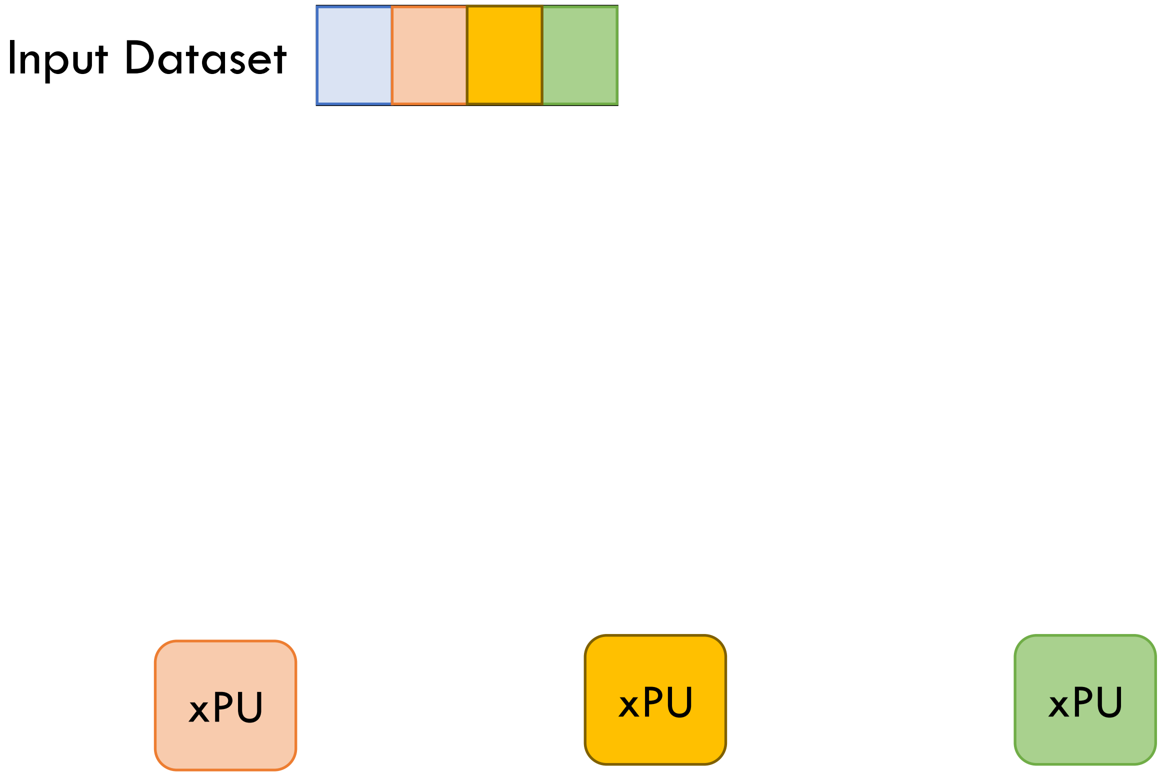
*Source: https://developer.nvidia.com/blog/openai-presents-gpt-3-a-175-billion-parameters-language-model/

Data-Parallel Training

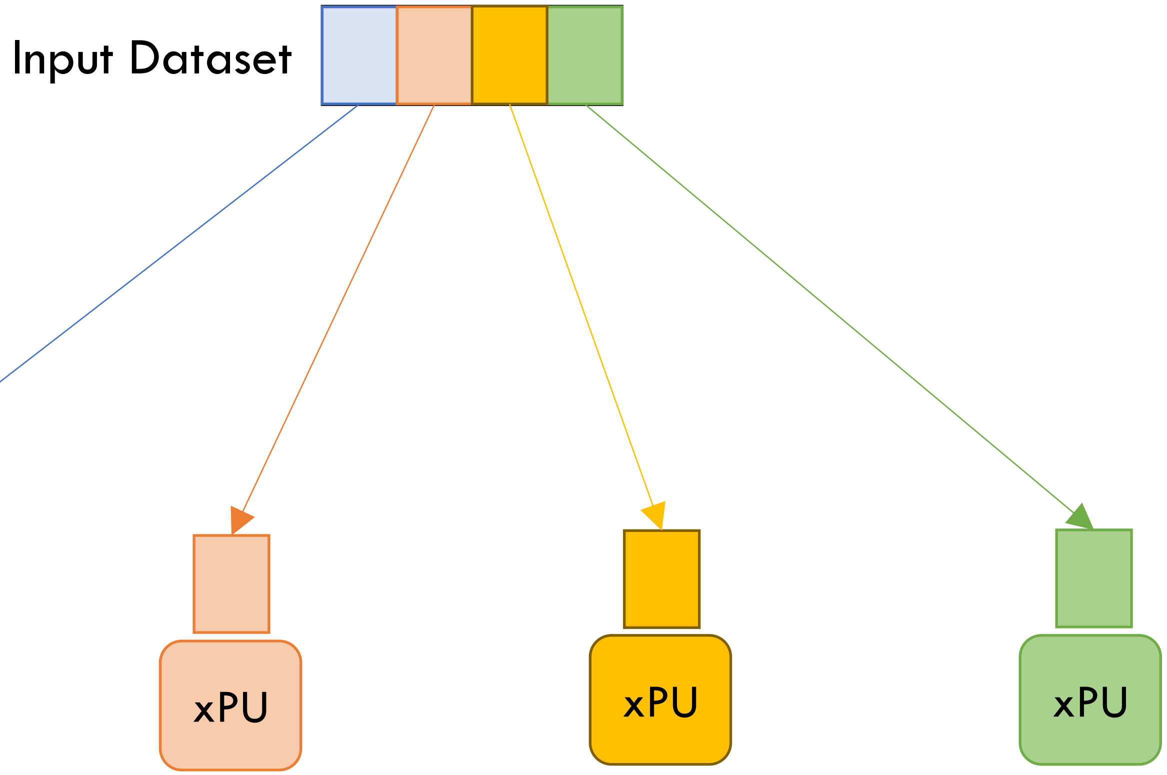
Data-Parallel Training



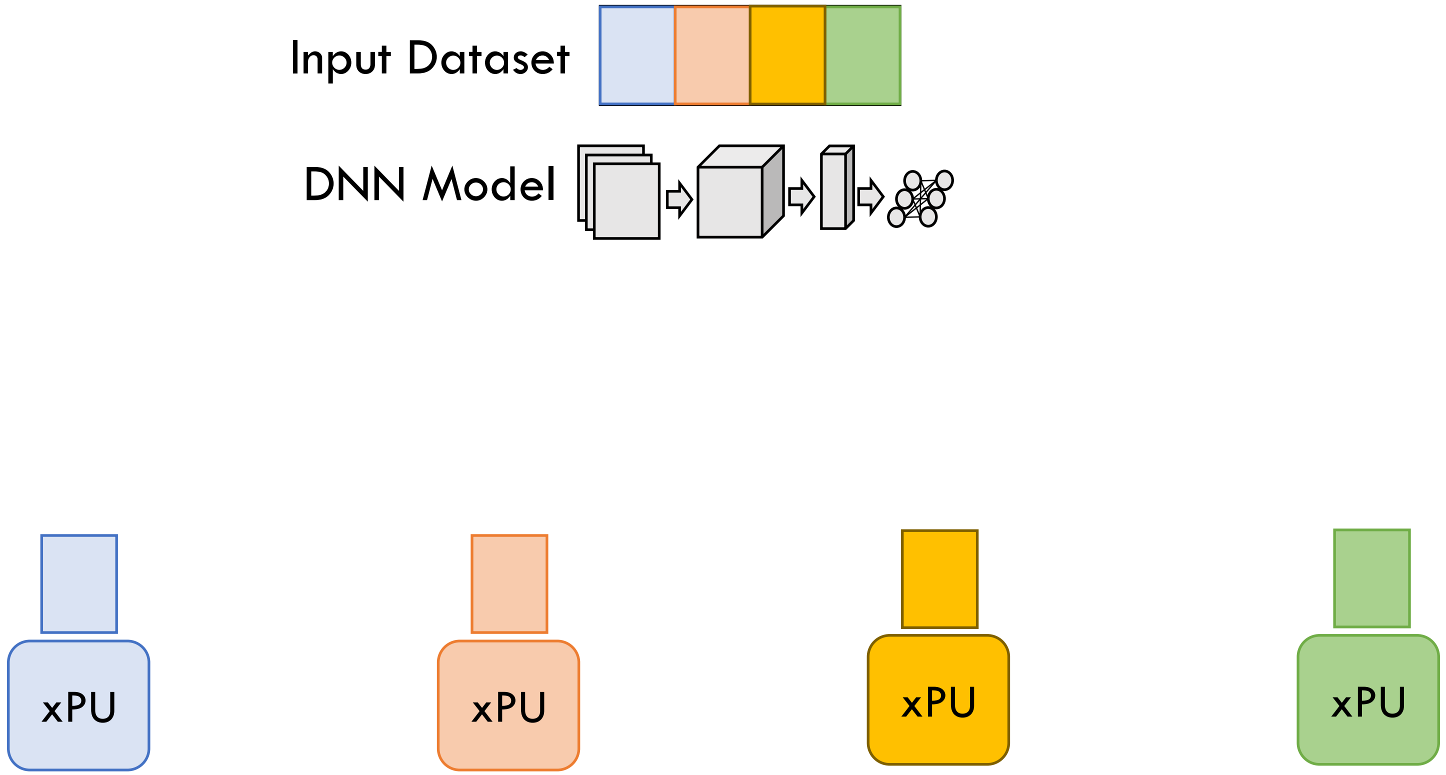
Data-Parallel Training



Data-Parallel Training

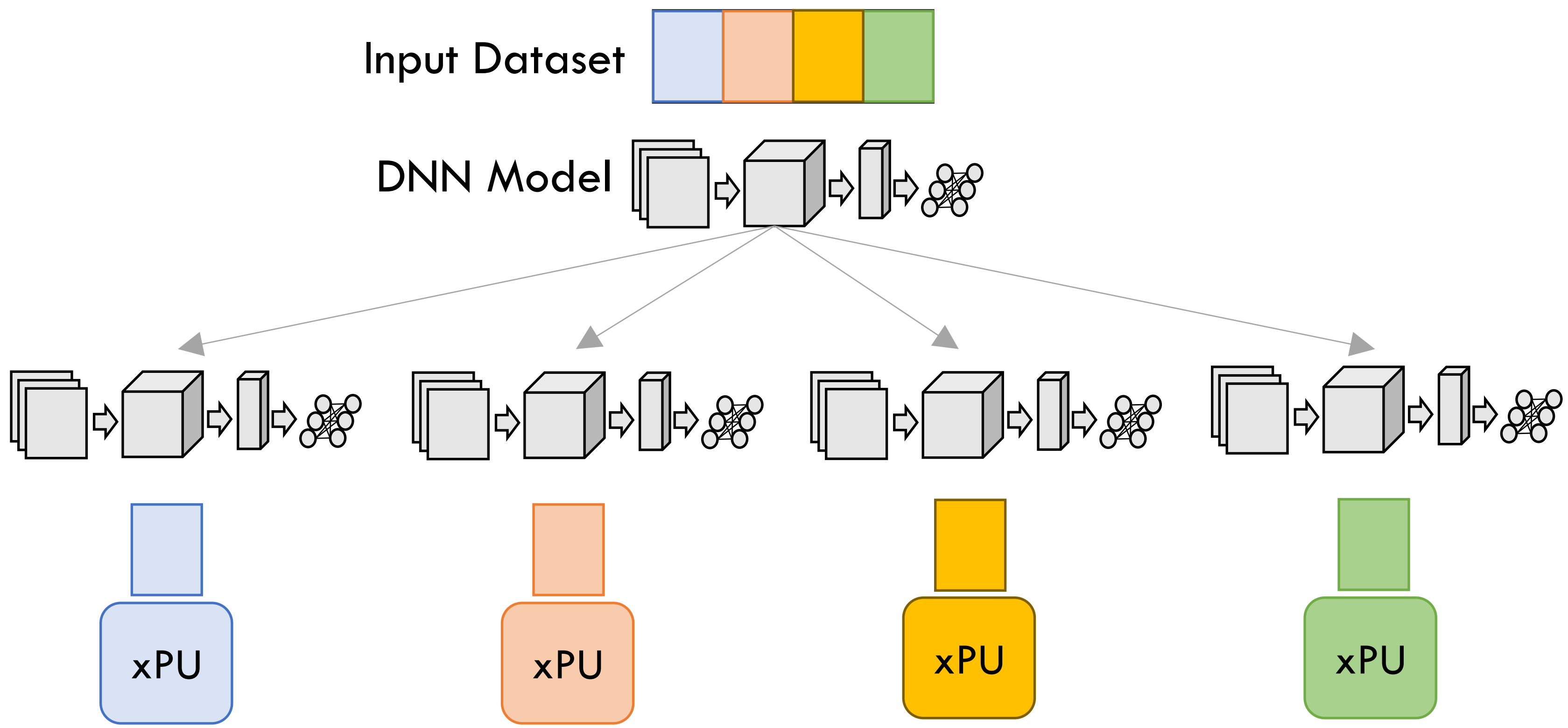


Data-Parallel Training



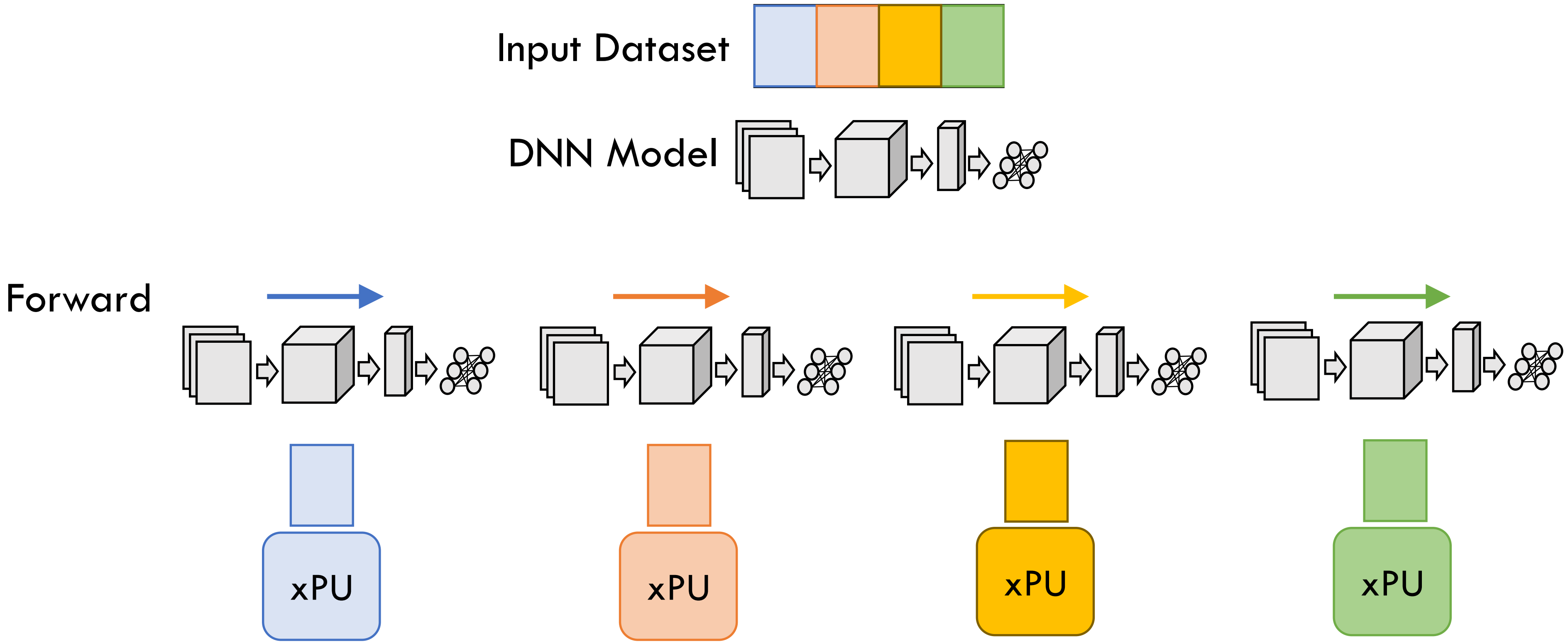
DNN model redrawn from Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019

Data-Parallel Training



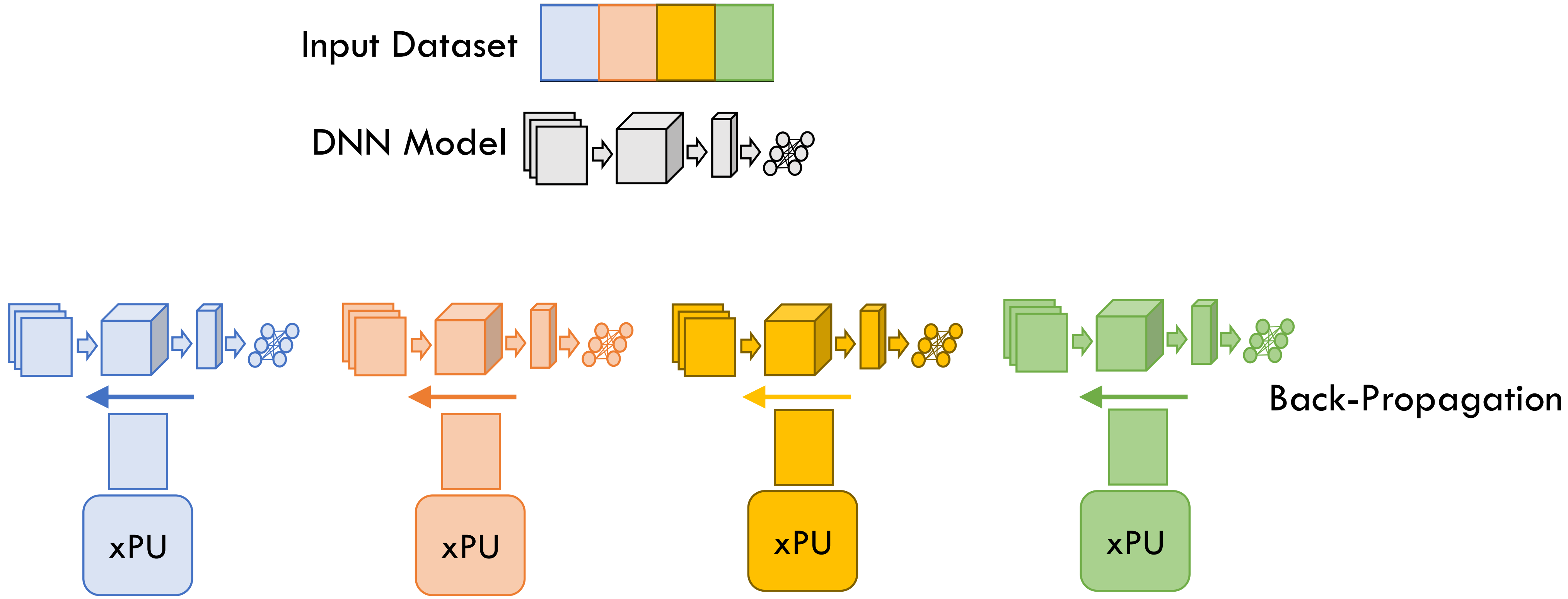
DNN model redrawn from Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019

Data-Parallel Training



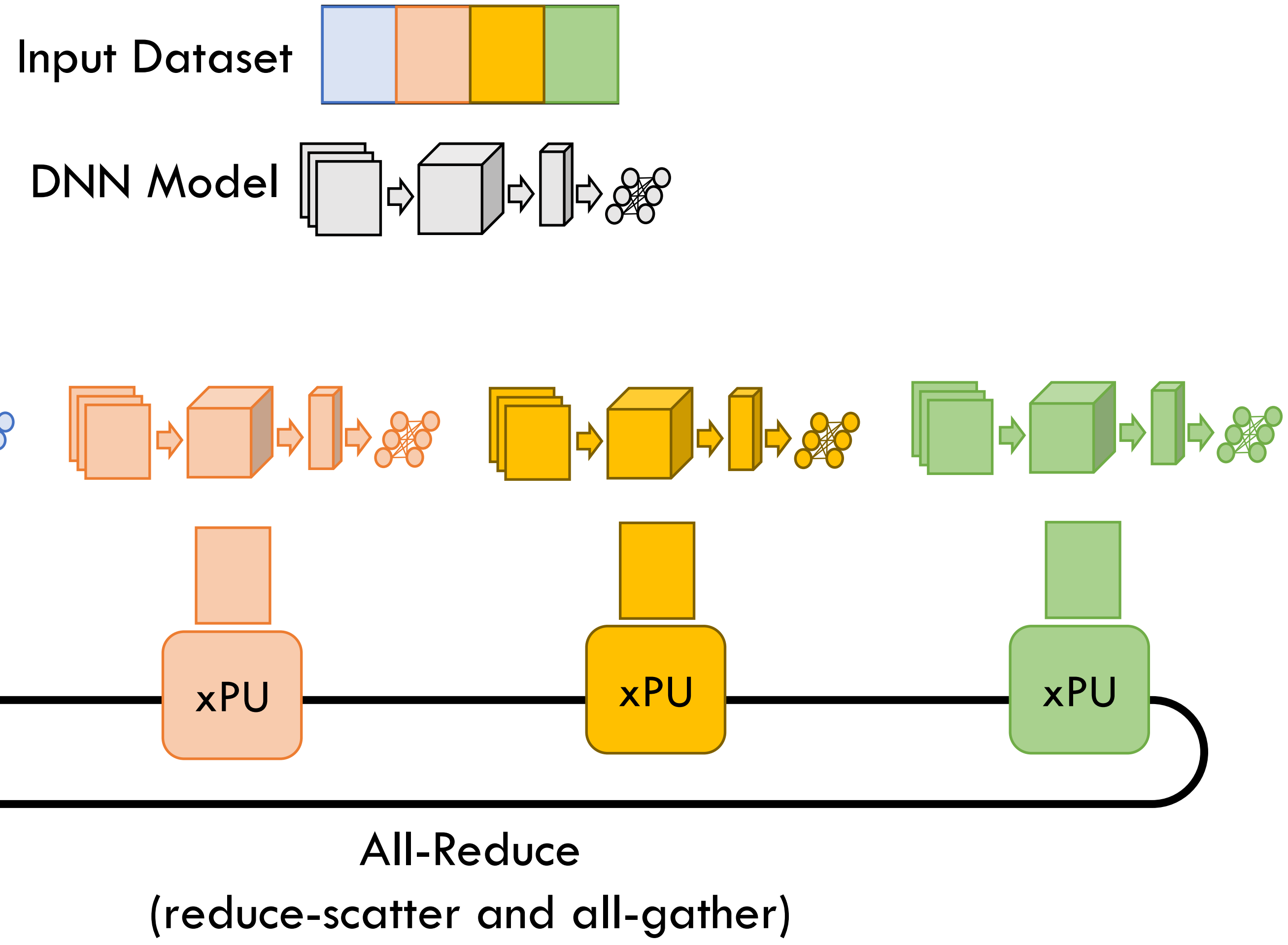
DNN model redrawn from Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019

Data-Parallel Training



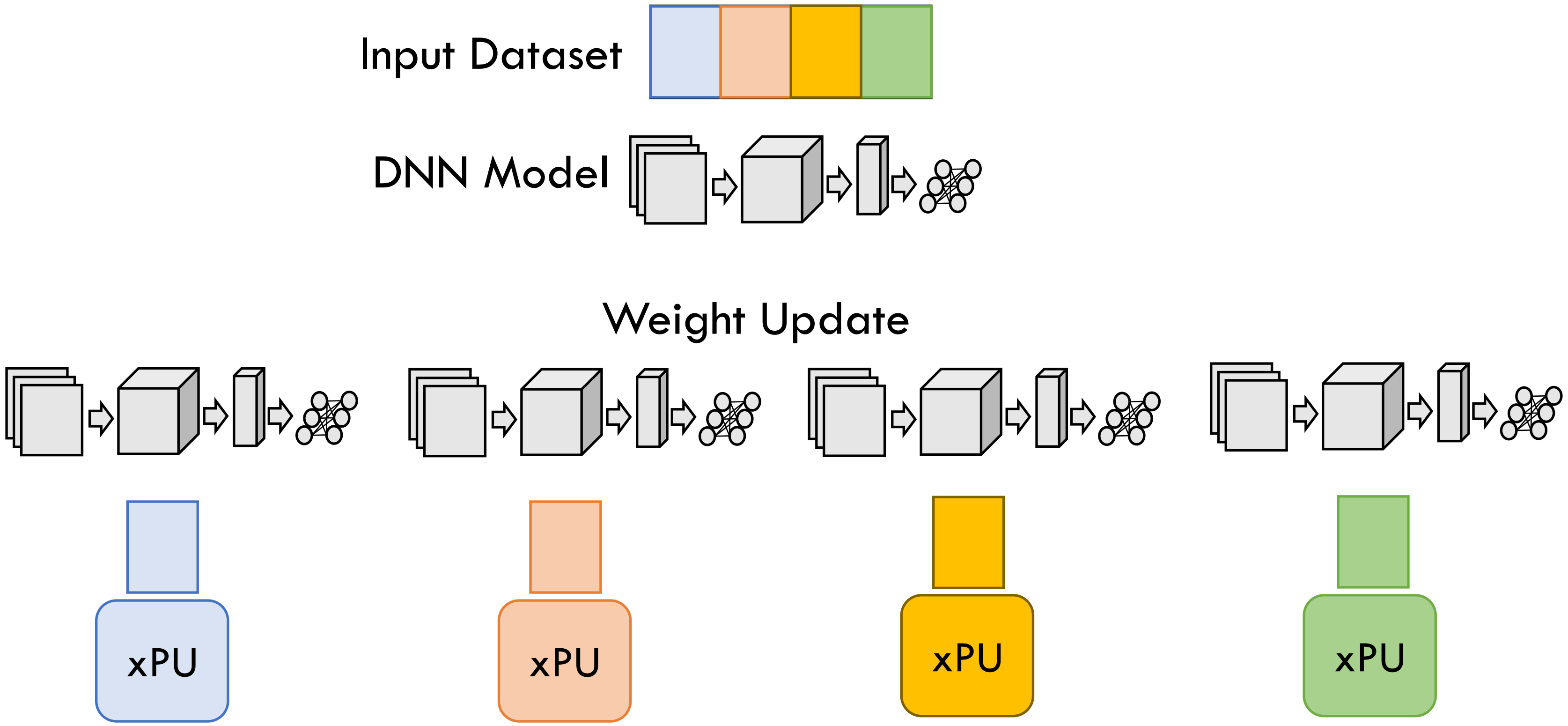
DNN model redrawn from Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019

Data-Parallel Training



DNN model redrawn from Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019

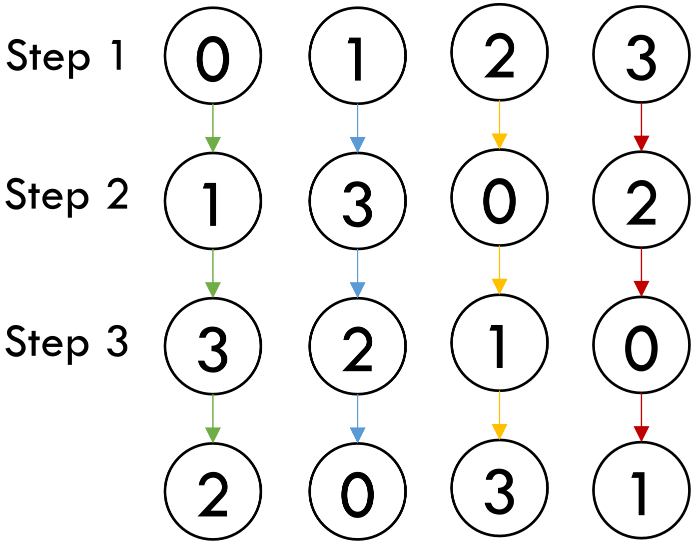
Data-Parallel Training



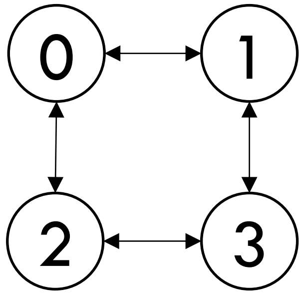
DNN model redrawn from Ben-Nun+ ACM Computing Surveys, vol. 52, no. 4, August 2019

Limitations in Existing All-Reduce Algorithms

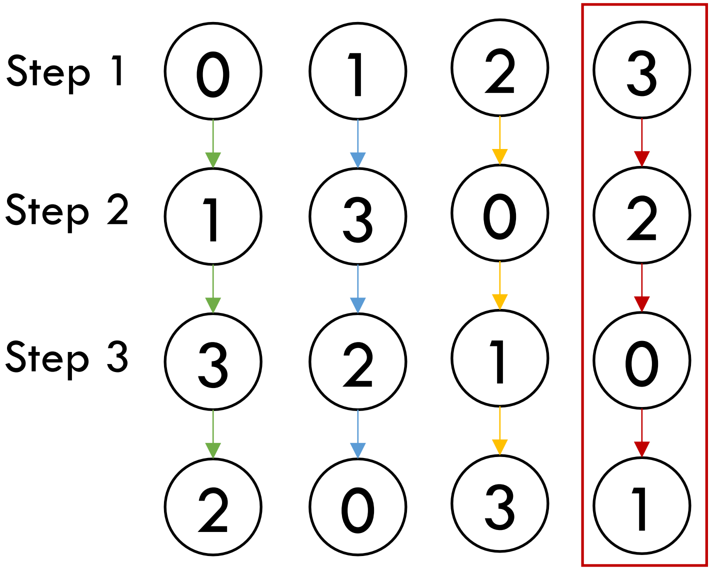
Limitations in Existing All-Reduce Algorithms



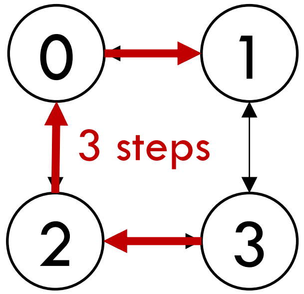
Ring All-Gather



Limitations in Existing All-Reduce Algorithms



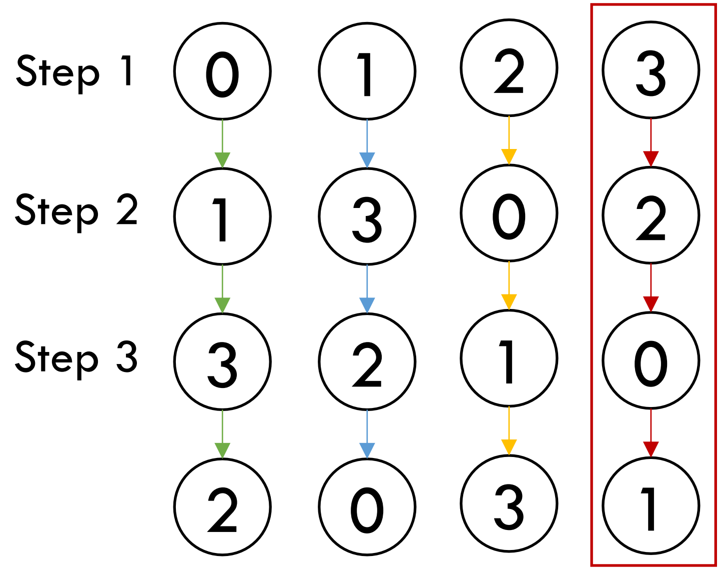
Ring All-Gather



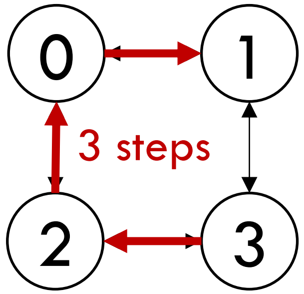
Linear to #nodes - 1

Long latency for small data

Limitations in Existing All-Reduce Algorithms



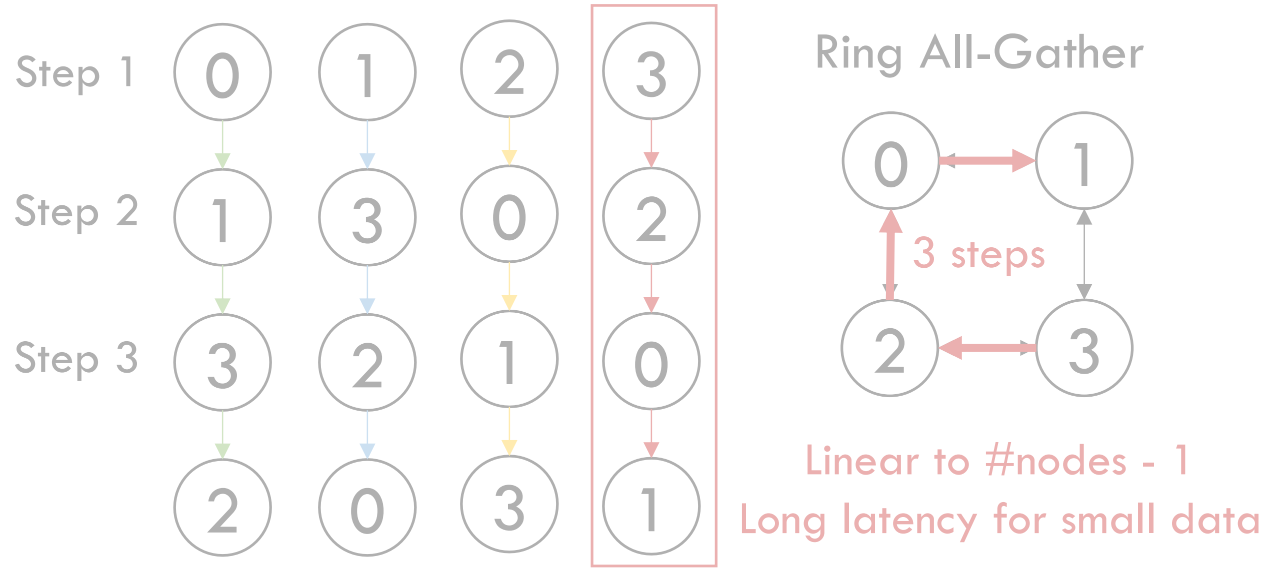
Ring All-Gather

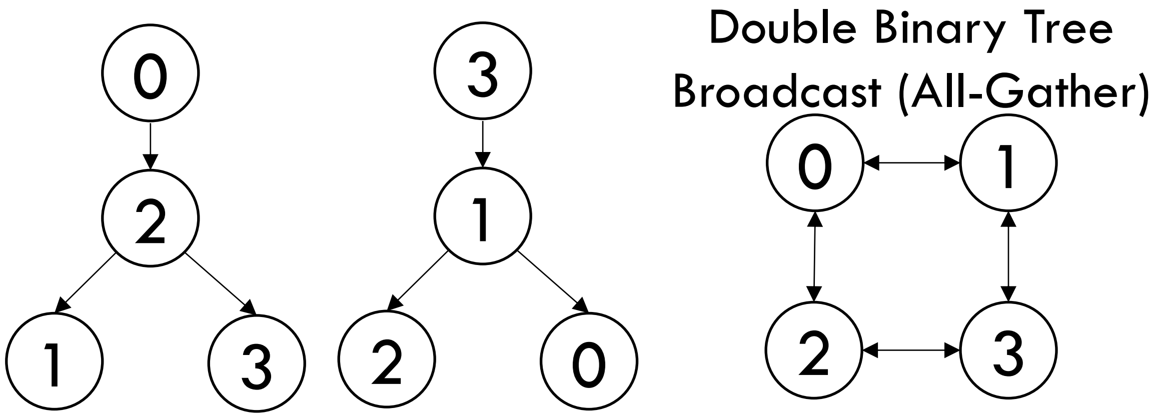


Linear to #nodes - 1
Long latency for small data

Algorithms	**(Small data) Latency**	**Large data**		**Applied Well on Various Topologies**
		Bandwidth	**Contention**	
Ring [Patarasuk+Yuan JPDC'09]	high	optimal	none	✓

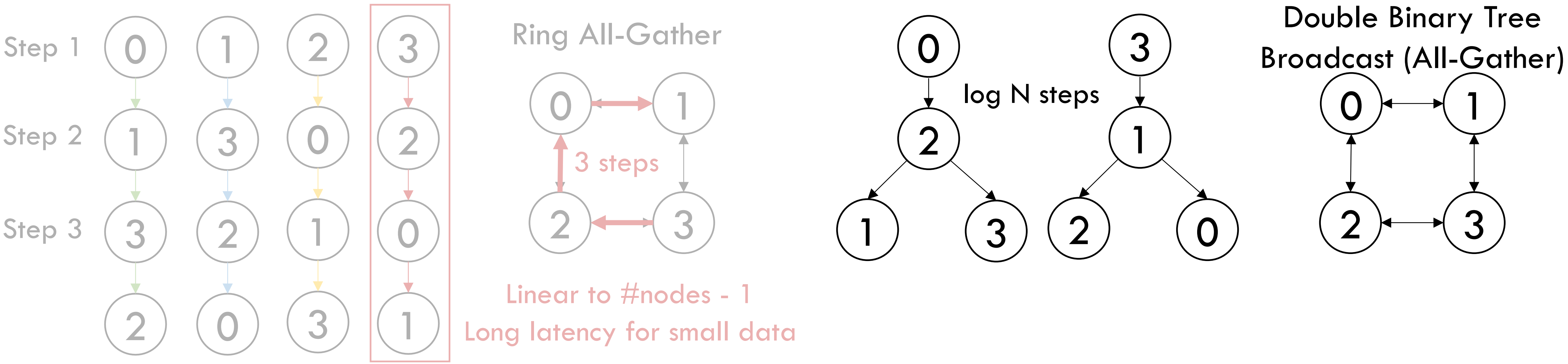
Limitations in Existing All-Reduce Algorithms





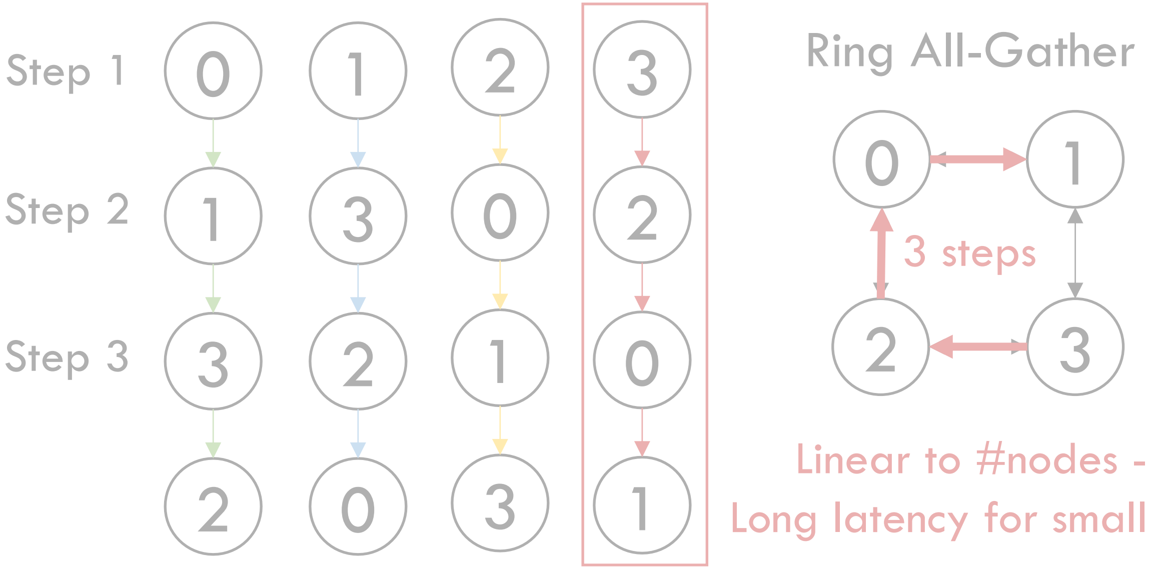
Algorithms	(Small data) Latency	Large data: Bandwidth	Large data: Contention	Applied Well on Various Topologies
Ring [Patarasuk+Yuan JPDC'09]	high	optimal	none	✓

Limitations in Existing All-Reduce Algorithms

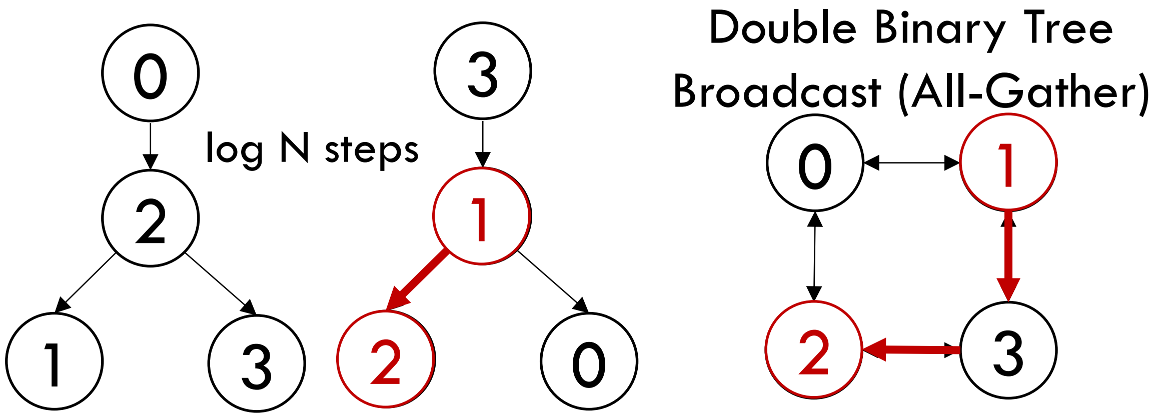


Algorithms	(Small data) Latency	Large data: Bandwidth	Large data: Contention	Applied Well on Various Topologies
Ring [Patarasuk+Yuan JPDC'09]	high	optimal	none	✓
Double binary tree [Sanders+ JPC'09]	low	optimal	high	× (Topology-oblivious)

Limitations in Existing All-Reduce Algorithms



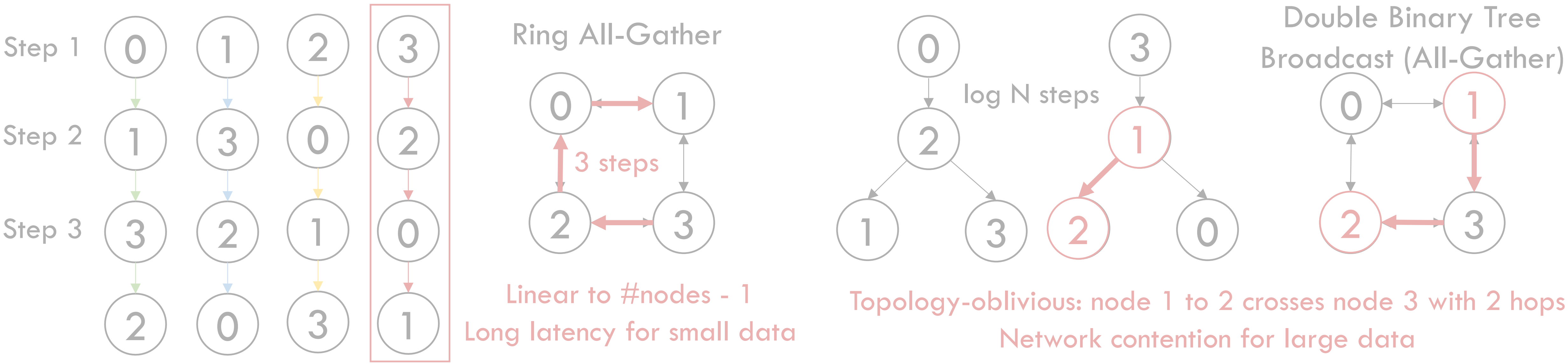
Linear to #nodes - 1
Long latency for small data



Topology-oblivious: node 1 to 2 crosses node 3 with 2 hops
Network contention for large data

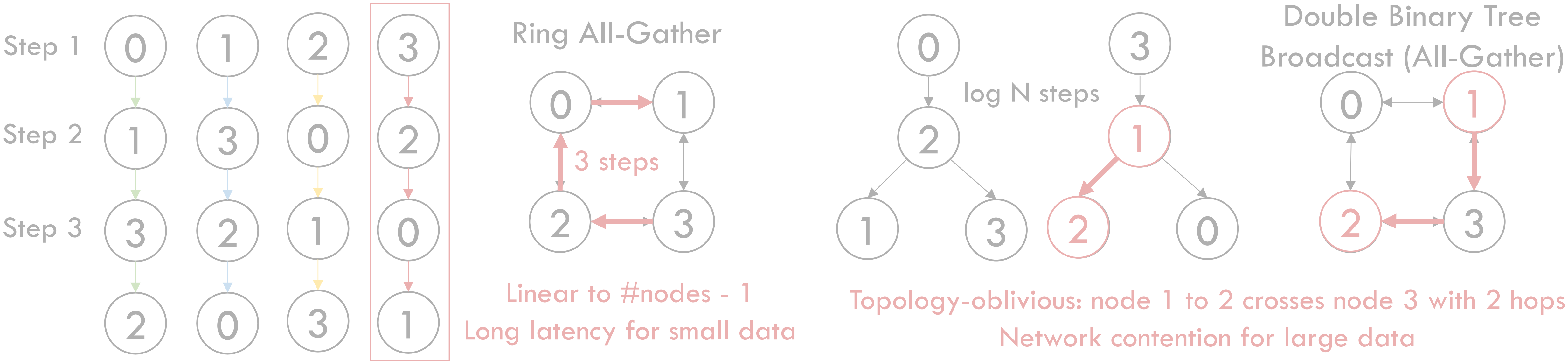
Algorithms	(Small data) Latency	Large data Bandwidth	Large data Contention	Applied Well on Various Topologies
Ring [Patarasuk+Yuan JPDC'09]	high	optimal	none	✓
Double binary tree [Sanders+ JPC'09]	low	optimal	high	✗ (Topology-oblivious)

Limitations in Existing All-Reduce Algorithms



Algorithms	(Small data) Latency	Large data		Applied Well on Various Topologies
		Bandwidth	Contention	
Ring [Patarasuk+Yuan JPDC'09]	high	optimal	none	✓
Double binary tree [Sanders+ JPC'09]	low	optimal	high	✕ (Topology-oblivious)
2D-Ring [Ying+ NeurIPsW'18]	low	sub-optimal	none	✕ (2D Torus/Mesh)

Limitations in Existing All-Reduce Algorithms



Algorithms	(Small data) Latency	Large data Bandwidth	Contention	Applied Well on Various Topologies
Ring [Patarasuk+Yuan JPDC'09]	high	optimal	none	√
Double binary tree [Sanders+ JPC'09]	low	optimal	high	× (Topology-oblivious)
2D-Ring [Ying+ NeurIPsW'18]	low	sub-optimal	none	× (2D Torus/Mesh)
HDRM [Dong+ HPCA'20]	low	optimal	none	× (EFLOPS's BiGraph)

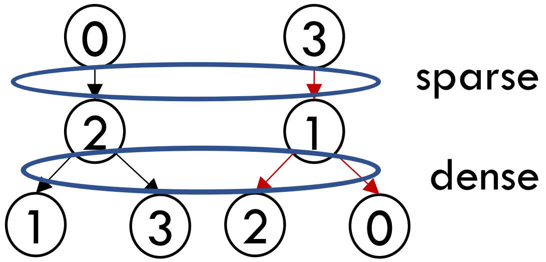
MultiTree: Algorithm-Architecture Co-Design

- Topology-aware All-Reduce Algorithm
  - Low latency and high bandwidth, applicable to different topologies
- Hardware-based All-Reduce Scheduling
  - Contention-free communication
- Message-based Flow Control
  - Exploit bulk transfer of large gradients for near perfect link bandwidth

MultiTree: Algorithm-Architecture Co-Design

- Topology-aware All-Reduce Algorithm
  - Low latency and high bandwidth, applicable to different topologies
- Hardware-based All-Reduce Scheduling
  - Contention-free communication
- Message-based Flow Control
  - Exploit bulk transfer of large gradients for near perfect link bandwidth

Double Binary Tree

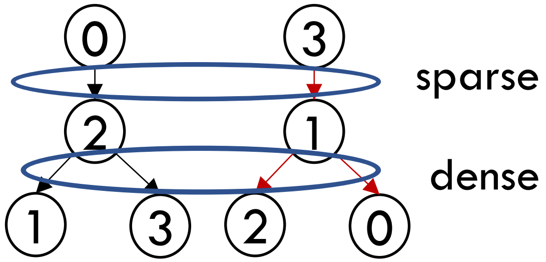


- Insight
  - *Tree levels closer to leaves are denser than tree levels closer to roots*
  - Top-down for tree construction: move more communications to roots

MultiTree: Algorithm-Architecture Co-Design

- Topology-aware All-Reduce Algorithm
  Low latency and high bandwidth, applicable to different topologies
- Hardware-based All-Reduce Scheduling
  Contention-free communication
- Message-based Flow Control
  Exploit bulk transfer of large gradients for near perfect link bandwidth

Double Binary Tree

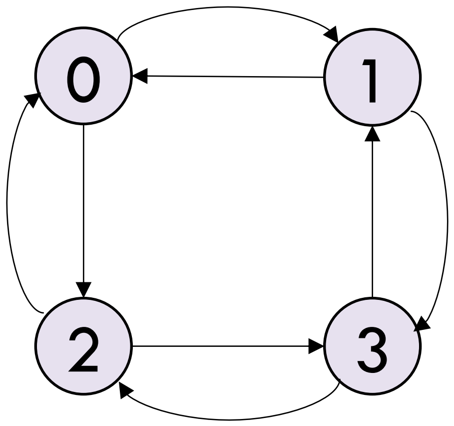


- Insight
  - *Tree levels closer to leaves are denser than tree levels closer to roots*
  - Top-down for tree construction: move more communications to roots

- Approach: tree constructions as a link allocation problem
  - Allocate link for each time step (level) to build the trees progressively

MultiTree Construction Example (Time Step 1)

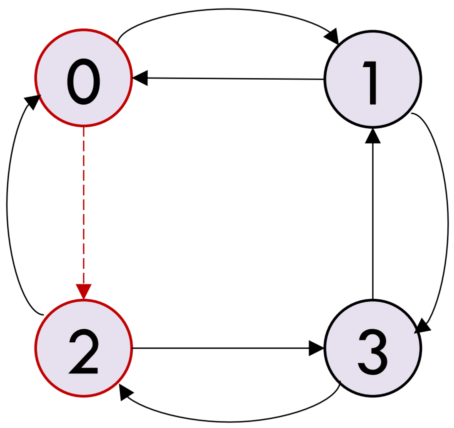
Construct 4 spanning trees for a 4-node system



Link allocation for time step 1 (tree level 1)

MultiTree Construction Example (Time Step 1)

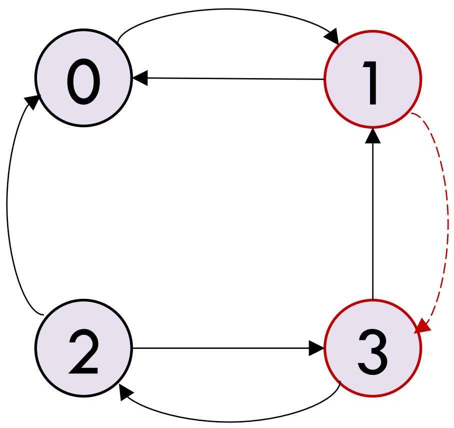
Construct 4 spanning trees for a 4-node system



Link allocation for time step 1 (tree level 1)

MultiTree Construction Example (Time Step 1)

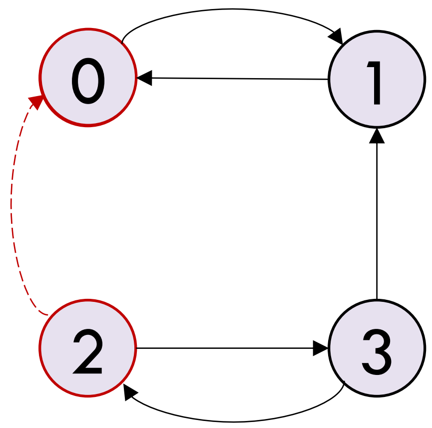
Construct 4 spanning trees for a 4-node system



Link allocation for time step 1 (tree level 1)

MultiTree Construction Example (Time Step 1)

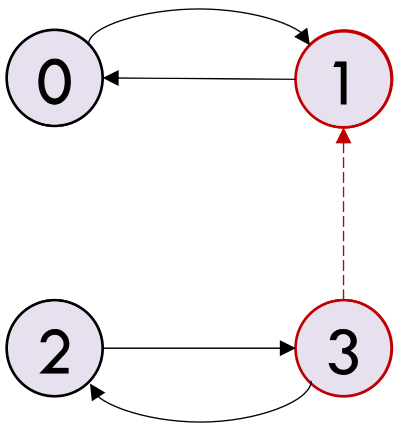
Construct 4 spanning trees for a 4-node system



Link allocation for time step 1 (tree level 1)

MultiTree Construction Example (Time Step 1)

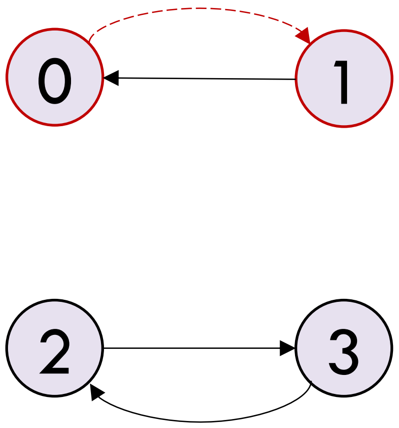
Construct 4 spanning trees for a 4-node system



Link allocation for time step 1 (tree level 1)

MultiTree Construction Example (Time Step 1)

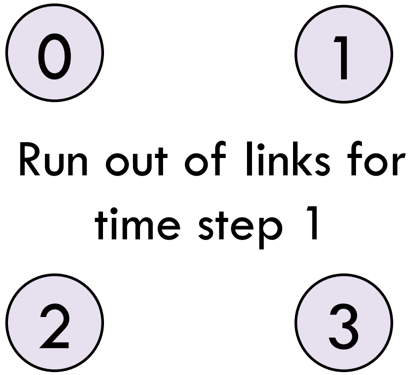
Construct 4 spanning trees for a 4-node system



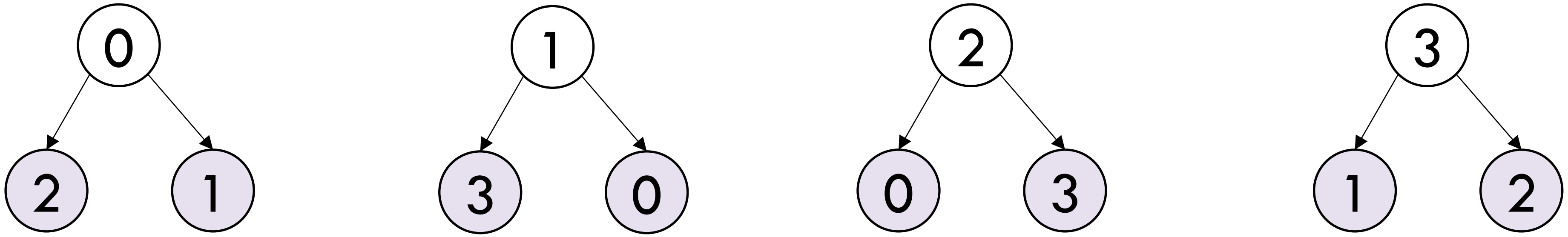
Link allocation for time step 1 (tree level 1)

MultiTree Construction Example (Time Step 1)

Construct 4 spanning trees for a 4-node system

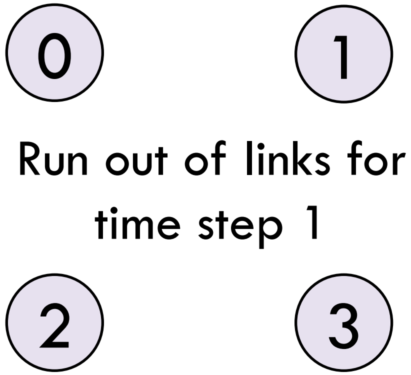


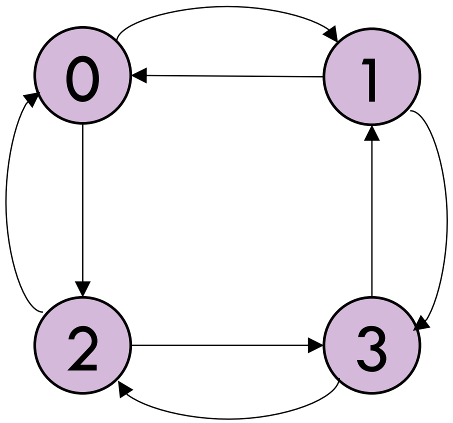
Link allocation for time step 1 (tree level 1)



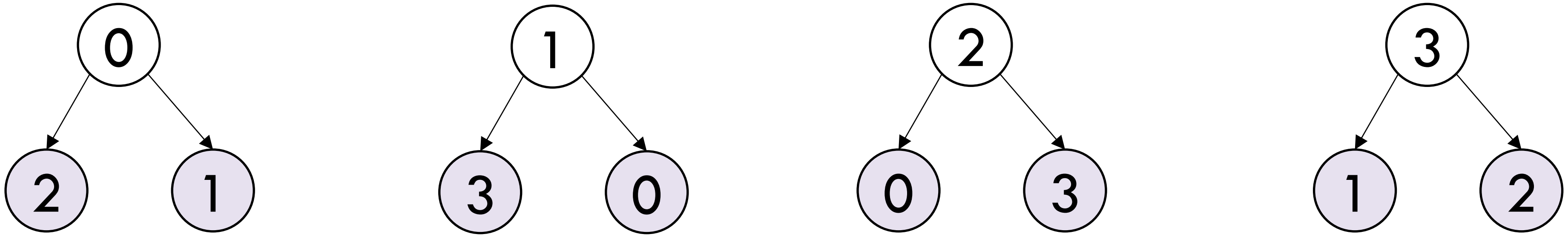
MultiTree Construction Example (Time Step 2)

Construct 4 spanning trees for a 4-node system



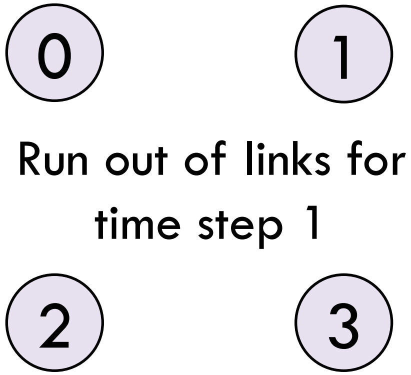


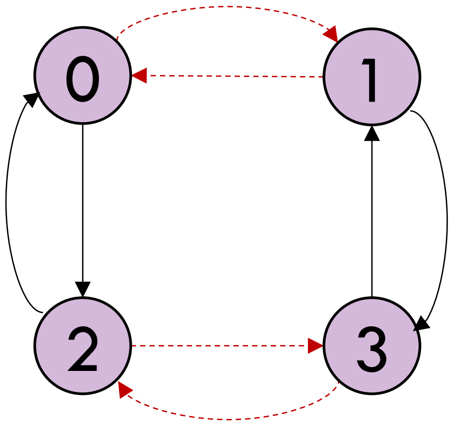
Link allocation for time step 2 (tree level 2)



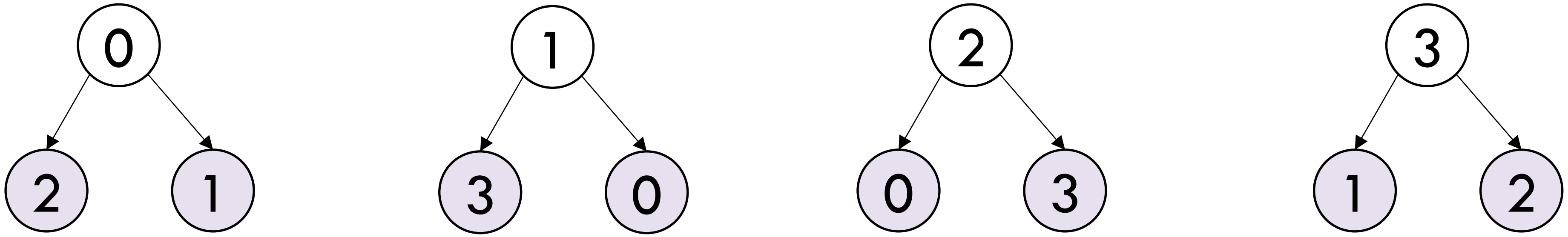
MultiTree Construction Example (Time Step 2)

Construct 4 spanning trees for a 4-node system



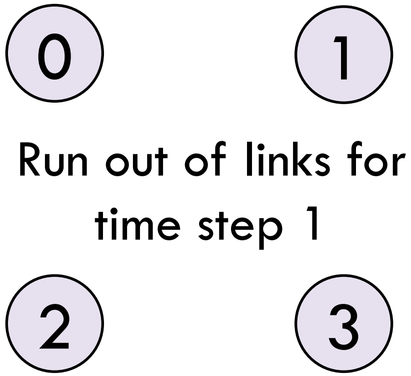


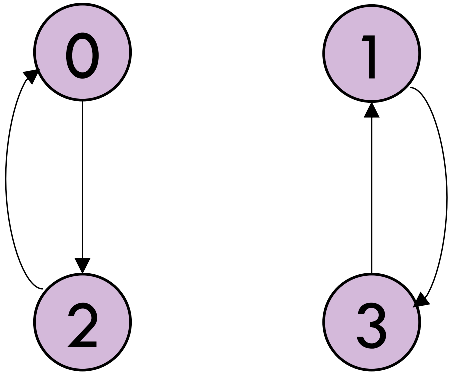
Link allocation for time step 2 (tree level 2)



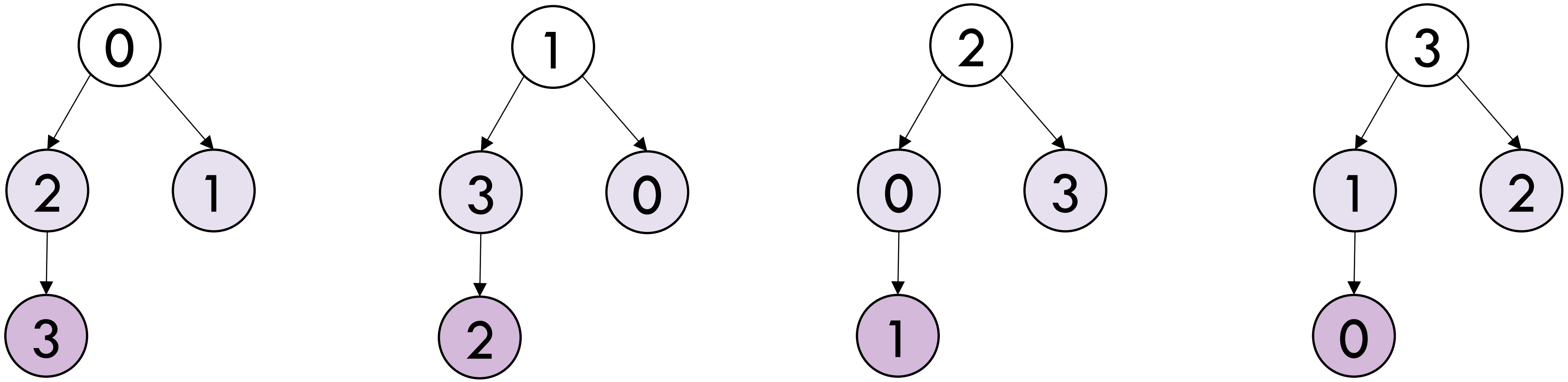
MultiTree Construction Example (Time Step 2)

Construct 4 spanning trees for a 4-node system

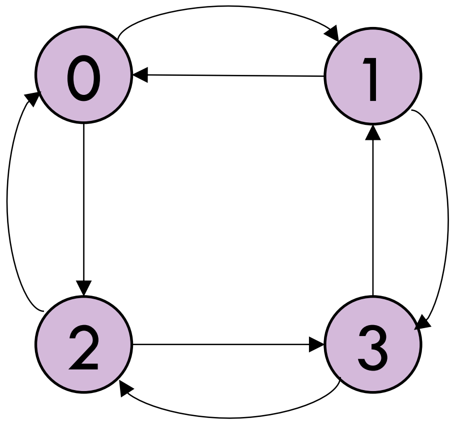




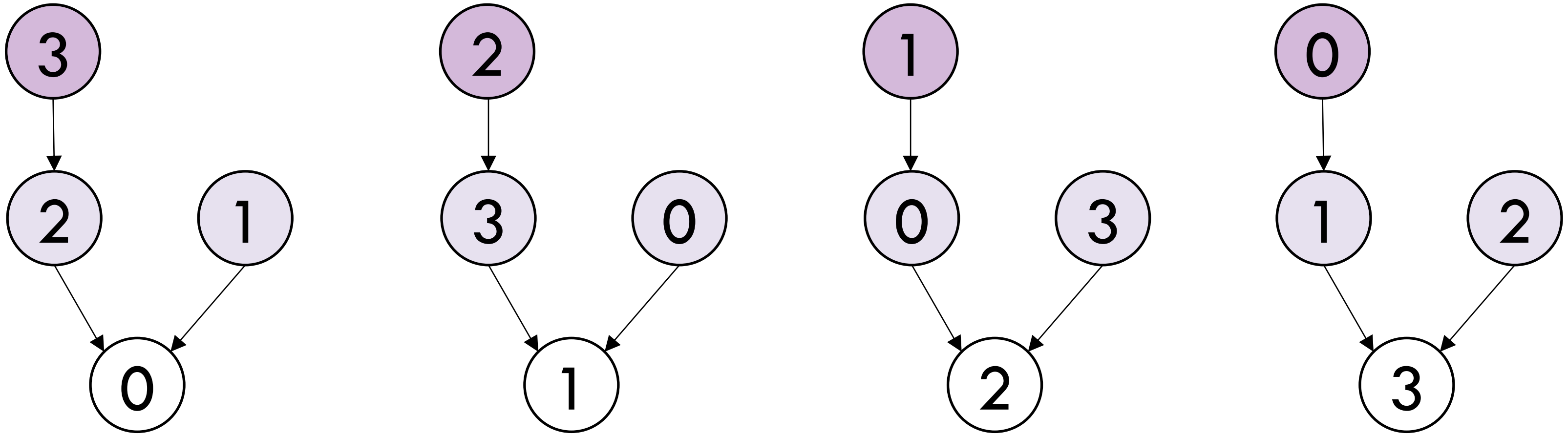
Link allocation for time step 2 (tree level 2)



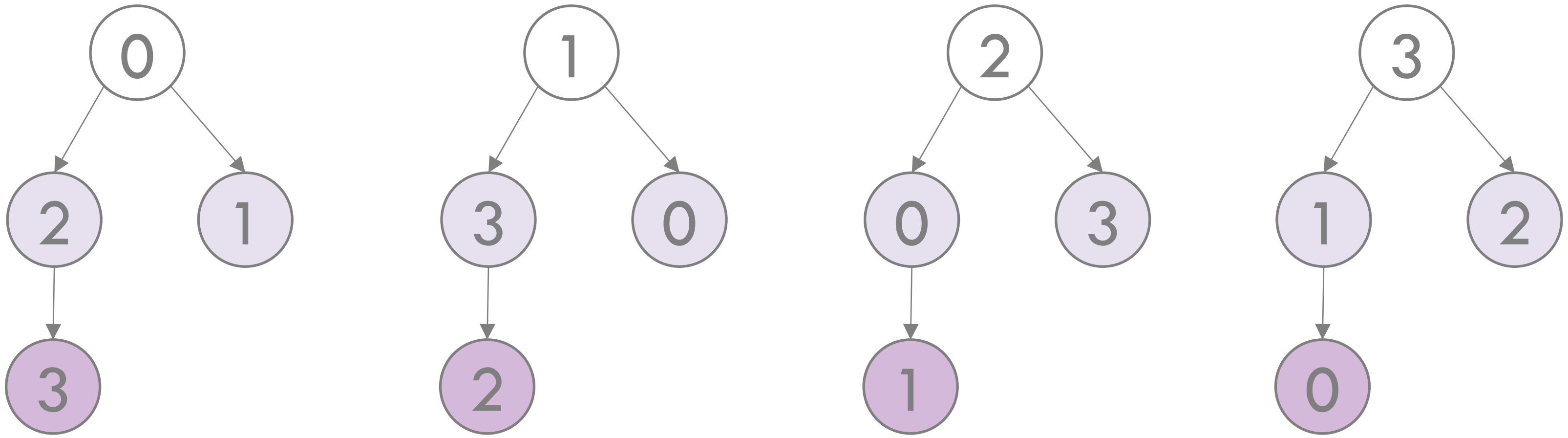
MultiTree All-Reduce: Reduce-Scatter



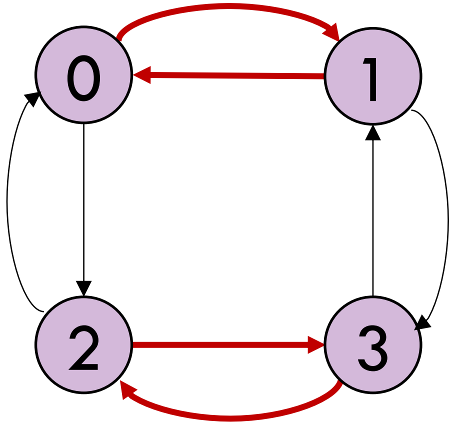
Reduce-Scatter (reduction from leaf level to root)



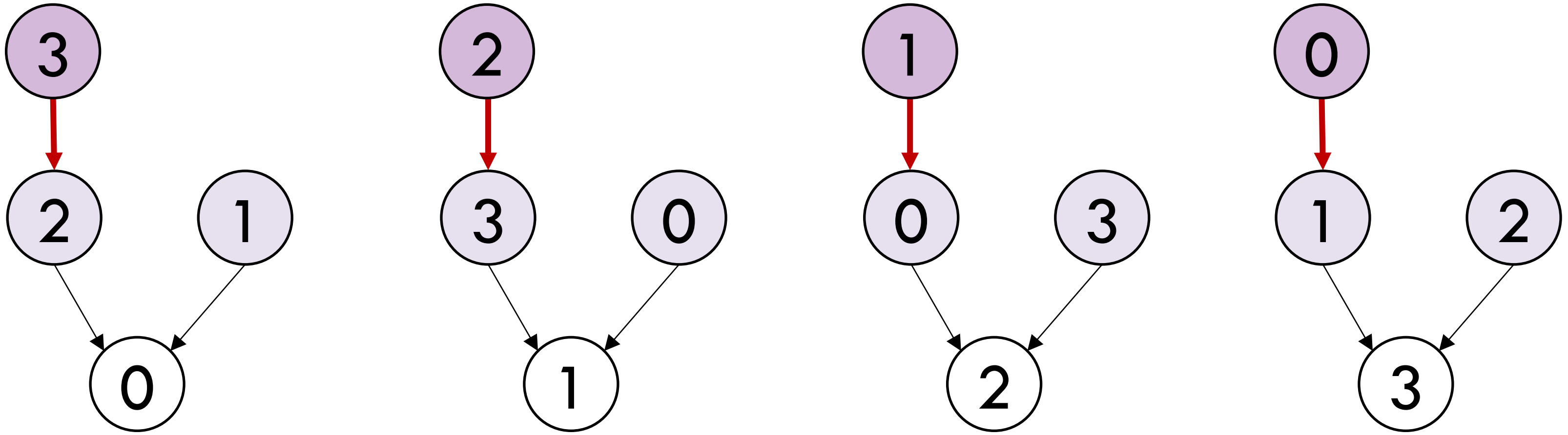
All-Gather (broadcast from root to leaf level)



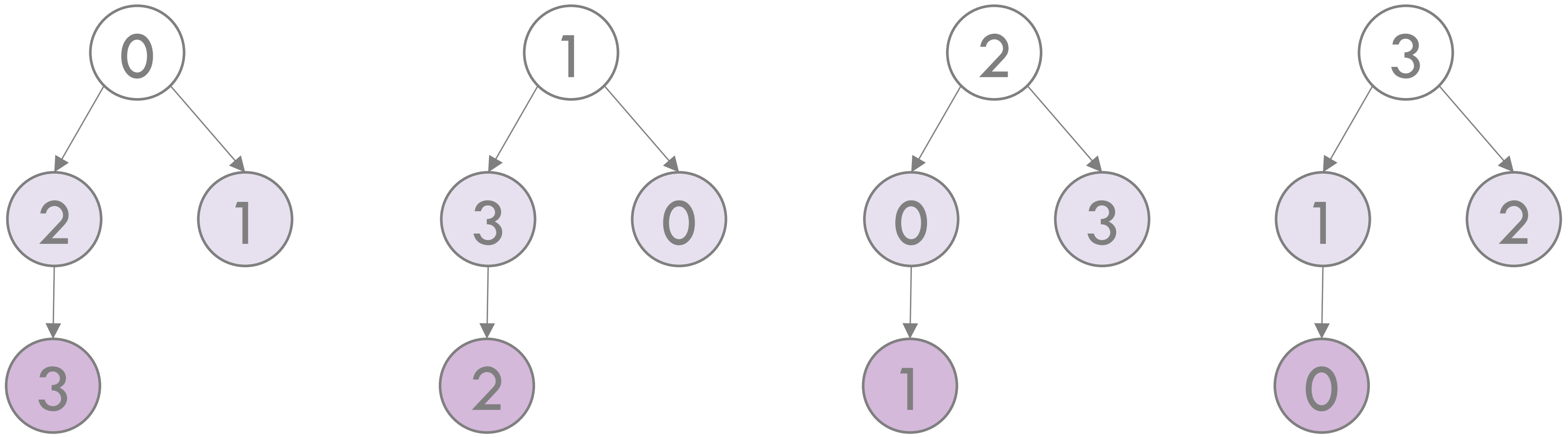
MultiTree All-Reduce: Reduce-Scatter



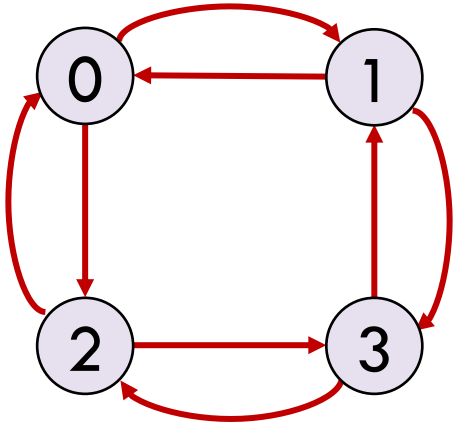
Reduce-Scatter (reduction from leaf level to root)



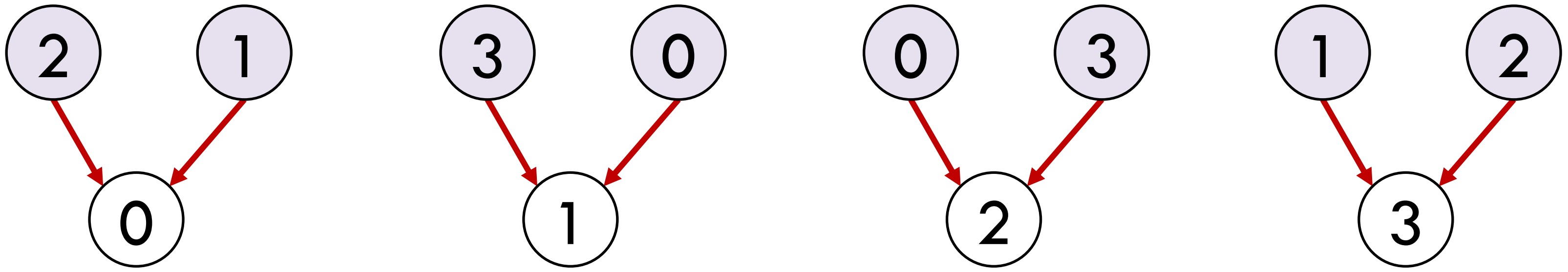
All-Gather (broadcast from root to leaf level)



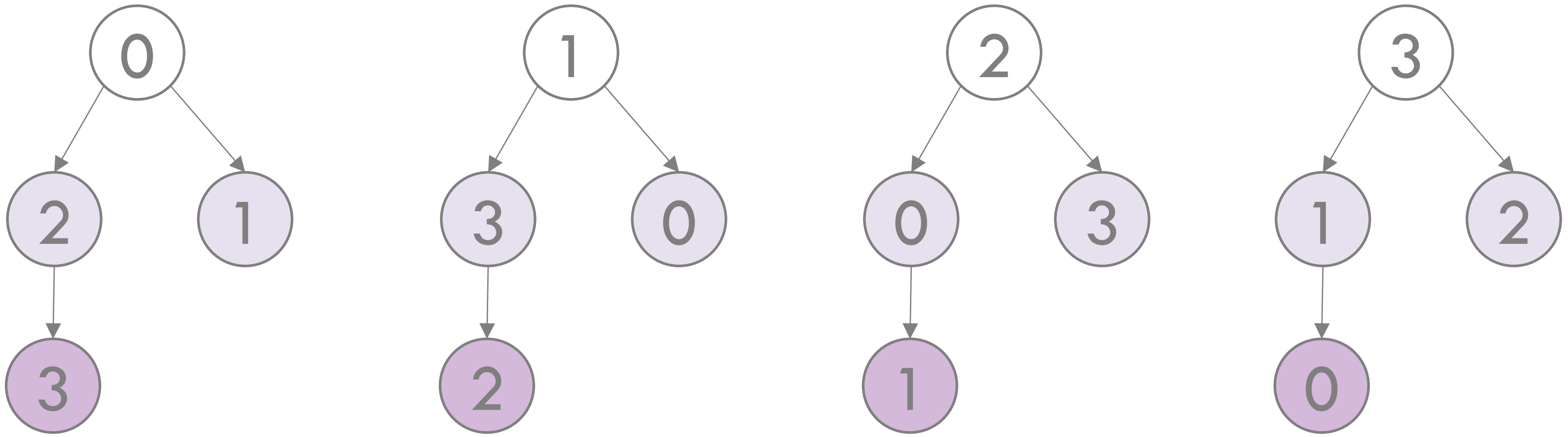
MultiTree All-Reduce: Reduce-Scatter



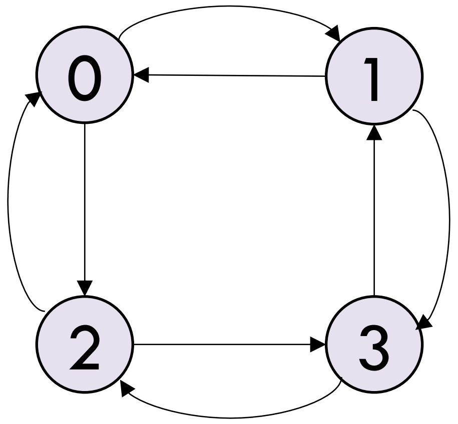
Reduce-Scatter (reduction from leaf level to root)



All-Gather (broadcast from root to leaf level)

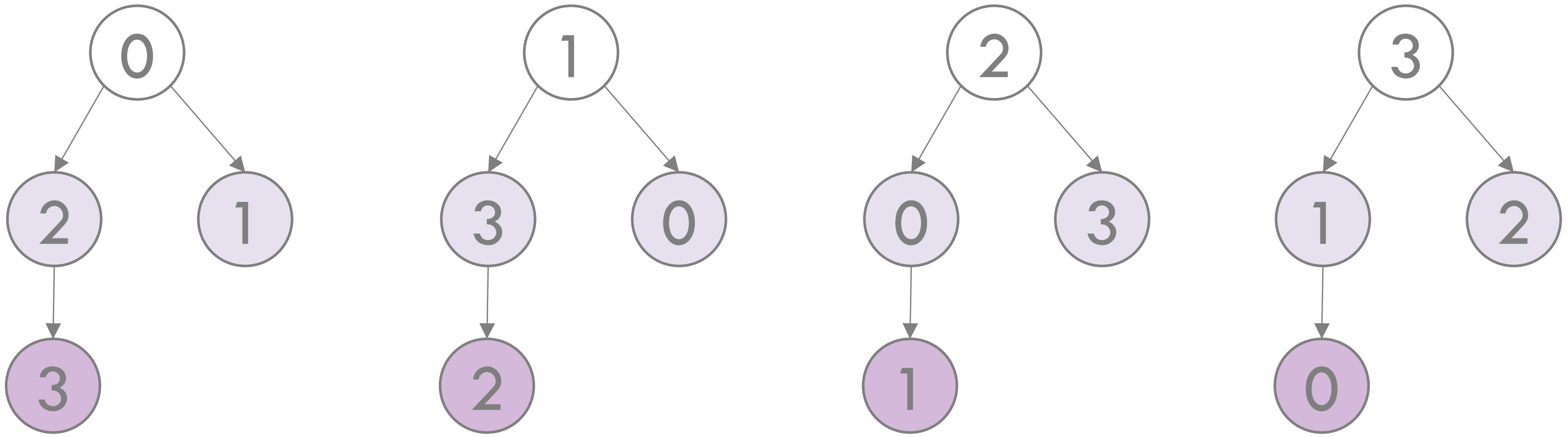


MultiTree All-Reduce: Reduce-Scatter

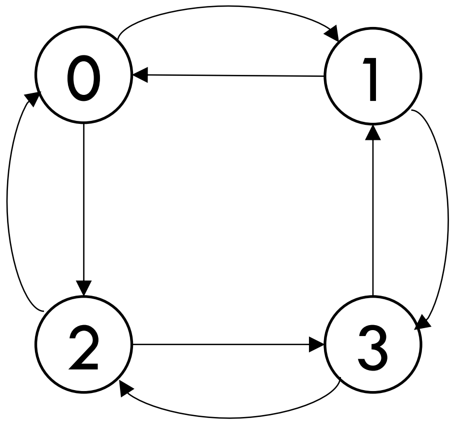


Reduce-Scatter (reduction from leaf level to root)

All-Gather (broadcast from root to leaf level)

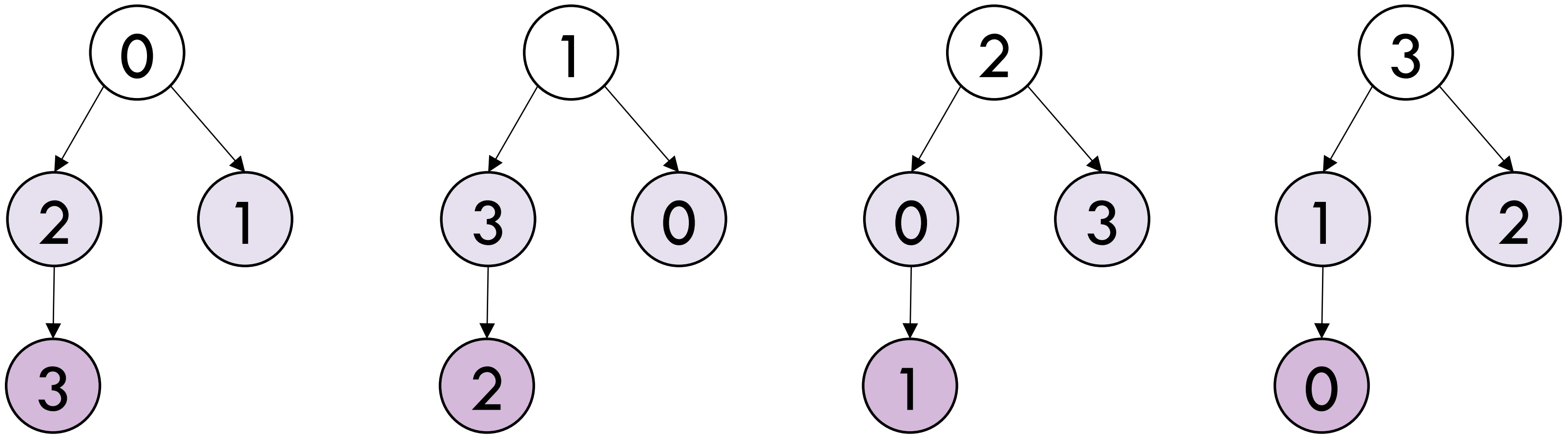


MultiTree All-Reduce: All-Gather

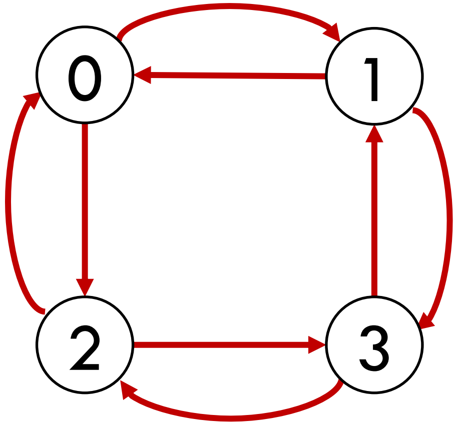


Reduce-Scatter (reduction from leaf level to root)

All-Gather (broadcast from root to leaf level)

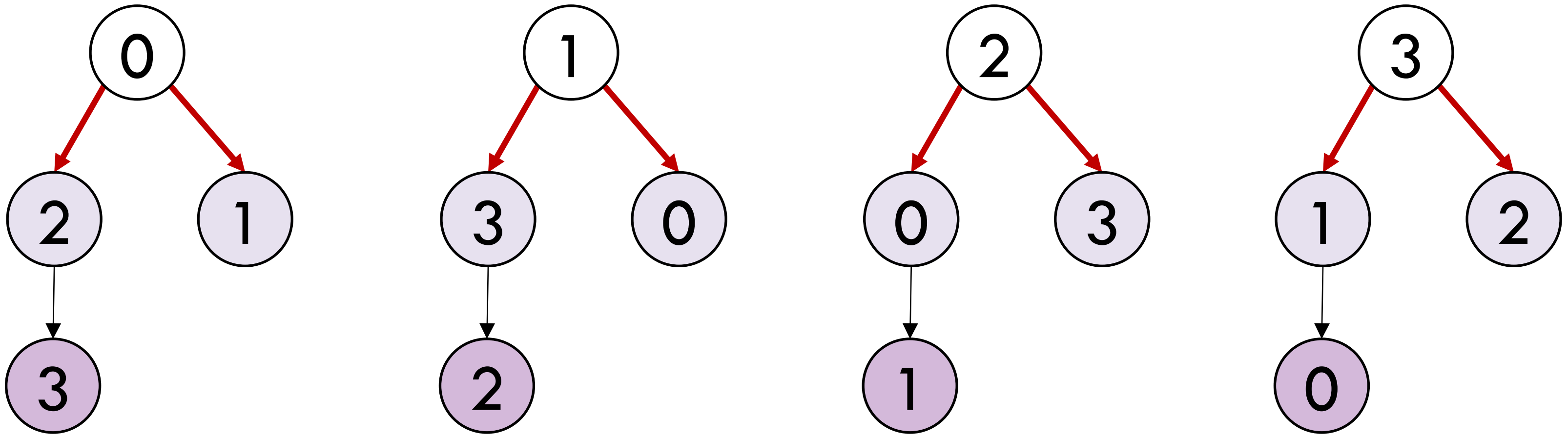


MultiTree All-Reduce: All-Gather

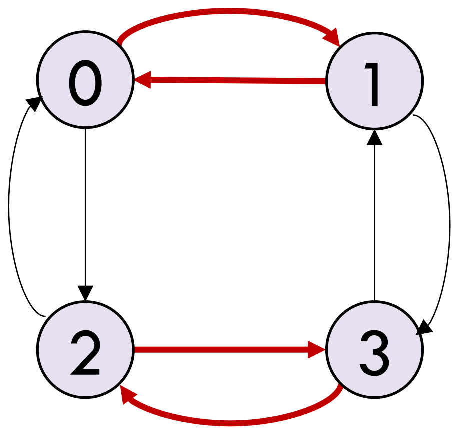


Reduce-Scatter (reduction from leaf level to root)

All-Gather (broadcast from root to leaf level)

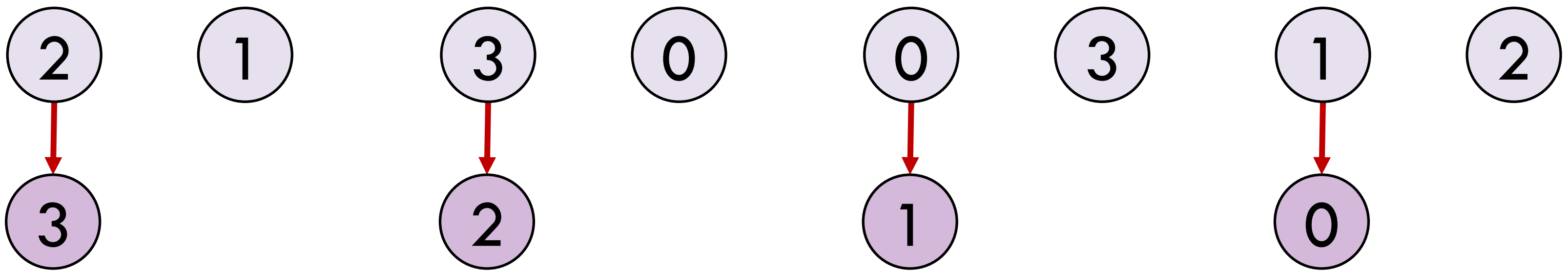


MultiTree All-Reduce: All-Gather



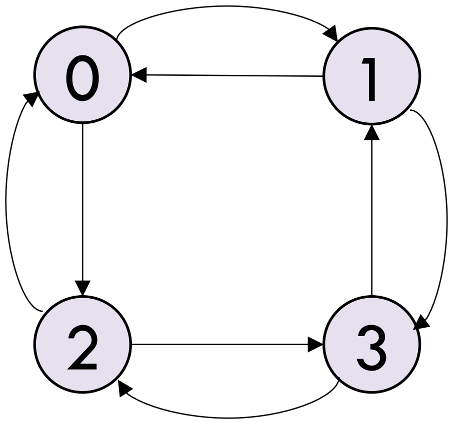
Reduce-Scatter (reduction from leaf level to root)

All-Gather (broadcast from root to leaf level)



MultiTree All-Reduce: All-Gather

Reduce-Scatter (reduction from leaf level to root)



0 1 2 3

All-Gather (broadcast from root to leaf level)

3 2 1 0

Hardware-based All-Reduce Scheduling and Example

- Message Command (Instruction): stored in an all-reduce schedule table entry

Op	FlowID	Parent	Children	Step	Start Addr	Size

Hardware-based All-Reduce Scheduling and Example

- Message Command (Instruction): stored in an all-reduce schedule table entry

Op	FlowID	Parent	Children	Step	Start Addr	Size

  - Op: `Reduce, Gather, NOP`

Hardware-based All-Reduce Scheduling and Example

- Message Command (Instruction): stored in an all-reduce schedule table entry

Op	FlowID	Parent	Children	Step	Start Addr	Size

  - Op: `Reduce, Gather, NOP`
  - FlowID: the ID of the spanning tree root

Hardware-based All-Reduce Scheduling and Example

- Message Command (Instruction): stored in an all-reduce schedule table entry

Op	FlowID	Parent	Children	Step	Start Addr	Size

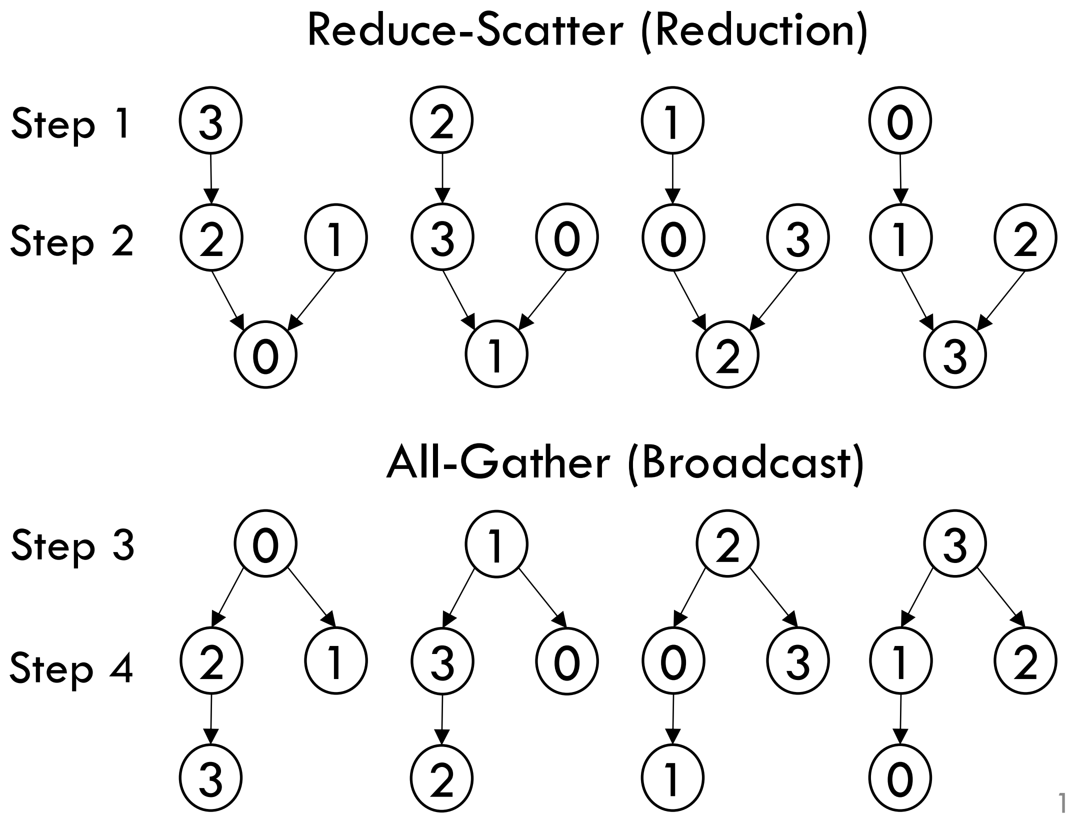
  - Op: `Reduce, Gather, NOP`
  - FlowID: the ID of the spanning tree root
  - Parent: for `Reduce` in tree reduction
  - Children: for `Gather` in tree broadcast

Hardware-based All-Reduce Scheduling and Example

- Message Command (Instruction): stored in an all-reduce schedule table entry

Op	FlowID	Parent	Children	Step	Start Addr	Size

  - Op: `Reduce, Gather, NOP`
  - FlowID: the ID of the spanning tree root
  - Parent: for `Reduce` in tree reduction
  - Children: for `Gather` in tree broadcast



Hardware-based All-Reduce Scheduling and Example

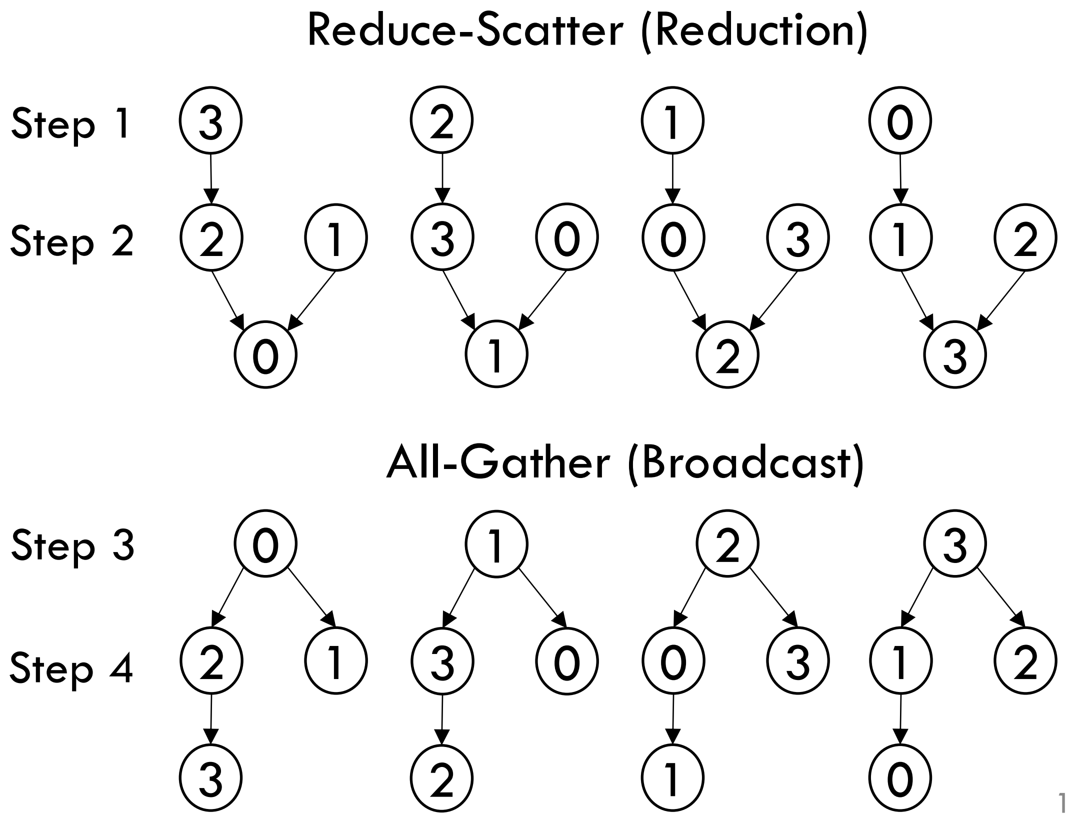
- Message Command (Instruction): stored in an all-reduce schedule table entry

Op	FlowID	Parent	Children	Step	Start Addr	Size

  - Op: `Reduce, Gather, NOP`
  - FlowID: the ID of the spanning tree root
  - Parent: for `Reduce` in tree reduction
  - Children: for `Gather` in tree broadcast

Accelerator 0

Op	FlowID	Parent	Children				Step
Reduce	3	1	nil	nil	nil	nil	1
Reduce	1	1	nil	nil	nil	nil	2
Reduce	2	2	1	nil	nil	nil	2
Gather	0	nil	1	2	nil	nil	3
Gather	2	2	1	nil	nil	nil	4



Hardware-based All-Reduce Scheduling and Example

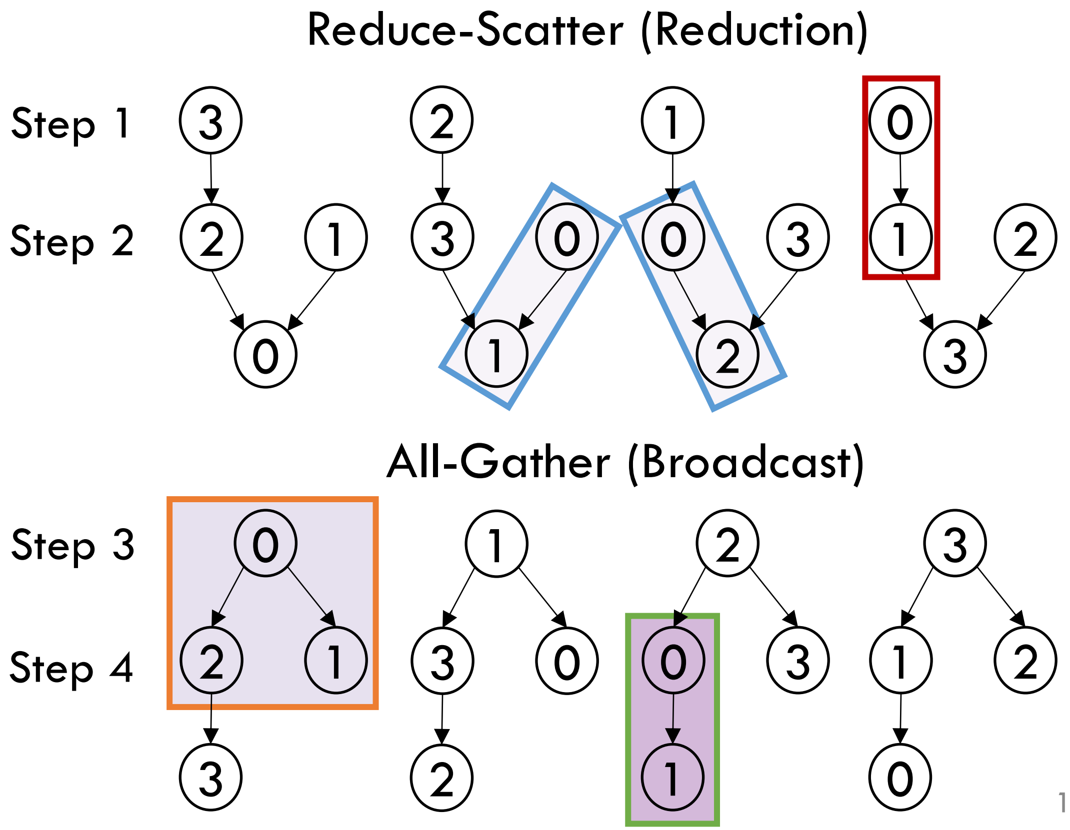
- Message Command (Instruction): stored in an all-reduce schedule table entry

Op	FlowID	Parent	Children	Step	Start Addr	Size

  - Op: `Reduce, Gather, NOP`
  - FlowID: the ID of the spanning tree root
  - Parent: for `Reduce` in tree reduction
  - Children: for `Gather` in tree broadcast

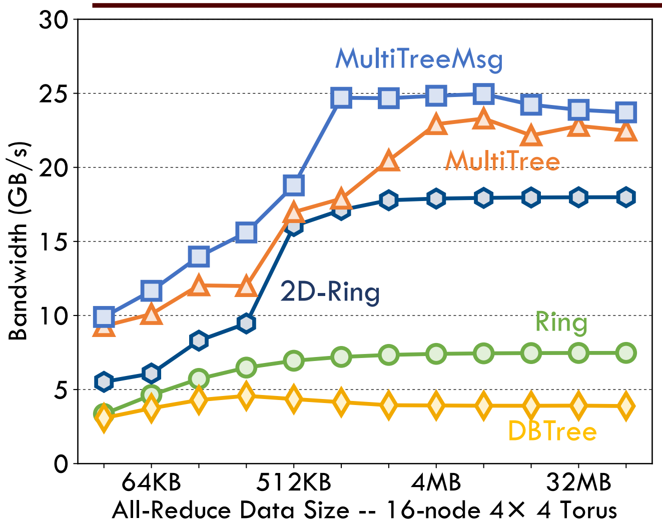
Accelerator 0

Op	FlowID	Parent	Children				Step
Reduce	3	1	nil	nil	nil	nil	1
Reduce	1	1	nil	nil	nil	nil	2
Reduce	2	2	1	nil	nil	nil	2
Gather	0	nil	1	2	nil	nil	3
Gather	2	2	1	nil	nil	nil	4

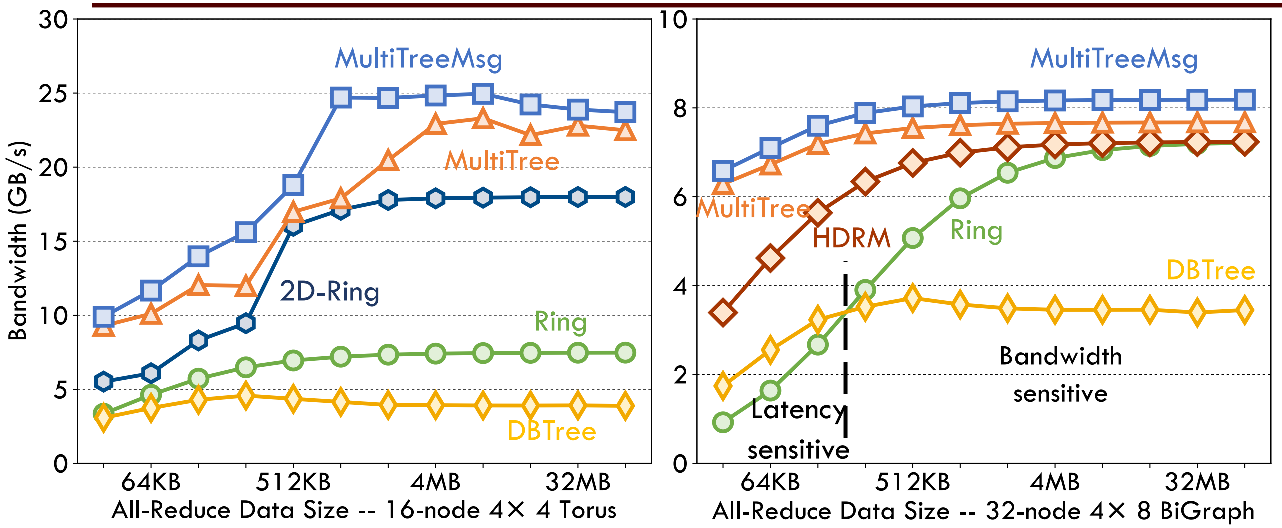


Evaluation – Bandwidth (top) and DNN Training Time (bottom)

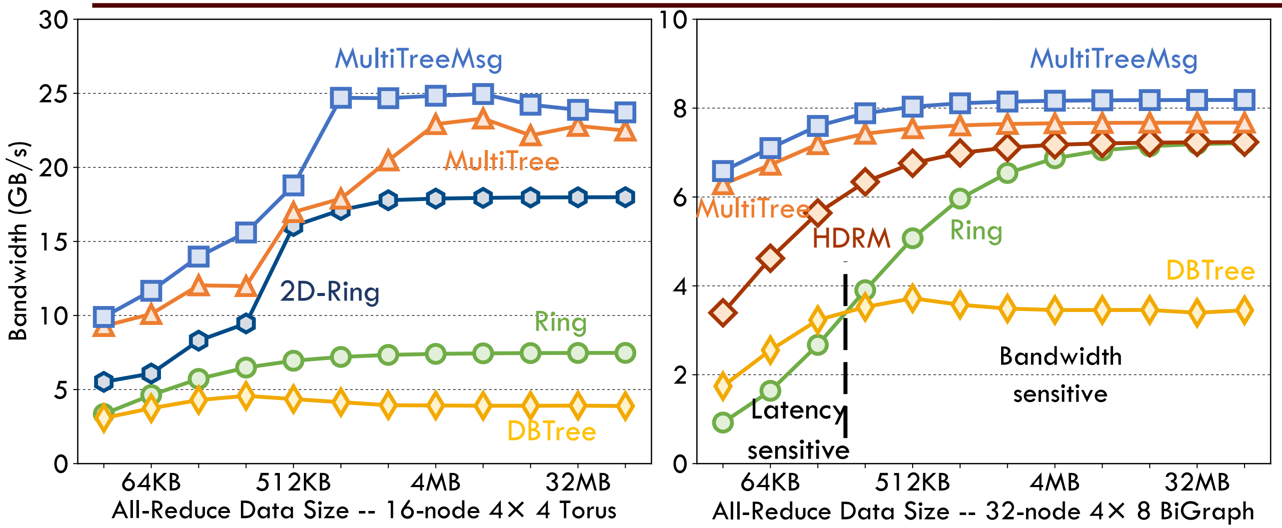
Evaluation – Bandwidth (top) and DNN Training Time (bottom)



Evaluation – Bandwidth (top) and DNN Training Time (bottom)

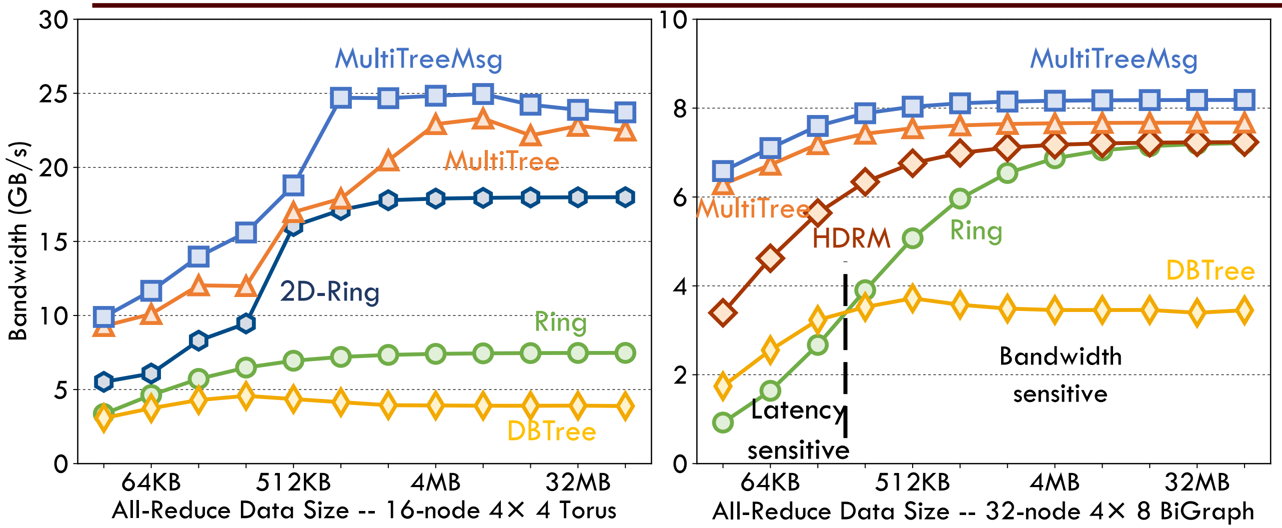


Evaluation – Bandwidth (top) and DNN Training Time (bottom)



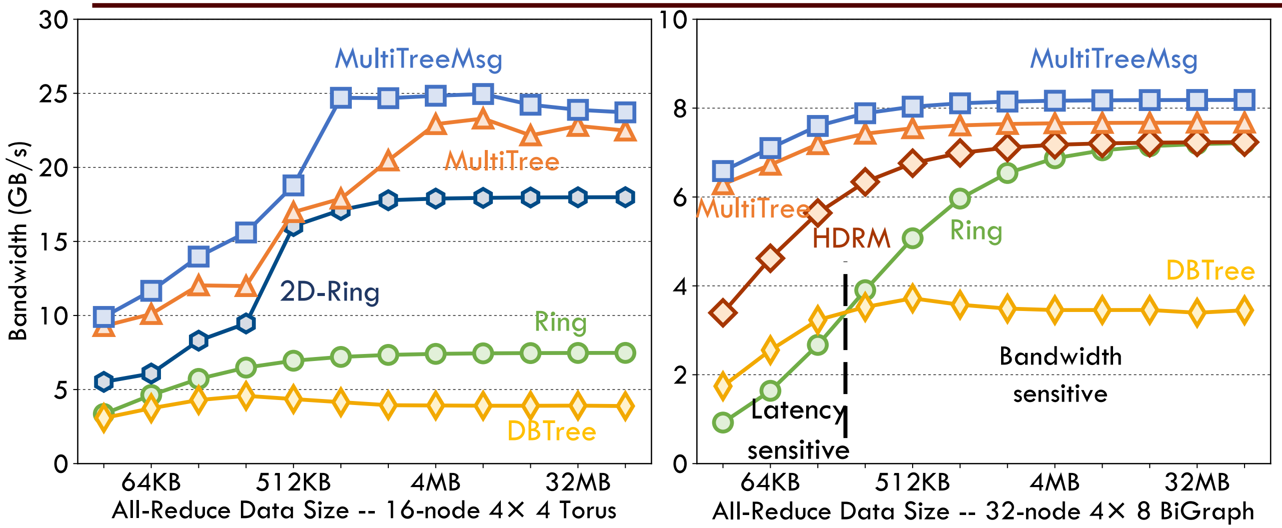
- BW in Torus and BiGraph
  - MultiTree achieves low latency and high BW

Evaluation – Bandwidth (top) and DNN Training Time (bottom)



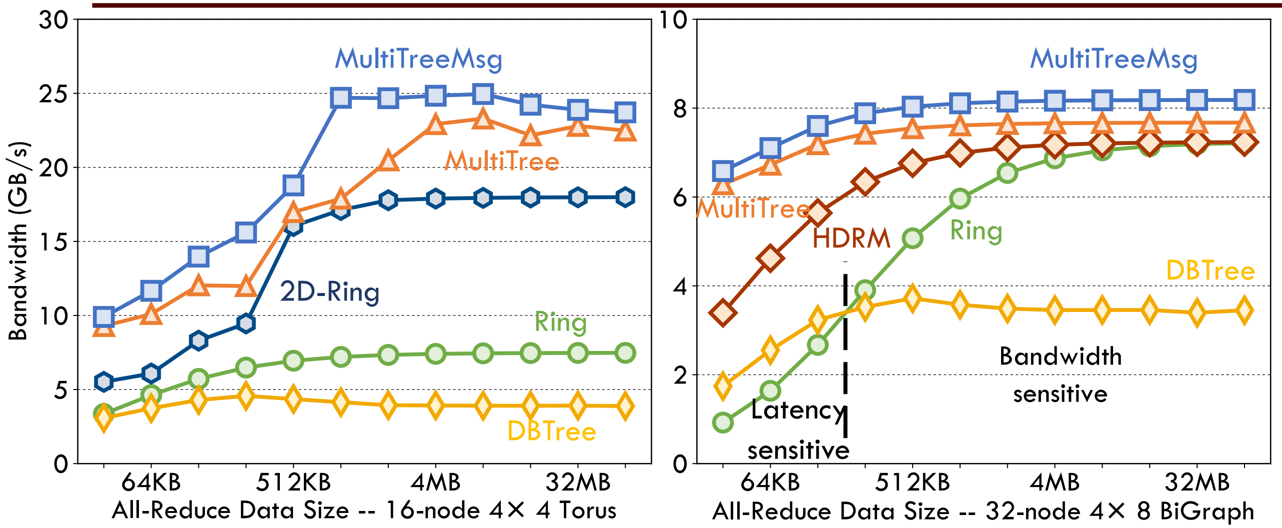
- BW in Torus and BiGraph
  - MultiTree achieves low latency and high BW
  - In Torus, 2D-Ring > Ring > DBTree

Evaluation – Bandwidth (top) and DNN Training Time (bottom)



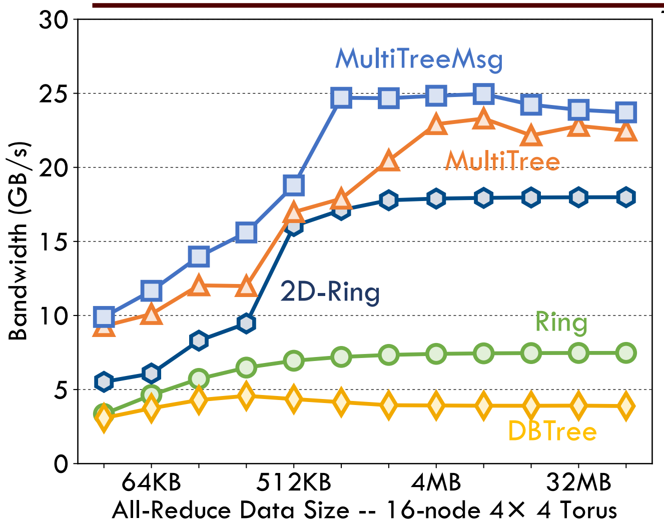
- BW in Torus and BiGraph
  - MultiTree achieves low latency and high BW
  - In Torus, 2D-Ring > Ring > DBTree
  - In Bigraph, HDRM good at latency and BW, but worse than MultiTree

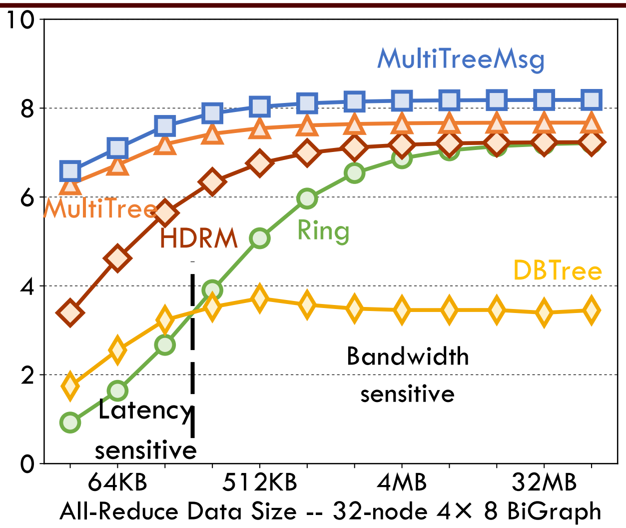
Evaluation – Bandwidth (top) and DNN Training Time (bottom)

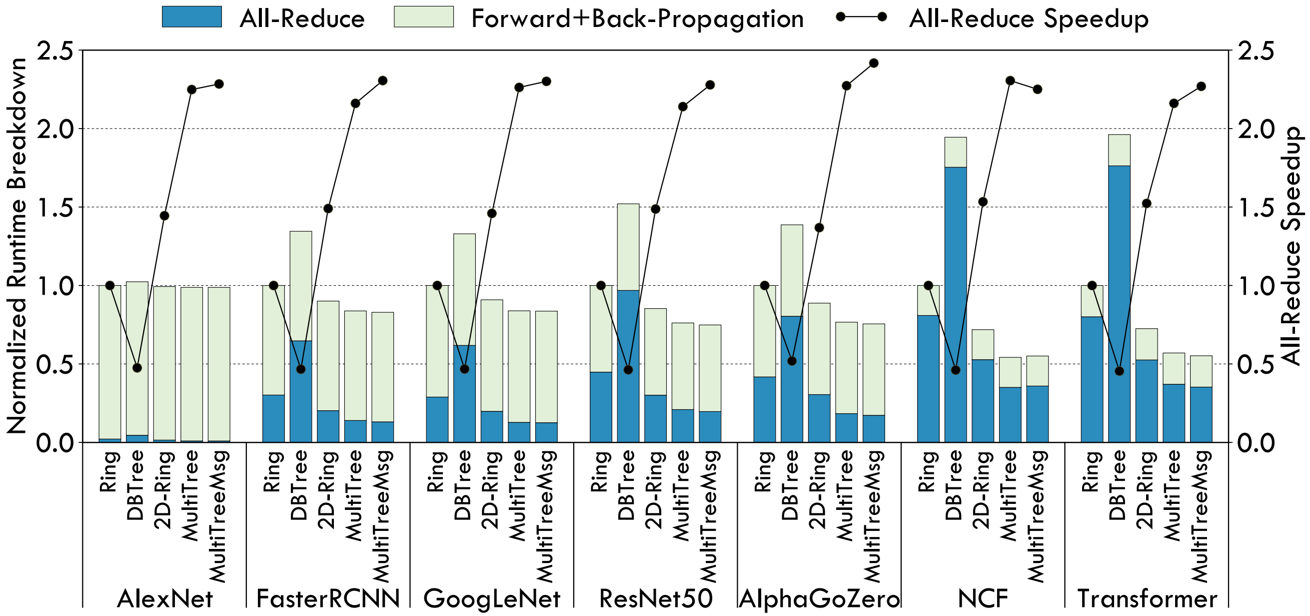


- BW in Torus and BiGraph
  - MultiTree achieves low latency and high BW
  - In Torus, 2D-Ring > Ring > DBTree
  - In Bigraph, HDRM good at latency and BW, but worse than MultiTree
  - Ring has good BW while DBTree has good latency in BiGraph

Evaluation – Bandwidth (top) and DNN Training Time (bottom)

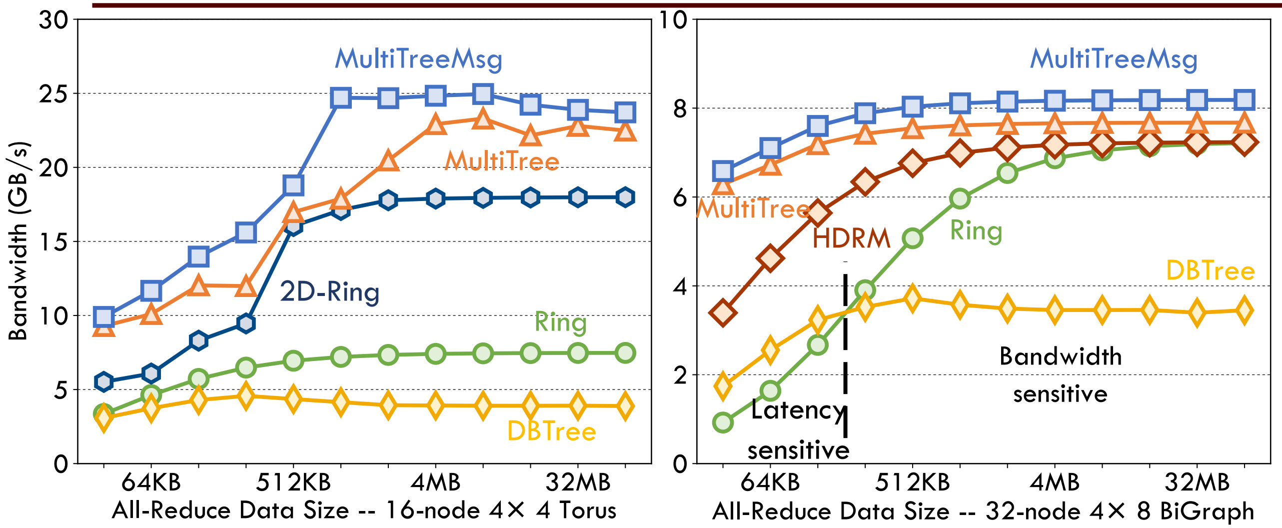


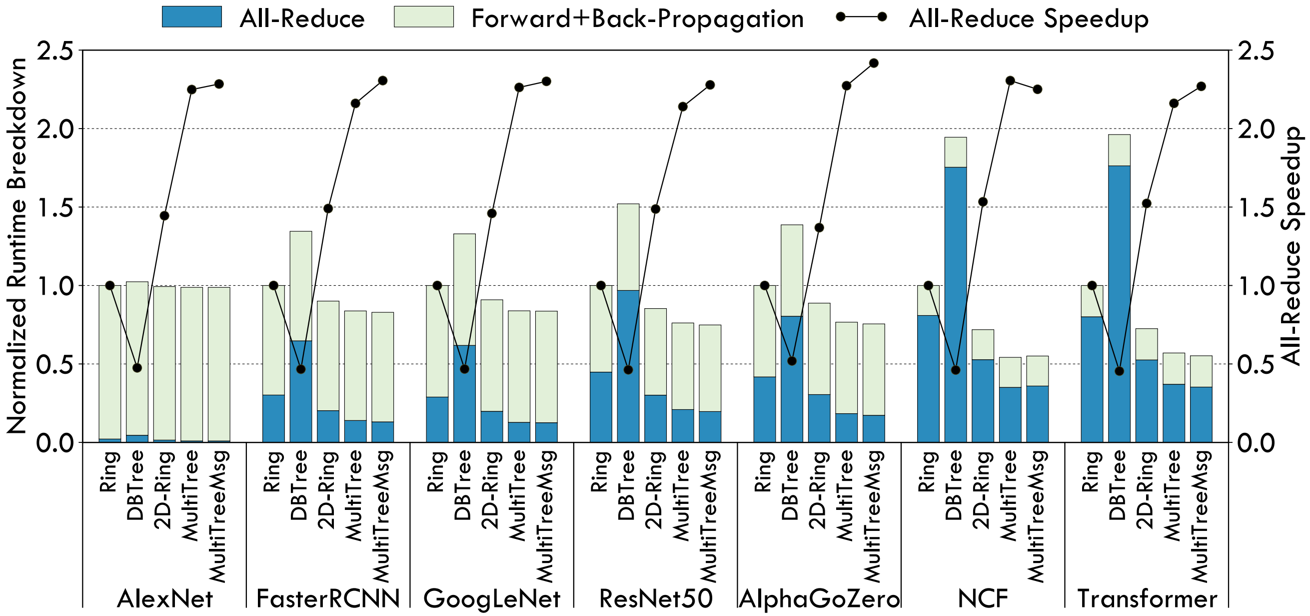




- BW in Torus and BiGraph
  - MultiTree achieves low latency and high BW
  - In Torus, 2D-Ring > Ring > DBTree
  - In Bigraph, HDRM good at latency and BW, but worse than MultiTree
  - Ring has good BW while DBTree has good latency in BiGraph
- DNN Training Time in 8x8 Torus

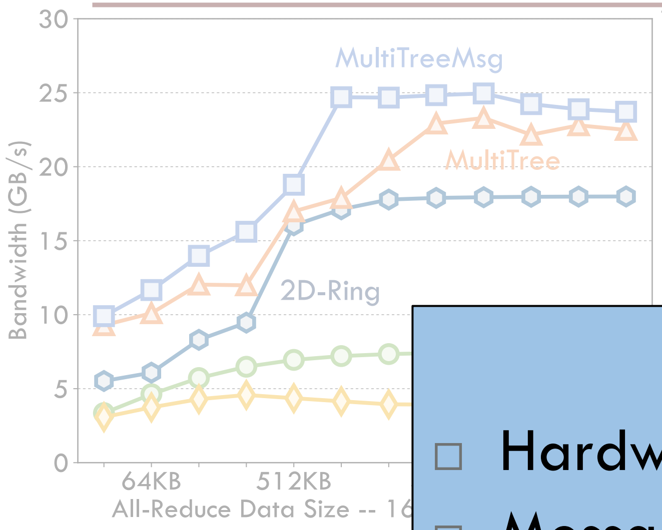
Evaluation – Bandwidth (top) and DNN Training Time (bottom)

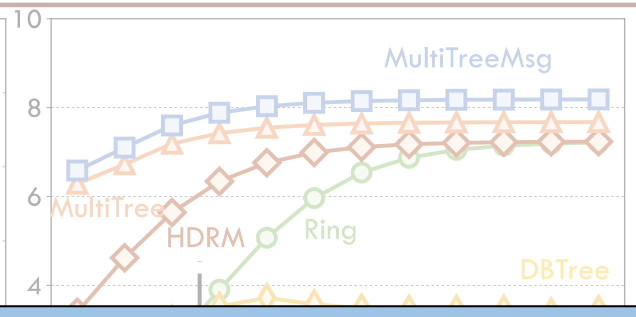


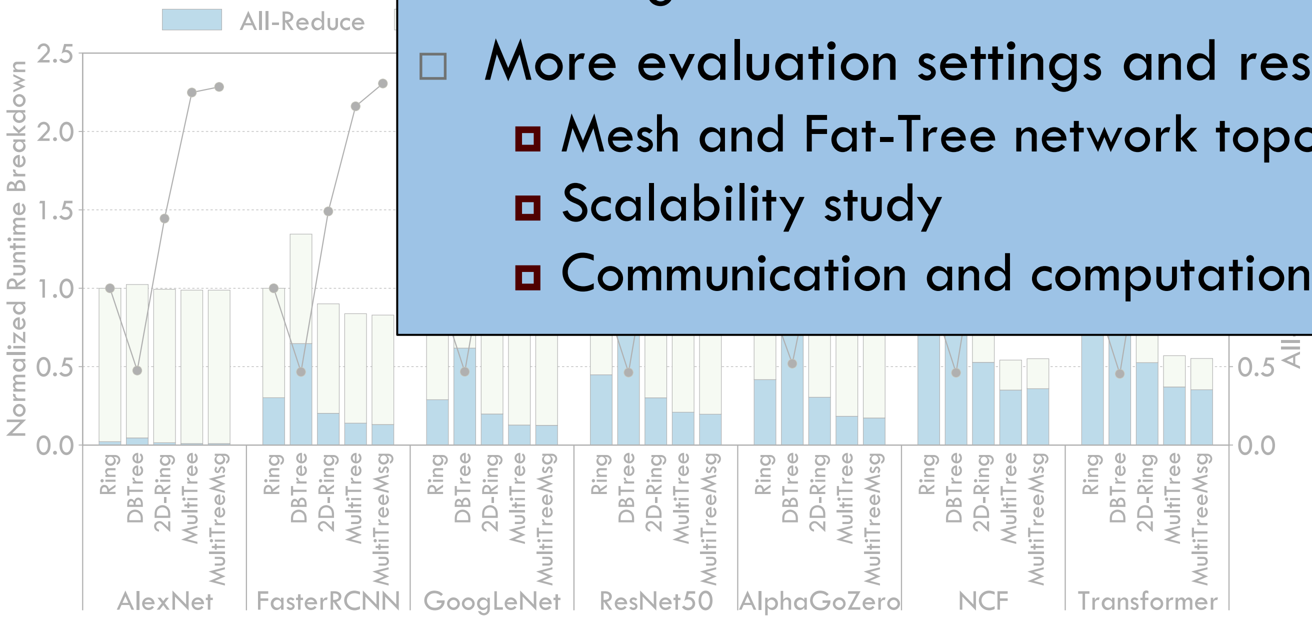


- BW in Torus and BiGraph
  - MultiTree achieves low latency and high BW
  - In Torus, 2D-Ring > Ring > DBTree
  - In Bigraph, HDRM good at latency and BW, but worse than MultiTree
  - Ring has good BW while DBTree has good latency in BiGraph
- DNN Training Time in 8x8 Torus
  - 2.3x and 1.56x *communication* speedup over Ring and 2D-Ring
  - Up to 81% and 31% training time reduction compared to Ring and 2D-Ring

Evaluation – Bandwidth (top) and DNN Training Time (bottom)







- BW in Torus and BiGraph
  - MultiTree achieves low latency and high BW
  - In Torus, 2D-Ring > Ring > DBTree
  - at latency
  - MultiTree
  - e DBTree has
  - n
- 3 Torus
  - *ication*
  - 2D-Ring
  - Up to 81% and 31% training time reduction compared to Ring and 2D-Ring

More in the paper

- Hardware-based scheduling control and datapath design
- Message-based flow control
- More evaluation settings and results
  - Mesh and Fat-Tree network topologies with different scales
  - Scalability study
  - Communication and computation overlap for DNN Training

Communication Algorithm-Architecture Co-Design for Distributed Deep Learning

Jiayi Huang Pritam Majumder Sungkeun Kim
Abdullah Muzahid Ki Hwan Yum EJ Kim

UC Santa Barbara (work done at TAMU) Texas A&M University