DSC-204A: Scalable Data Systems, Winter 2024

13: Collective Communication 2

Lecturer: Hao Zhang *Scribe: Longxuan Yu, Tushar Mohan, Zihe Liu*

1 Course Topic

- Collective communication
- Connection between distributed SGD and collective comm
- Communication Model: $\alpha + n\beta, \quad \beta = \frac{1}{B}$
- Small Message size ($n \to 0$): α dominates, emphasize latency
- Large Message Size ($n \to +\infty$): $n\beta$ dominate, emphasize bandwidth utilization

2 Pros and Cons of MST algorithms

- Emphasize low latency
  - MST-based algorithm is latency-optimal
  - How to prove? (Taking broadcast as an example)
- Problem of Minimum Spanning Tree Algorithm?
  - Some links are idle

2.1 Example of Broadcasts

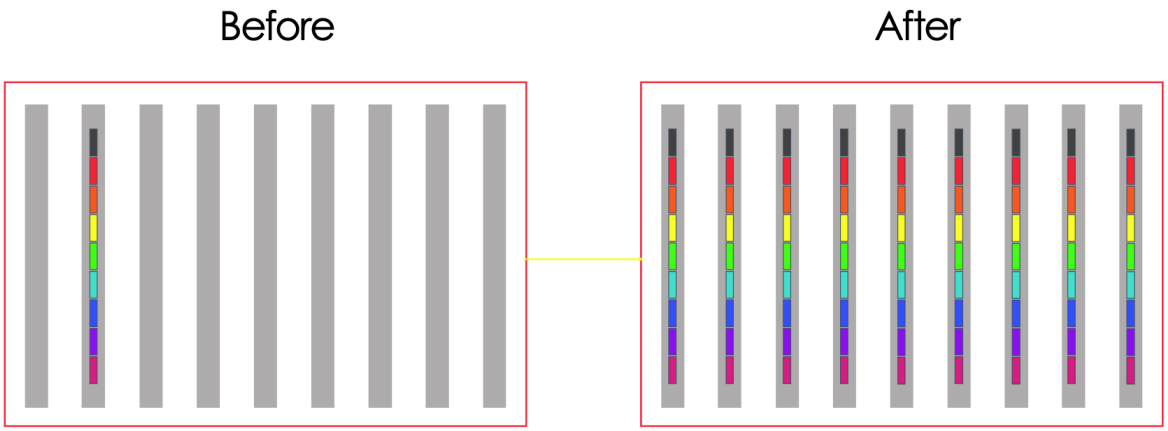

Figure 1: Example figure 1

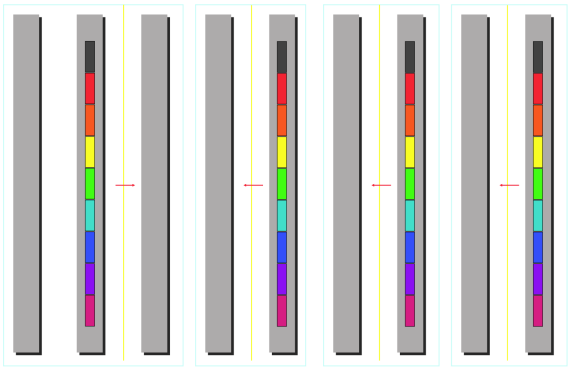

Figure 2: Braodcasts progress

Large Message

Communication Model:

$$\alpha + n\beta, \quad \beta = \frac{1}{B} \tag{1}$$

- The second term dominates – we want to minimize the second term.
- We want to utilize the bandwidth as much as possible.

Long Vector Building Blocks

- We will show how the following building blocks:

  - collect/distributed combine
  - scatter/gather

  can be implemented using "bucket" algorithms while attaining minimal cost due to the length of vectors.

- Implementation for arbitrary numbers of nodes.
- No network conflicts.
- **NOTICE:** scatter and gather already satisfy these conditions.

General Principles

- Use all the links between every two nodes.
- A logical ring can be embedded in a physical linear array with worm-hole routing, since the "wrap-around" message doesn't conflict.
- Ring Algorithm has 2 main advantages:

  1. Full utilization of bandwidth
  2. Any number of node can be implemented

3 Collective Communication (with Ring Algo)

3.1 Allgather

When a fraction of the message is in each node and we need all the nodes to have the whole message :

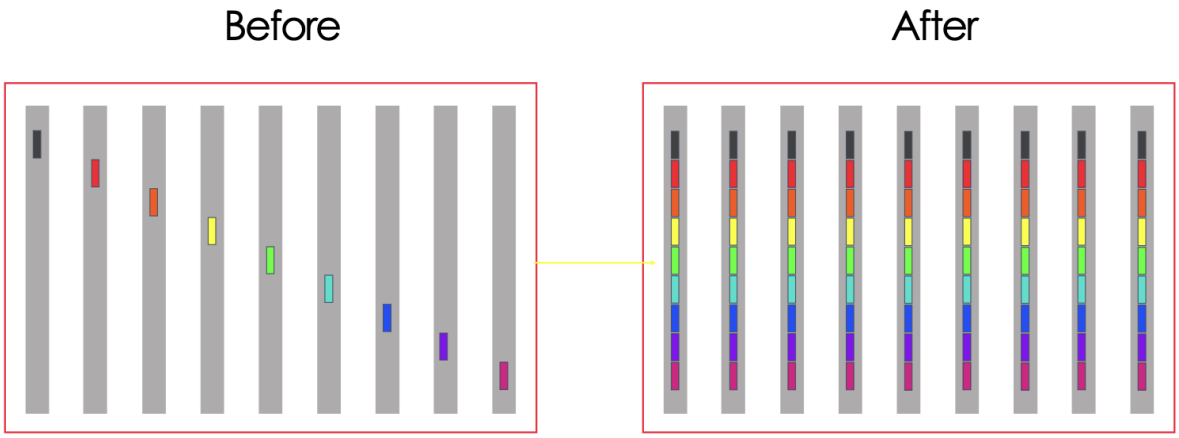

Figure 3: AllGather

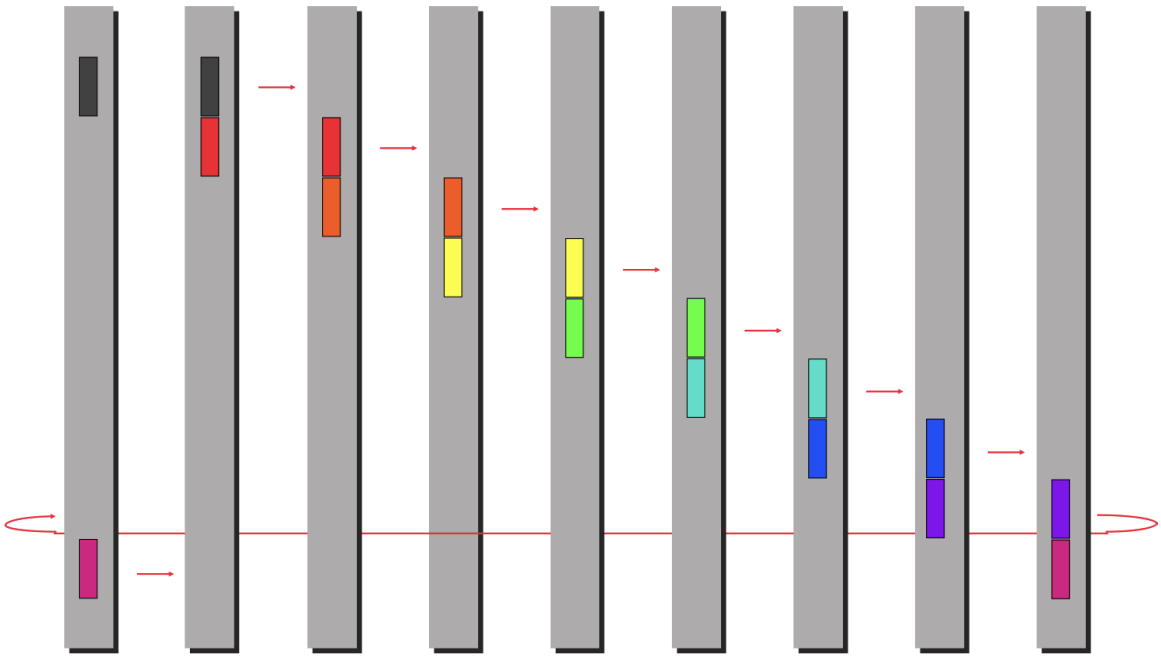

Figure 4: Step 2 in the Allgather communication

Steps:

1. We start by connecting all the subsequent nodes and the last node feeding back to the first one.
2. Then by taking a partial sum between the current states of the corresponding nodes, the missing data fraction is fed from one node to another.
3. Step 2 is repeated till all the nodes have the whole message.

There is a cost to everything, so does Allgather method:

$$(p-1)(\alpha + \frac{n}{p}\beta)$$

where, p is the number of nodes $\Rightarrow$ (p-1) is the number of steps, $(\alpha + \frac{n}{p}\beta)$ is the cost per step. Rest all symbols have their standard meanings.

3.2 Reduce-Scatter

This methods involves every node having the full message and ending with each node with just a fraction of the message. This is a combination of 2 fundamental algorithms - Reduce and Scatter.

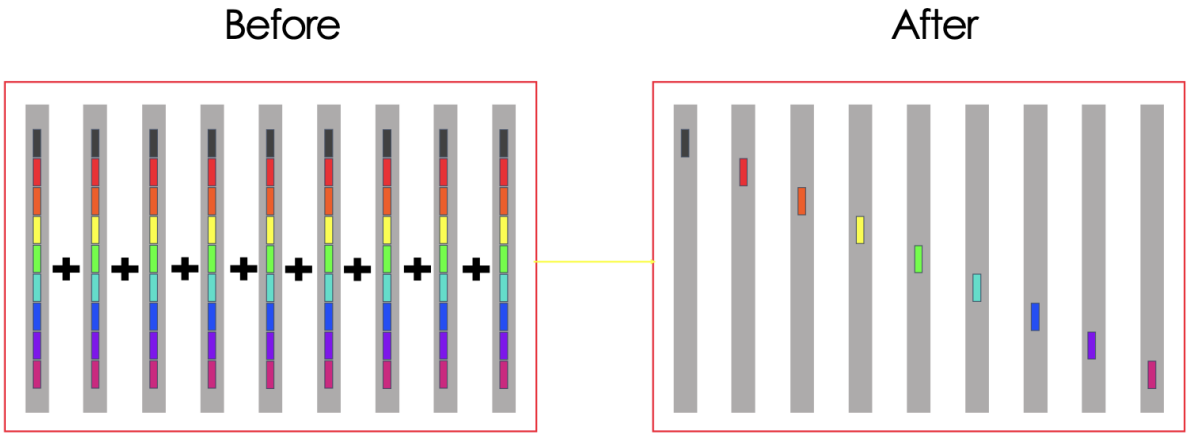

Figure 5: Reduce-Scatter

Steps:

1. We start by constructing a ring with the nodes.
2. Propagate 1 message and take a partial sum.
3. Continue the step 2 till all the workers have the same sum.

The cost for this is:

$$(p-1)(\alpha + \frac{n}{p}\beta + \frac{n}{p}\gamma)$$

where p-1 is the number of steps and $(\alpha + \frac{n}{p}\beta + \frac{n}{p}\gamma)$ is the cost per step.

3.3 Scatter

This algorithm includes take one message from a worker and distribute a fraction to each worker. It is implemented in MST, can ring do better? The answer is no. All the other workers do not have the message and they need a fraction, so MST will be the most optimal both in bandwidth and latency.

3.4 Broadcast (Long Vector)

This algorithm consists of 2 algorithms - first scatter (with the ring algorithm) and then allgather. It starts with one worker with the whole message, and ending with all the workers with all the message.

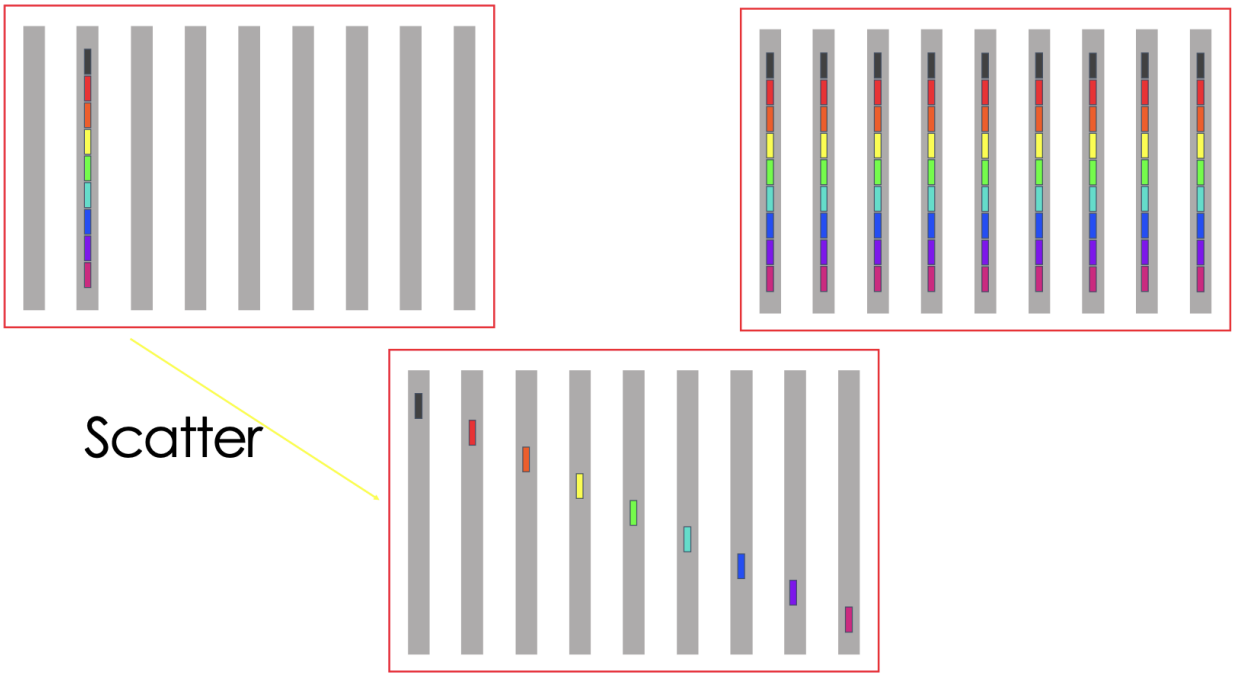

Figure 6: Step 1

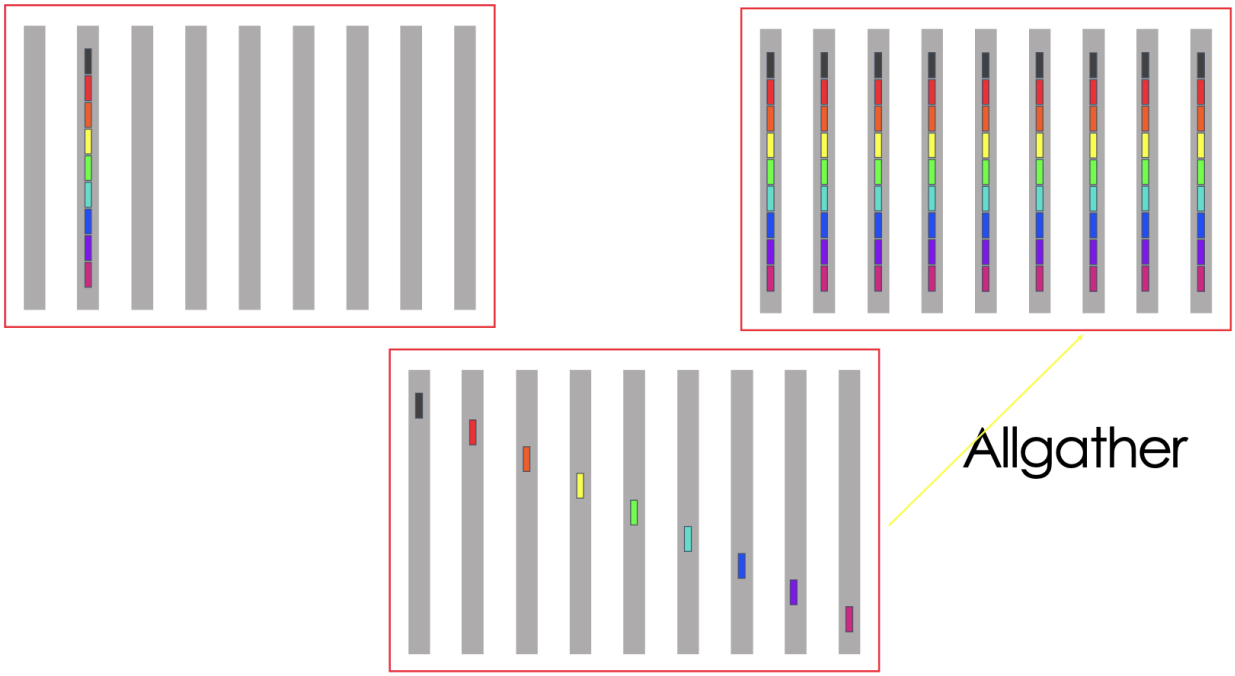

Figure 7: Step 2

Cost for this is the sum of costs of scatter and allgather algorithms by definition:

$$\begin{aligned} & log(p)\alpha + \frac{p-1}{p}n\beta \\ & \quad + \\ & (p-1)\alpha + \frac{p-1}{p}n\beta \\ & \quad = (log(p) + p - 1)\alpha + 2\frac{p-1}{p}n\beta \end{aligned}$$

On the other hand MST broadcast has a cost $= log(p)(\alpha + n\beta)$. The multiplier term for ring algo will become larger. We are doing better in terms of bandwidth as $2\frac{p-1}{p}$ is less as compared to $log(p)$. But ring algo takes a hit in latency.

3.5 Reduce (-to-one)

This is an amalgamation of 2 algorithms – Reduce-scatter followed by Gather. First, every worker gets a fraction of the sum and then 1 workers gets the whole message.

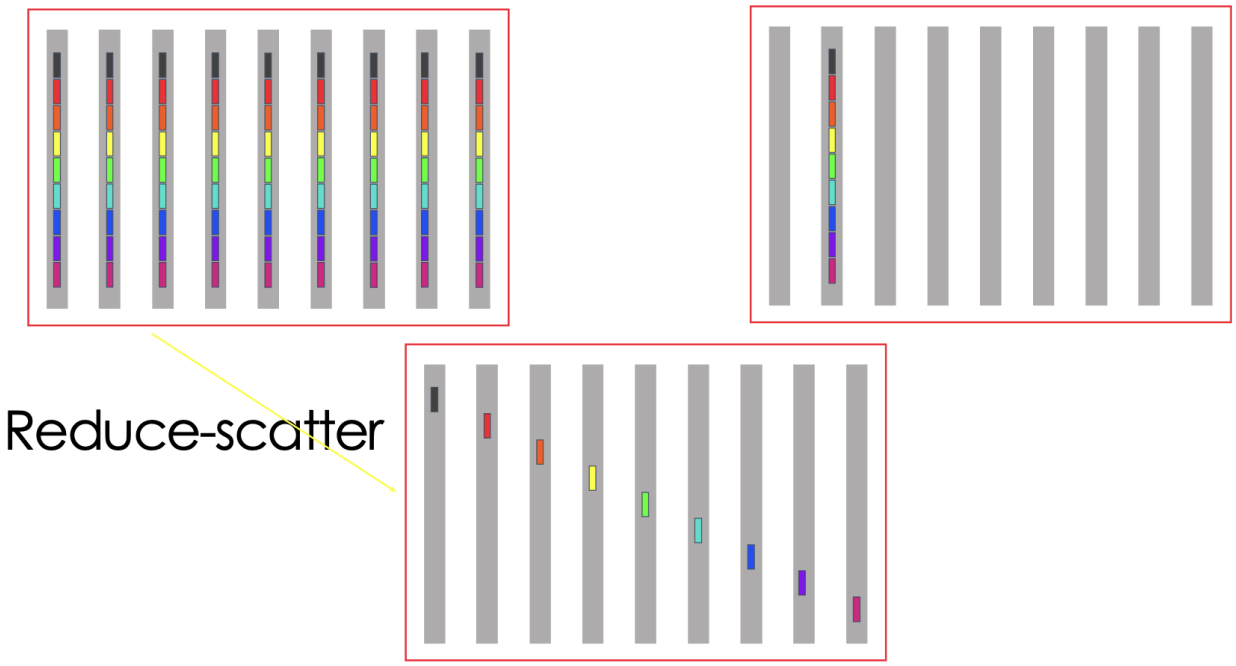

Figure 8: Reduce Step 1

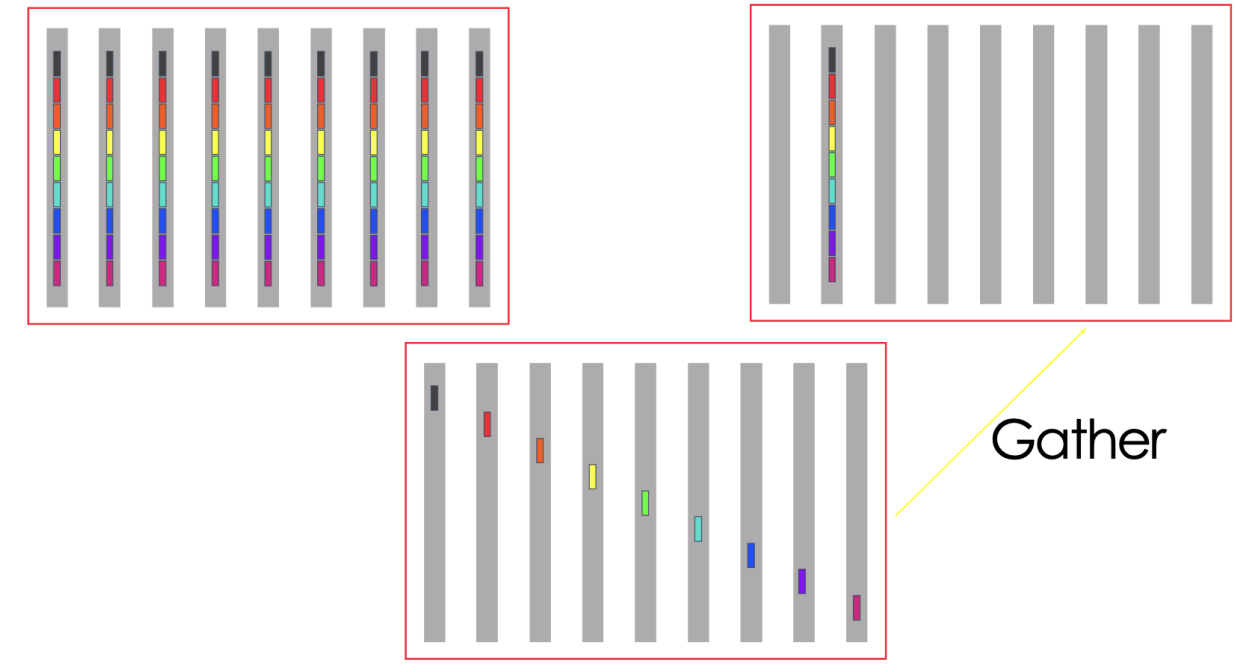

Figure 9: Reduce Step 2

Cost associated with this is:

$$(p-1)\alpha + \frac{p-1}{p}n\beta + \frac{p-1}{p}n\gamma$$
$$+$$
$$log(p)\alpha + \frac{p-1}{p}n\beta$$
$$= (log(p) + p - 1)\alpha + 2\frac{p-1}{p}n\beta + \frac{p-1}{p}n\gamma$$

3.6 Allreduce (Large Message)

This algorithm involves Reduce-scatter followed by Allgather. So first get a fraction of sum in each and then get each worker to hold the whole message.

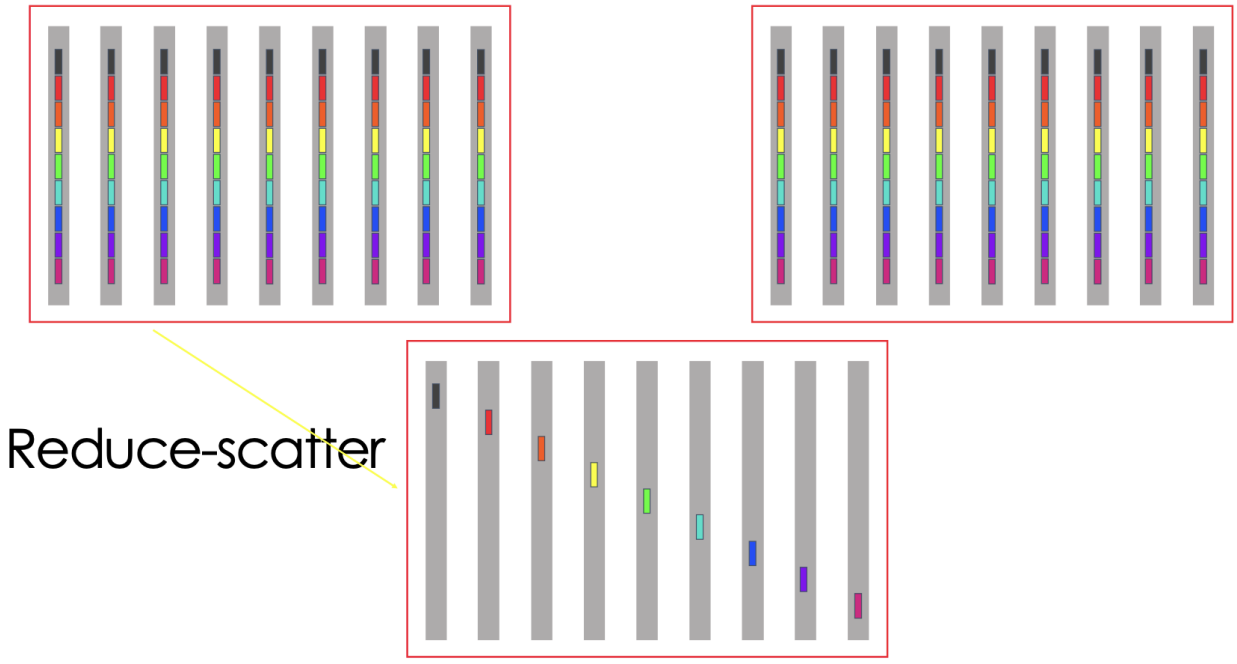

Figure 10: AllReduce Step 1

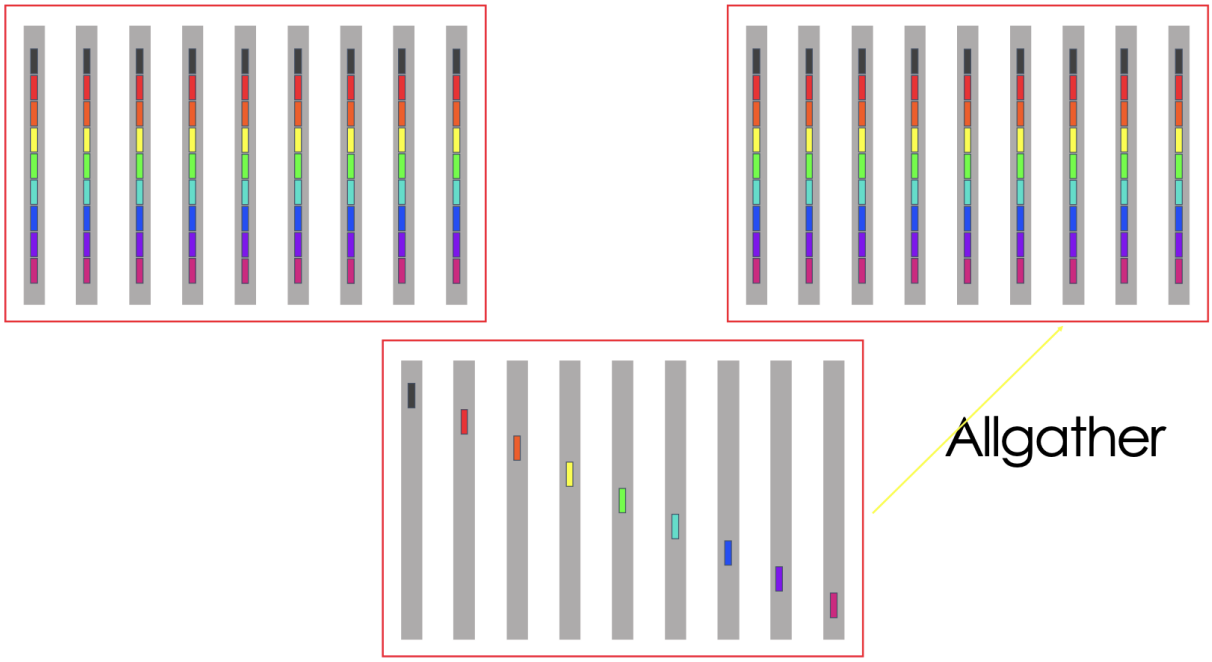

Figure 11: AllReduce Step 2

Cost:

$$\begin{aligned}
&(p-1)\alpha + \frac{p-1}{p}n\beta + \frac{p-1}{p}n\gamma \\
&+ \\
&(p-1)\alpha + \frac{p-1}{p}n\beta \\
&= 2(p-1)\alpha + 2\frac{p-1}{p}n\beta + \frac{p-1}{p}n\gamma
\end{aligned}$$

As compared to Reduce-broadcast allreduce, which has a cost $= 2log(p)\alpha + 2log(p)n\beta + log(p)n\gamma$, this does better in bandwidth. Moreover, this algorithm maps to distributed SGD gradient synchronization step.

3.7 Recap

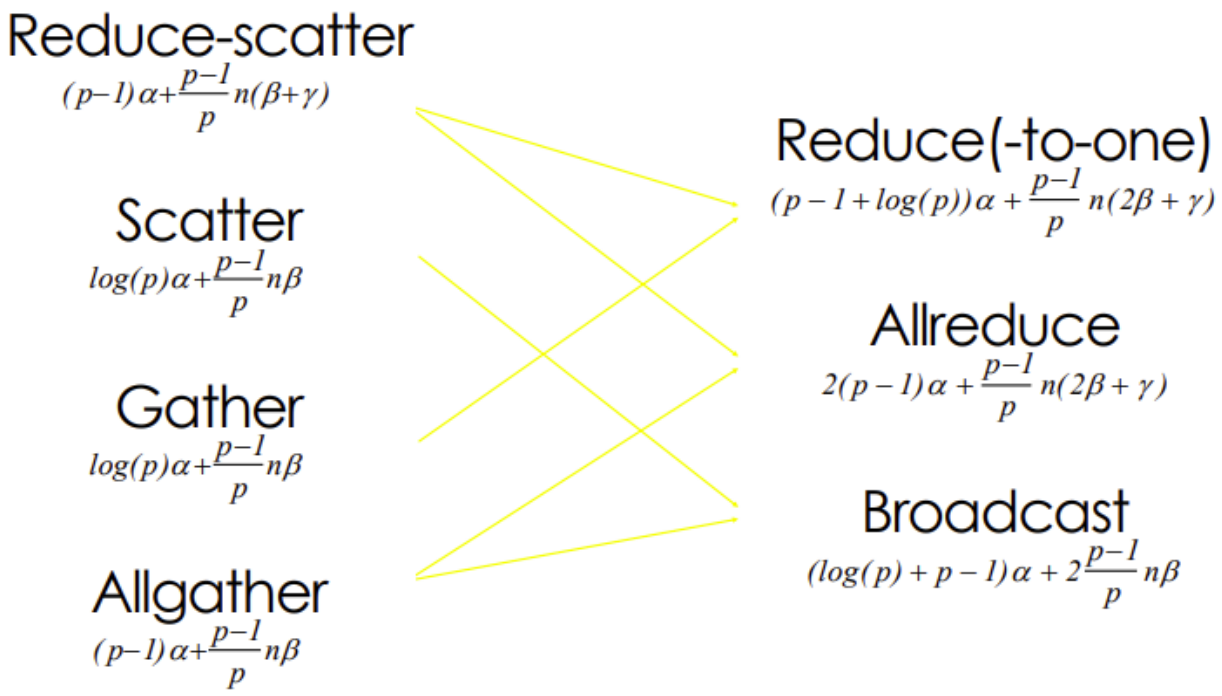

Figure 12: Collective Communication

Reduce(-to-one) is equal to Reduce-scatter plus Gather. Broadcast is equal to Scatter plus Allgather. Allreduce is equal to Reduce-scatter plus Allgather. Especially, Allreduce, which is equal to Reduce-scatter plus Allgather, is the key technique enabling ChatGPT. This characteristic is also used to develop a effective and famous distributed training technology called ZeRo or PyTorch Fully Sharded Data Parallel(FSDP). In PyTorch FSDP, this charateristic is used to optimize memory.

4 Real Cluster to Train ChatGPT

One way is 2D Mesh and the other way is 3D mesh. If using GPU, then it is 2D mesh. If using TPU, then it is 3D mesh.

In 2D mesh, we have many GPU boxes. Inside each box, there are GPUs arranged in linear array. Therefore, one dimension extends along the GPU boxes, while the other dimension extends along the GPUs.

Figure 13 shows the 3D mesh case.

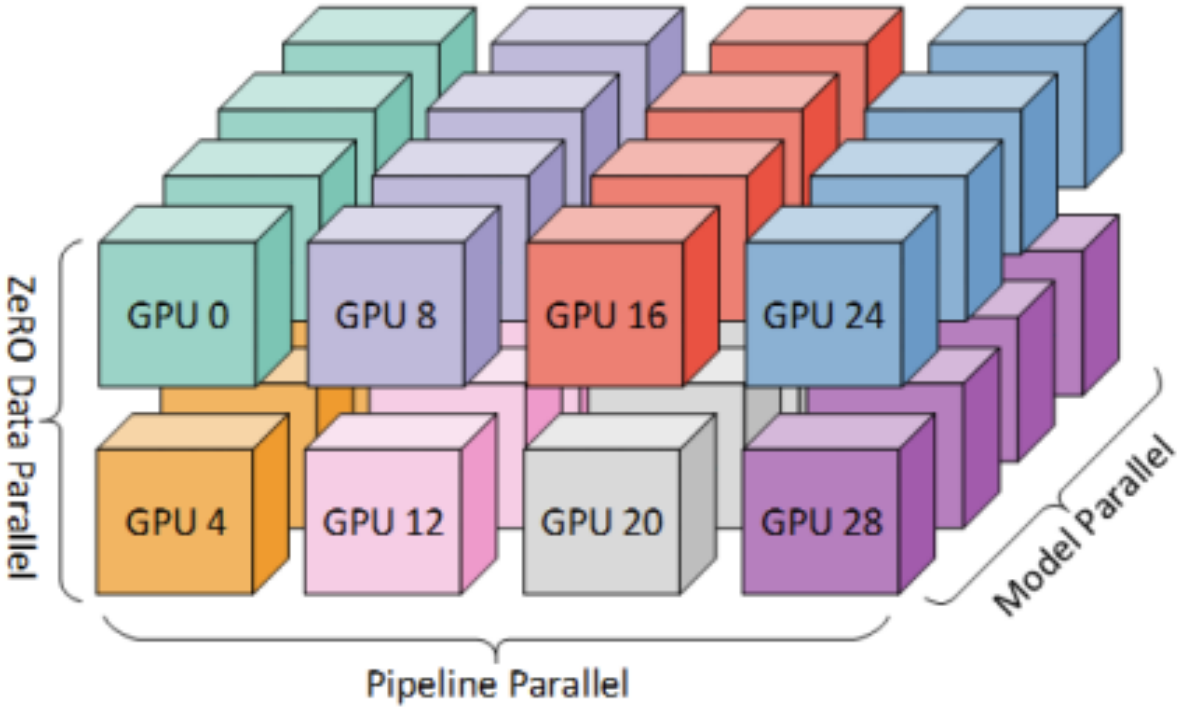

Figure 13: 3D Mesh

4.1 2D Broadcast

Figure 14: 2D Mesh Broadcast

The idea of the broadcast is to use 1D to compose 2D. There are at least 3 potions to do this.

4.2 2D Broadcast Option 1

Figure 15: option1 step1

Figure 16: option1 step2

The first option is to first use MST to broadcast in column and then broadcast in rows.

4.3 2D Broadcast Option 2

Figure 17: option2 step1

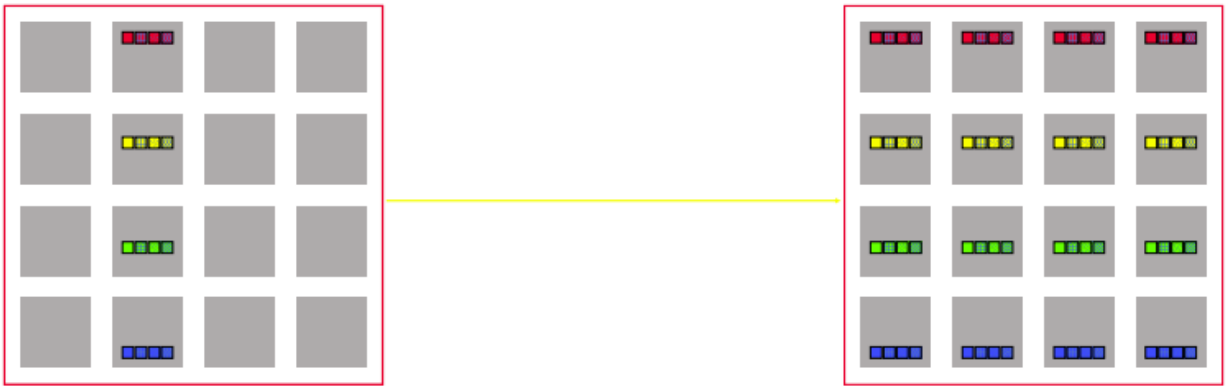

Figure 18: option2 step2

Figure 19: option2 step3

The second option is to first scatter in column, then use MST broadcast in rows, and finally allgather in columns.

4.4 2D Broadcast Option 3

Figure 20: option3 first step

Figure 21: option3 last step

The third option is to first scatter in column, then scatter in rows, then allgather in rows and finally allgather in columns.

4.5 Cost Comparison

The cost of option 1 is

$$\begin{aligned} & log(c)\alpha + log(c)n \\ & + \\ & log(r)\alpha + log(r)n\beta \\ = & log(p)\alpha + log(p)n\beta \end{aligned}$$

The cost of option 2 is

$$\begin{aligned} & log(c)\alpha + \frac{c-1}{c}n\beta \\ & + \\ & log(r)\alpha + log(r)\frac{n}{c}\beta \\ & + \\ & (c-1)\alpha + \frac{c-1}{c}n\beta \\ = & (log(p) + c - 1)\alpha + 2\frac{c-1+log(r)}{c}n\beta \end{aligned}$$

The cost of option 3 is

$$\begin{aligned} & log(c)\alpha + \frac{c-1}{c}n\beta \\ & + \\ & log(r)\alpha + \frac{r-1}{r}\frac{n}{c}\beta \\ & + \\ & (r-1)\alpha + \frac{r-1}{r}\frac{n}{c}\beta \\ & + \\ & (c-1)\alpha + \frac{c-1}{c}n\beta \\ = & (log(p) + r + c - 2)\alpha + 2\frac{p-1}{p}n\beta \end{aligned}$$

4.6 Summary

When α dominates, we can use MST algorithm. When $n * \beta$ dominates, we can use Ring algorithm. 2D can be composed using 1D, 3D can be composed using 2D. We need to make Latency / Bandwidth trade-offs to decide which algorithms to use.

4.7 Recap Questions

Q1: Which collective primitive maps to the distributed SGD gradient synchronization step?

A1: Allreduce

Q2: How many messages do we need to transfer over the network for a single iteration of GPT-3 SGD update assuming 8-gpu parallelism?

A2: We focus on the lower bound. The size of the message is 350 gigabyte of parameters. Each worker will have such size of message. We need to perform Allreduce on 8 workers because there are 8 GPUs. Therefore, the lower bound is 8*350 gigabyte messages.

Q3: For Q2, assuming 1D mesh, should we use MST or Ring? A3: Ring. Because message is supper big so the second term dominates.