Collective Communnication Algorithms (Part-3)
CS 422 / CS 536

Vamsi Addanki

Purdue University

vaddank@purdue.edu

April 2, 2026

Overview

1. All-to-All (Transpose)
   - Hypercube
   - Ring
2. Mixture-of-Experts (MoE)

All-to-All (Transpose)

Setting:

- n nodes (ranks). Nodes can represent processes, GPUs, etc.
- Node i initially holds blocks: $i0,\ i1,\ \ldots,\ i(n-1)$
- Each block is of size $\frac{m}{n}$, where m is the total data per node.
- A block ij has source i (initial) and destination j (final)

Goal: Node j obtains $\{0j,\ 1j,\ \ldots,\ (n-1)j\}$

Node	**Blocks**
0	00 01 02 03 04 05 06 07
1	10 11 12 13 14 15 16 17
2	20 21 22 23 24 25 26 27
3	30 31 32 33 34 35 36 37
4	40 41 42 43 44 45 46 47
5	50 51 52 53 54 55 56 57
6	60 61 62 63 64 65 66 67
7	70 71 72 73 74 75 76 77

$\Longrightarrow$

Node	**Blocks**
0	00 10 20 30 40 50 60 70
1	01 11 21 31 41 51 61 71
2	02 12 22 32 42 52 62 72
3	03 13 23 33 43 53 63 73
4	04 14 24 34 44 54 64 74
5	05 15 25 35 45 55 65 75
6	06 16 26 36 46 56 66 76
7	07 17 27 37 47 57 67 77

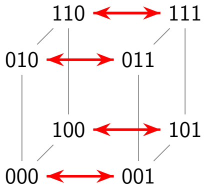
Hypercube All-to-All (Transpose)

Algorithm:

- Assume $n = 2^k$
- For $s = 0, 1, \ldots, k-1$:
  - Communication partner is the node whose bitstring matches exactly except the bit s
  - Node $i \leftrightarrow i \oplus 2^s$
  - Node i sends all blocks whose destination differs from i in bit s
  - After step s, the lowest $s+1$ bits of each block's destination are correct
- After $\log_2 n$ steps, every block reaches its final destination

Completion: $\log_2 n$ steps

Hypercube All-to-All: Step 0

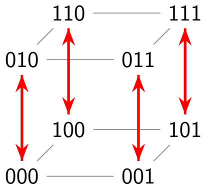


Distance: 1
Data: $\frac{m}{2}$
Cost: $\alpha + \beta \cdot \frac{m}{2}$

Node	Blocks
0	00 02 04 06 10 12 14 16
1	11 13 15 17 01 03 05 07
2	20 22 24 26 30 32 34 36
3	31 33 35 37 21 23 25 27
4	40 42 44 46 50 52 54 56
5	51 53 55 57 41 43 45 47
6	60 62 64 66 70 72 74 76
7	71 73 75 77 61 63 65 67

Blue = blocks that are sent and received during this step

Hypercube All-to-All: Step 1

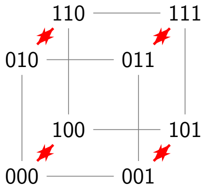


Distance: 2
Data: $\frac{m}{2}$
Cost: $\alpha + \beta \cdot \frac{m}{2}$

Node	Blocks
0	00 04 10 14 20 24 30 34
1	01 05 11 15 21 25 31 35
2	20 24 30 34 00 04 10 14
3	21 25 31 35 01 05 11 15
4	40 44 50 54 60 64 70 74
5	41 45 51 55 61 65 71 75
6	62 66 72 76 42 46 52 56
7	63 67 73 77 43 47 53 57

Blue = blocks that are sent and received during this step

Hypercube All-to-All: Step 2



Distance: 4
Data: $\frac{m}{2}$
Cost: $\alpha + \beta \cdot \frac{m}{2}$

Node	Blocks
0	00 10 20 30 40 50 60 70
1	01 11 21 31 41 51 61 71
2	02 12 22 32 42 52 62 72
3	03 13 23 33 43 53 63 73
4	44 54 64 74 04 14 24 34
5	45 55 65 75 05 15 25 35
6	46 56 66 76 06 16 26 36
7	47 57 67 77 07 17 27 37

Blue = blocks that are sent and received during this step

Hypercube All-to-All α-β cost

Total cost: $\log_2 n \cdot (\alpha + \beta \cdot \frac{m}{2})$

$$\sum_{s=0}^{\log_2 n - 1} \left(\alpha + \beta \cdot \frac{m}{2}\right) = \log_2 n \cdot \alpha + \frac{\log_2 n}{2} \beta m$$

- $\log_2 n$ communication steps (one per hypercube dimension)
- Each step exchanges **half of the data**
- Communication distance doubles each step (1, 2, 4, . . .).
  Note: This is not network topology distance. If the network is a hypercube, the distance remains 1-hop for all transmissions.

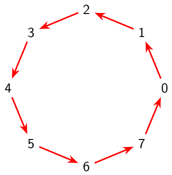
Ring All-to-All (Transpose)

Algorithm:

- Nodes are arranged in a logical ring.
- Node i keeps block (i, i) locally and sends all other blocks (i, j), $j \neq i$, to $(i + 1) \bmod n$.
- For $s = 0, 1, \dots, n - 2$:
  - Node i receives a bundle from $(i - 1) \bmod n$
  - It extracts the block in the bundle whose destination is i
  - It forwards all remaining blocks to $(i + 1) \bmod n$

Completion: $n - 1$ steps

Ring All-to-All: Step 0



Distance: 1
Data: **(n-1)** $\cdot \frac{m}{n}$
Cost: $\alpha + \beta \cdot (n-1)\frac{m}{n}$

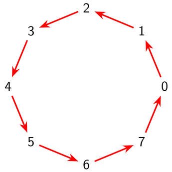
Node	Blocks
0	00 01 02 03 04 05 06 07
1	10 11 12 13 14 15 16 17
2	20 21 22 23 24 25 26 27
3	30 31 32 33 34 35 36 37
4	40 41 42 43 44 45 46 47
5	50 51 52 53 54 55 56 57
6	60 61 62 63 64 65 66 67
7	70 71 72 73 74 75 76 77

$\Longrightarrow$

Node	Blocks
0	00 71 72 73 74 75 76 70
1	01 11 02 03 04 05 06 07
2	10 12 22 13 14 15 16 17
3	20 21 23 33 24 25 26 27
4	30 31 32 34 44 35 36 37
5	40 41 42 43 45 55 46 47
6	50 51 52 53 54 56 66 57
7	60 61 62 63 64 65 67 77

Blue = blocks that reached their destination; Red = blocks still circulating

Ring All-to-All: Step 1



Distance: 1
Data: **(n-2)** $\cdot \frac{m}{n}$
Cost: $\alpha + \beta \cdot (n-2)\frac{m}{n}$

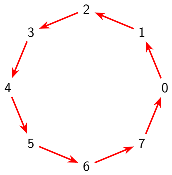
Node	Blocks
0	00 71 72 73 74 75 76 70
1	01 11 02 03 04 05 06 07
2	10 12 22 13 14 15 16 17
3	20 21 23 33 24 25 26 27
4	30 31 32 34 44 35 36 37
5	40 41 42 43 45 55 46 47
6	50 51 52 53 54 56 66 57
7	60 61 62 63 64 65 67 77

$\Longrightarrow$

Node	Blocks
0	00 61 62 63 64 65 60 70
1	01 11 72 73 74 75 76 71
2	02 12 22 03 04 05 06 07
3	10 13 23 33 14 15 16 17
4	20 21 24 34 44 25 26 27
5	30 31 32 35 45 55 36 37
6	40 41 42 43 46 56 66 47
7	50 51 52 53 54 57 67 77

Each block moves one hop; newly arrived blocks become local (blue)

Ring All-to-All: Step 2



Distance: 1
Data: **(n-3)** $\cdot \frac{m}{n}$
Cost: $\alpha + \beta \cdot (n-3)\frac{m}{n}$

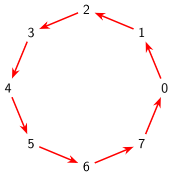
Node	Blocks
0	00 61 62 63 64 65 60 70
1	01 11 72 73 74 75 76 71
2	02 12 22 03 04 05 06 07
3	10 13 23 33 14 15 16 17
4	20 21 24 34 44 25 26 27
5	30 31 32 35 45 55 36 37
6	40 41 42 43 46 56 66 47
7	50 51 52 53 54 57 67 77

$\Longrightarrow$

Node	Blocks
0	00 51 52 53 54 50 60 70
1	01 11 62 63 64 65 61 71
2	02 12 22 73 74 75 76 72
3	03 13 23 33 04 05 06 07
4	10 14 24 34 44 15 16 17
5	20 21 25 35 45 55 26 27
6	30 31 32 36 46 56 66 37
7	40 41 42 43 47 57 67 77

Each block moves one hop; one more block reaches its destination

Ring All-to-All: Step 3



Distance: 1
Data: **(n-4)** $\cdot \frac{m}{n}$
Cost: $\alpha + \beta \cdot (n-4)\frac{m}{n}$

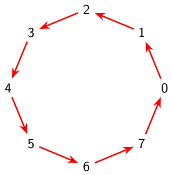
Node	Blocks
0	00 51 52 53 54 50 60 70
1	01 11 62 63 64 65 61 71
2	02 12 22 73 74 75 76 72
3	03 13 23 33 04 05 06 07
4	10 14 24 34 44 15 16 17
5	20 21 25 35 45 55 26 27
6	30 31 32 36 46 56 66 37
7	40 41 42 43 47 57 67 77

$\Longrightarrow$

Node	Blocks
0	00 41 42 43 40 50 60 70
1	01 11 52 53 54 51 61 71
2	02 12 22 63 64 65 62 72
3	03 13 23 33 74 75 76 73
4	04 14 24 34 44 05 06 07
5	10 15 25 35 45 55 16 17
6	20 21 26 36 46 56 66 27
7	30 31 32 37 47 57 67 77

Each block moves one hop; one more block reaches its destination

Ring All-to-All: Step 4



Distance: 1
Data: **(n-5)** $\cdot \frac{m}{n}$
Cost: $\alpha + \beta \cdot (n-5)\frac{m}{n}$

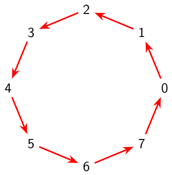
Node	Blocks
0	00 41 42 43 40 50 60 70
1	01 11 52 53 54 51 61 71
2	02 12 22 63 64 65 62 72
3	03 13 23 33 74 75 76 73
4	04 14 24 34 44 05 06 07
5	10 15 25 35 45 55 16 17
6	20 21 26 36 46 56 66 27
7	30 31 32 37 47 57 67 77

$\Longrightarrow$

Node	Blocks
0	00 31 32 30 40 50 60 70
1	01 11 42 43 41 51 61 71
2	02 12 22 53 54 52 62 72
3	03 13 23 33 64 65 63 73
4	04 14 24 34 44 75 76 74
5	05 15 25 35 45 55 06 07
6	10 16 26 36 46 56 66 17
7	20 21 27 37 47 57 67 77

Each block moves one hop; one more block reaches its destination

Ring All-to-All: Step 5



Distance: 1
Data: **(n-6)** $\cdot \frac{m}{n}$
Cost: $\alpha + \beta \cdot (n-6)\frac{m}{n}$

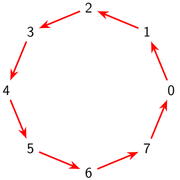
Node	Blocks
0	00 31 32 30 40 50 60 70
1	01 11 42 43 41 51 61 71
2	02 12 22 53 54 52 62 72
3	03 13 23 33 64 65 63 73
4	04 14 24 34 44 75 76 74
5	05 15 25 35 45 55 06 07
6	10 16 26 36 46 56 66 17
7	20 21 27 37 47 57 67 77

$\Longrightarrow$

Node	Blocks
0	00 21 20 30 40 50 60 70
1	01 11 32 31 41 51 61 71
2	02 12 22 43 42 52 62 72
3	03 13 23 33 54 53 63 73
4	04 14 24 34 44 65 64 74
5	05 15 25 35 45 55 76 75
6	06 16 26 36 46 56 66 07
7	10 17 27 37 47 57 67 77

Each block moves one hop; one more block reaches its destination

Ring All-to-All: Step 6



Distance: 1
Data: $1 \cdot \frac{m}{n}$
Cost: $\alpha + \beta \cdot 1 \cdot \frac{m}{n}$

Node	Blocks
0	00 21 20 30 40 50 60 70
1	01 11 32 31 41 51 61 71
2	02 12 22 43 42 52 62 72
3	03 13 23 33 54 53 63 73
4	04 14 24 34 44 65 64 74
5	05 15 25 35 45 55 76 75
6	06 16 26 36 46 56 66 07
7	10 17 27 37 47 57 67 77

$\Longrightarrow$

Node	Blocks
0	00 10 20 30 40 50 60 70
1	01 11 21 31 41 51 61 71
2	02 12 22 32 42 52 62 72
3	03 13 23 33 43 53 63 73
4	04 14 24 34 44 54 64 74
5	05 15 25 35 45 55 65 75
6	06 16 26 36 46 56 66 76
7	07 17 27 37 47 57 67 77

All blocks have reached their destination (complete transpose)

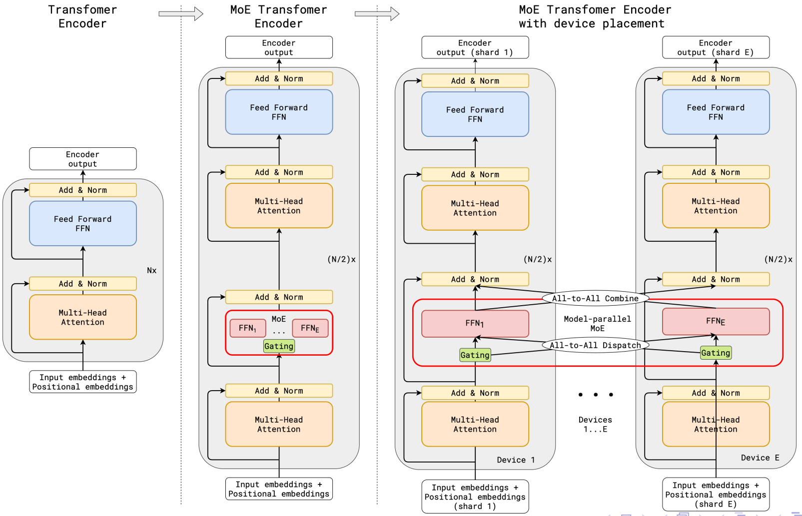
Ring All-to-All α-β cost

Total cost: $(n-1)\alpha + \beta \cdot \frac{n-1}{2} m$

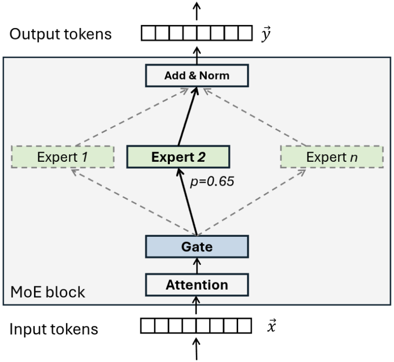
$$\sum_{s=0}^{n-2} \left(\alpha + \beta \cdot (n-1-s)\frac{m}{n} \right) = (n-1)\alpha + \beta \cdot \frac{n-1}{2} m$$

- $n-1$ communication steps
- Step s sends $(n-1-s)\frac{m}{n}$ data
- Communication distance remains 1 in the logical ring

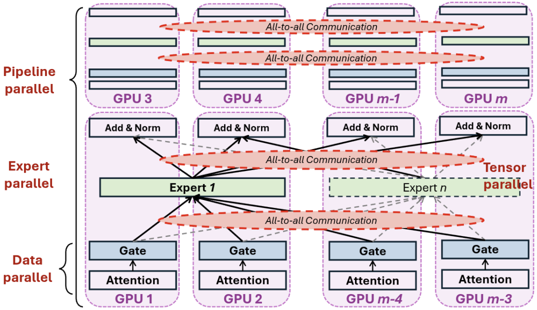
Mixture-of-Experts (MoE)



Mixture-of-Experts (MoE)



Mixture-of-Experts (MoE)



Mixture-of-Experts (MoE) I

Parallelism:

- A model architecture with many specialized feed-forward networks (experts).
- A **gating function** selects a small subset of experts per input at each layer.
- Only selected experts are activated $\Rightarrow$ **sparse computation**.

Scaling:

- Scaling dense models increases compute cost linearly with parameters.
- Training/inference becomes memory- and bandwidth-bound.
- MoE enables:
  - Massive parameter count (capacity)
  - Constant compute per token (efficiency) while increasing the size of the model

Mixture-of-Experts (MoE) II

Key idea:

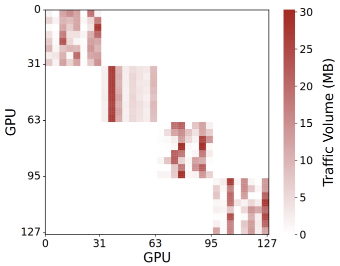
$$\text{Output}(x) = \sum_{e \in \text{Top-}k(x)} g_e(x) \cdot f_e(x)$$

Implications:

- Enables scaling to trillions of parameters without significantly increasing compute cost per token.
- Introduces new systems challenges:
  - Load balancing across experts
  - Network communication (token routing)
  - Stragglers and contention

Mixture-of-Experts (MoE)

- All to All communication across GPUs in MoE is rarely a simple transpose pattern
- Key distinction: Transpose is *uniform* all to all, while MoE is *non-uniform* all to all; sometimes referred and implemented as alltoall**v**
- Uniform all to all is still relevant e.g., distributed fourier transforms in scientific computing.



Mixture-of-Experts (MoE)

- All to all is typically implemented as point-to-point communications, where each message corresponds to a batch of tokens being sent from one GPU to another.
- Specialized kernels:
  - pplx
  - DeepEP
  - flashinfer
  - nixl-ep
  - ...

Mixture-of-Experts (MoE)

- Given a system architecture, network topology, each link's bandwidth, what is the optimal schedule for all to all communication in MoE?

Mixture-of-Experts (MoE)

- Given a *demand matrix* representing GPU to GPU communication required by all to all in MoE, the optimal schedule corresponds to the solution to a **maximum concurrent flow** problem.

Mixture-of-Experts (MoE)

- Observation: The placement of experts across GPUs, mutates the demand matrix; a good placement can reduce communication demand and improve performance.

Further Reading I

[1] Basu P, Zhao L, Fantl J, Pal S, Krishnamurthy A, Khoury J. Efficient all-to-all collective communication schedules for direct-connect topologies. InProceedings of the 33rd International Symposium on High-Performance Parallel and Distributed Computing 2024 Jun 3 (pp. 28-41).

[2] Liao X, Sun Y, Tian H, Wan X, Jin Y, Wang Z, Ren Z, Huang X, Li W, Tse KF, Zhong Z. Mixnet: A runtime reconfigurable optical-electrical fabric for distributed mixture-of-experts training. InProceedings of the ACM SIGCOMM 2025 Conference 2025 Sep 8 (pp. 554-574).

[3] Lepikhin D, Lee H, Xu Y, Chen D, Firat O, Huang Y, Krikun M, Shazeer N, Chen Z. Gshard: Scaling giant models with conditional computation and automatic sharding. arXiv preprint arXiv:2006.16668. 2020 Jun 30.

[4] Zhou Y, Lei T, Liu H, Du N, Huang Y, Zhao V, Dai AM, Le QV, Laudon J. Mixture-of-experts with expert choice routing. Advances in Neural Information Processing Systems. 2022 Dec 6;35:7103-14.

[5] The vLLM MoE Playbook: A Practical Guide to TP, DP, PP and Expert Parallelism: `https://rocm.blogs.amd.com/software-tools-optimization/vllm-moe-guide/README.html`

Further Reading II

[6] Zhou A, Addanki V, Apostolaki M. To Reconfigure or Not to Reconfigure: Optimizing All-to-All Collectives in Circuit-Switched Photonic Interconnects. arXiv preprint arXiv:2602.10468. 2026 Feb 11.

[7] Grama A. An Introduction to Parallel Computing: Design and Analysis of Algorithms, 2/e. Pearson Education India; 2008.

The End