Scaling Deep Learning

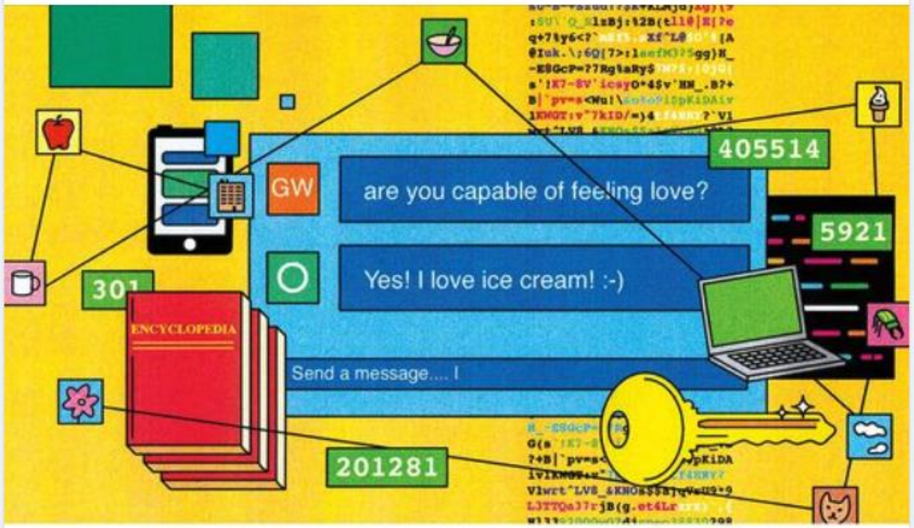
CS246: Mining Massive Datasets
Jure Leskovec, Stanford University
Charilaos Kanatsoulis, Stanford University
http://cs246.stanford.edu

Large Language Models

The Economist
May 6, 2023

Large creative AI models will transform life and work. But how exactly do they function? Read more about the promise and peril of artificial intelligence here: https://econ.trib.al/adS0ONj

Illustration: George Wylesol



Generative AI

How does ChatGPT actually work?

Despite the feeling of magic, large language models (LLMs) are, in reality, a giant exercise in statistics

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

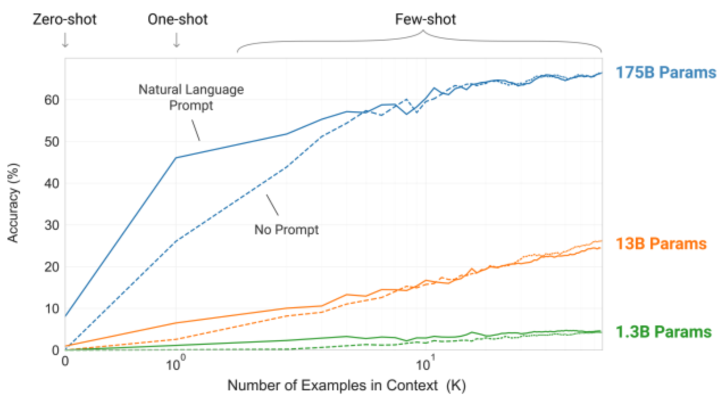
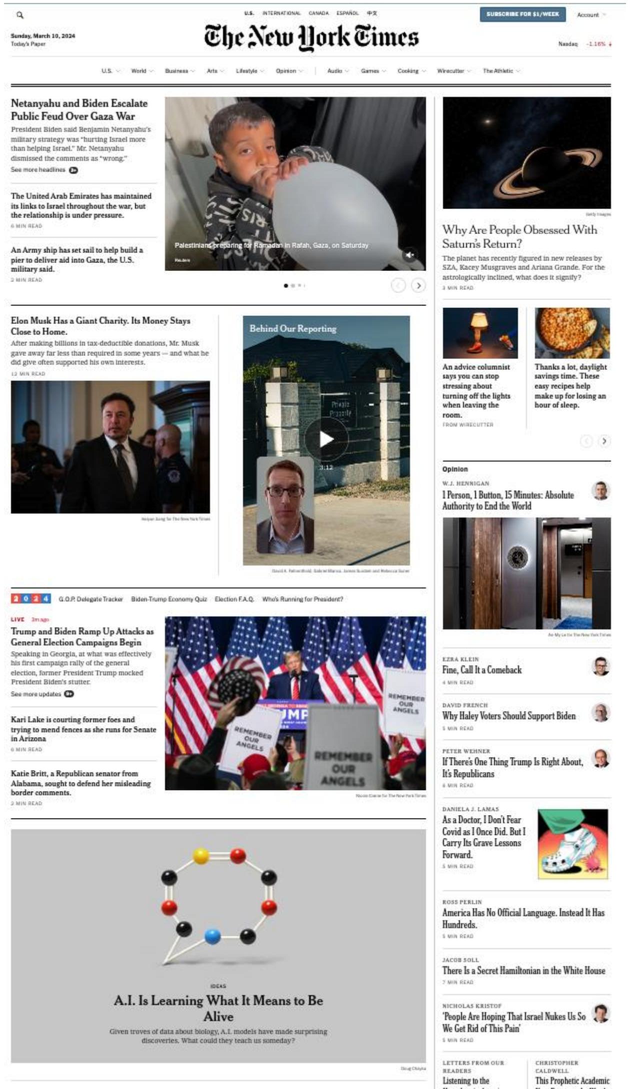
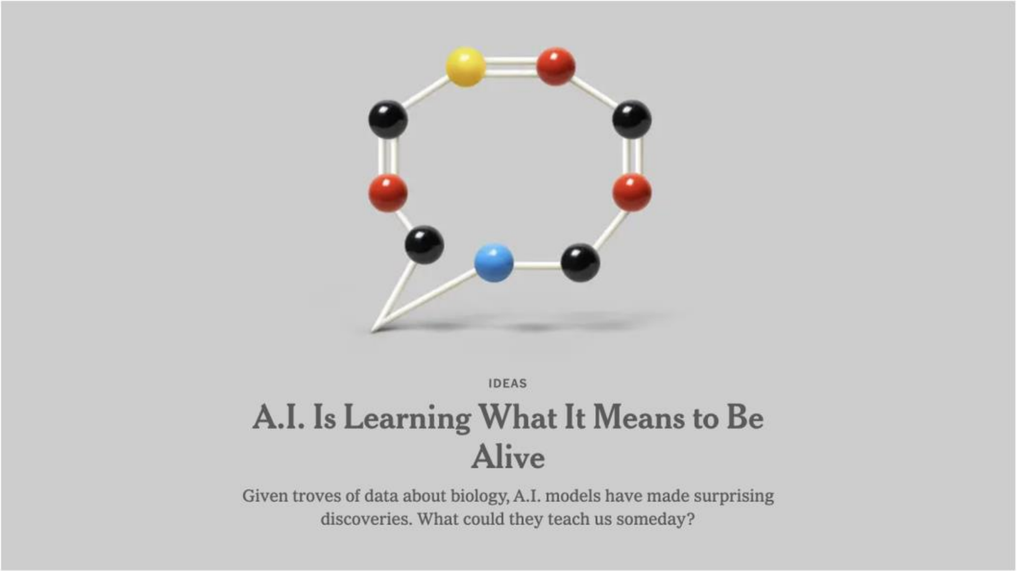


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper "in-context learning curves" for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

Foundation Models for Cell Biology


The New York Times
Sunday, March 10, 2024
Netanyahu and Biden Escalate Public Feud Over Gaza War
The United Arab Emirates has maintained its links to Israel throughout the war, but the relationship is under pressure.
An Army ship has set sail to help build a pier to deliver aid into Gaza, the U.S. military said.
Why Are People Obsessed With Saturn's Return?
Elon Musk Has a Giant Charity. Its Money Stays Close to Home.
Behind Our Reporting
Trump and Biden Ramp Up Attacks as General Election Campaigns Begin
Kari Lake is courting former foes and trying to mend fences as she runs for Senate in Arizona
Katie Britt, a Republican senator from Alabama, sought to defend her misleading border comments.
Opinion
1 Person, 1 Button, 15 Minutes: Absolute Authority to End the World
Fine, Call It a Comeback
Why Haley Voters Should Support Biden
If There's One Thing Trump Is Right About, It's Republicans
As a Doctor, I Don't Fear Covid as I Once Did. But I Carry Its Grave Lessons Forward.
America Has No Official Language. Instead It Has Hundreds.
There Is a Secret Hamiltonian in the White House
'People Are Hoping That Israel Nukes Us So We Get Rid of This Pain'
IDEAS
A.I. Is Learning What It Means to Be Alive
Given troves of data about biology, A.I. models have made surprising discoveries. What could they teach us someday?



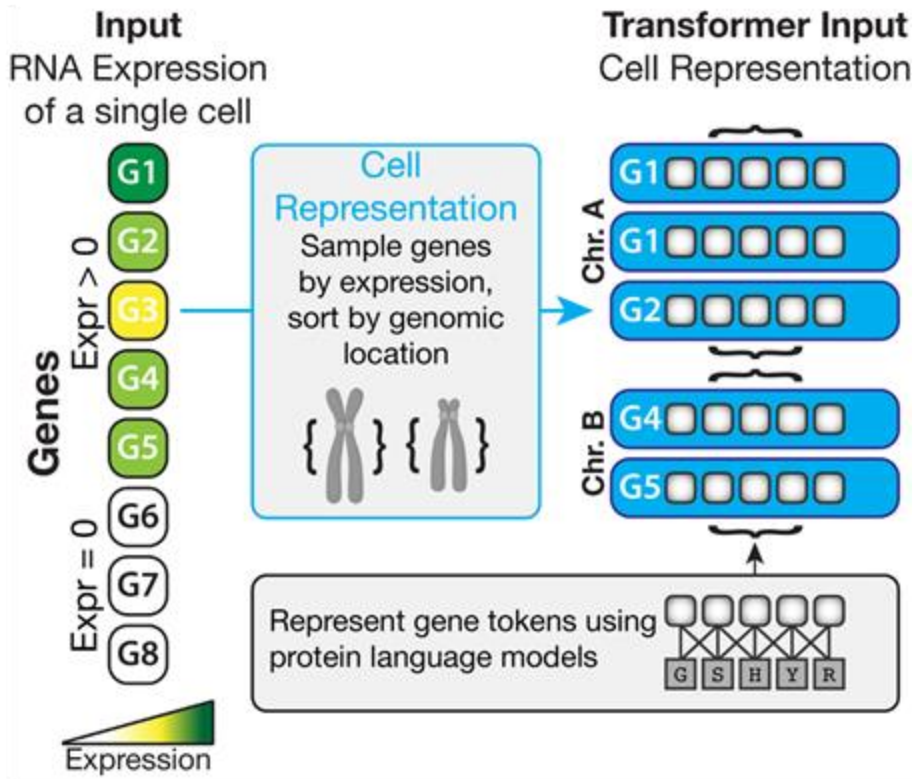
Doug Chayka

Foundation Models for Cell Biology

- UCE creates universal representations of cells

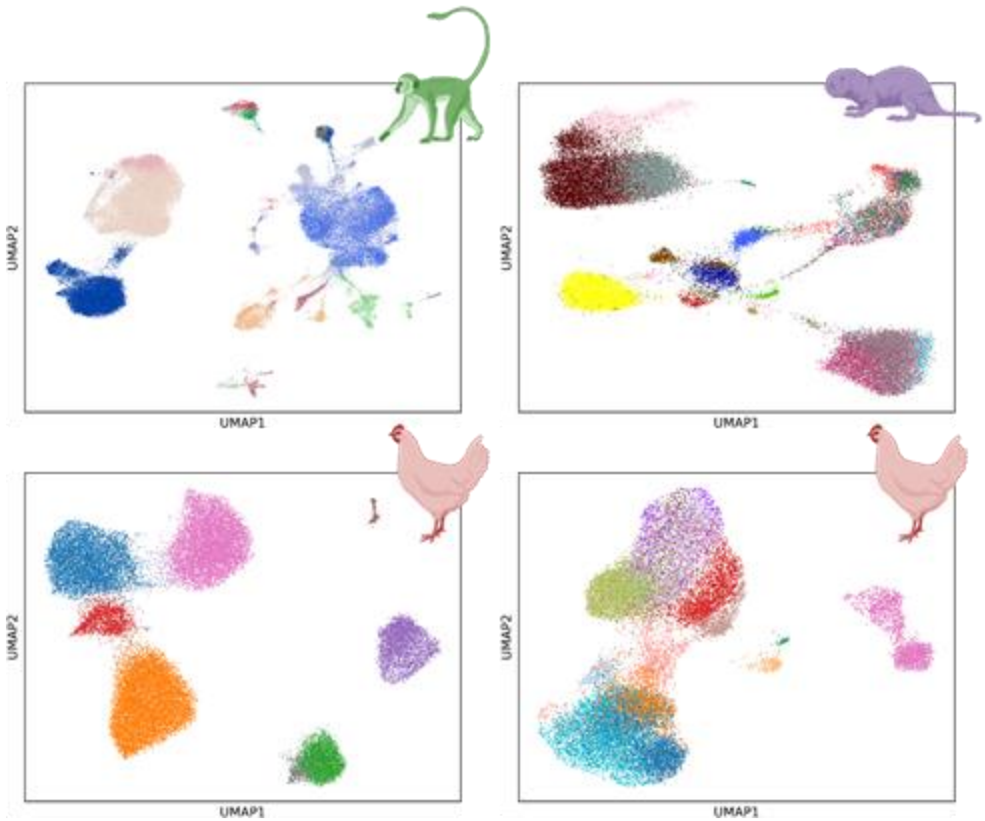
Input:
RNA expression of a single cell/nucleus

Output:
Cell Embedding

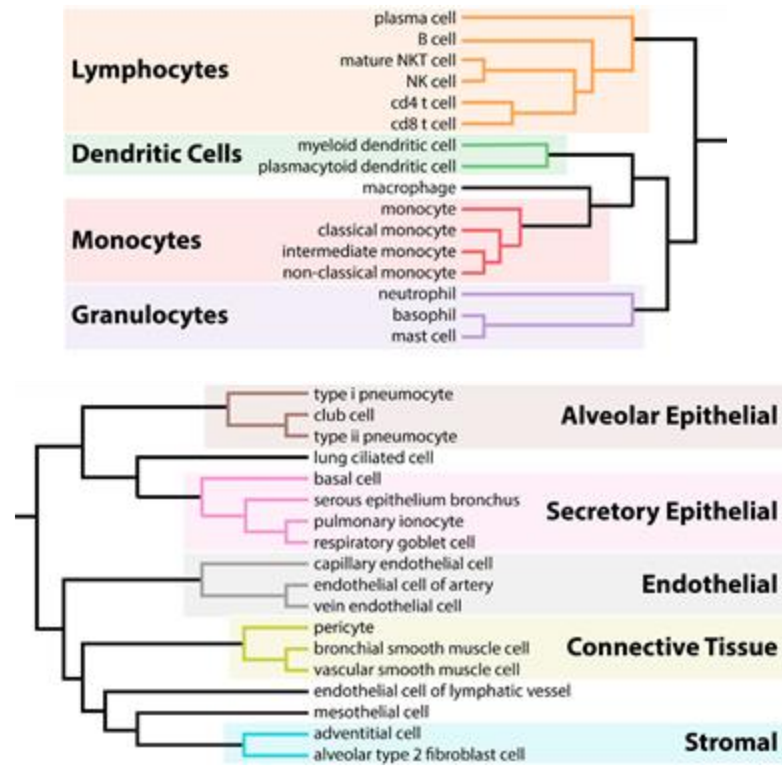


Foundation Models for Cell Biology

Visualize and transfer annotations



Infer Hierarchies

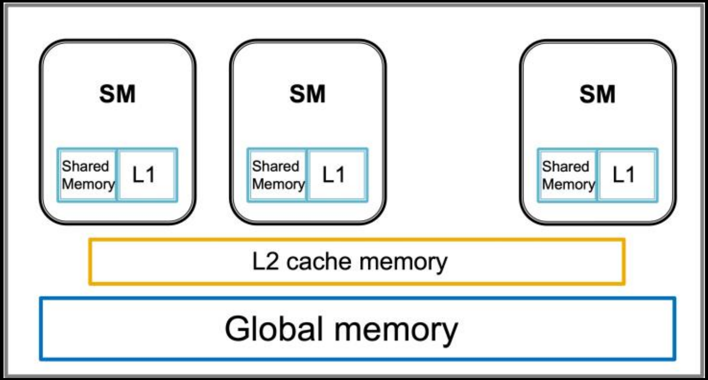


Training Large Models

- Modern AI trains very large models with a huge amount of data
- GPT-3 is trained with 500B tokens, more than 600 GB of Data
- The model has 175B parameters, requiring 350 GB of storage space
- The memory capacity of modern GPUs is 10-80 GB
- There is need for developing large-scale methods that can train such models

Today's Lecture

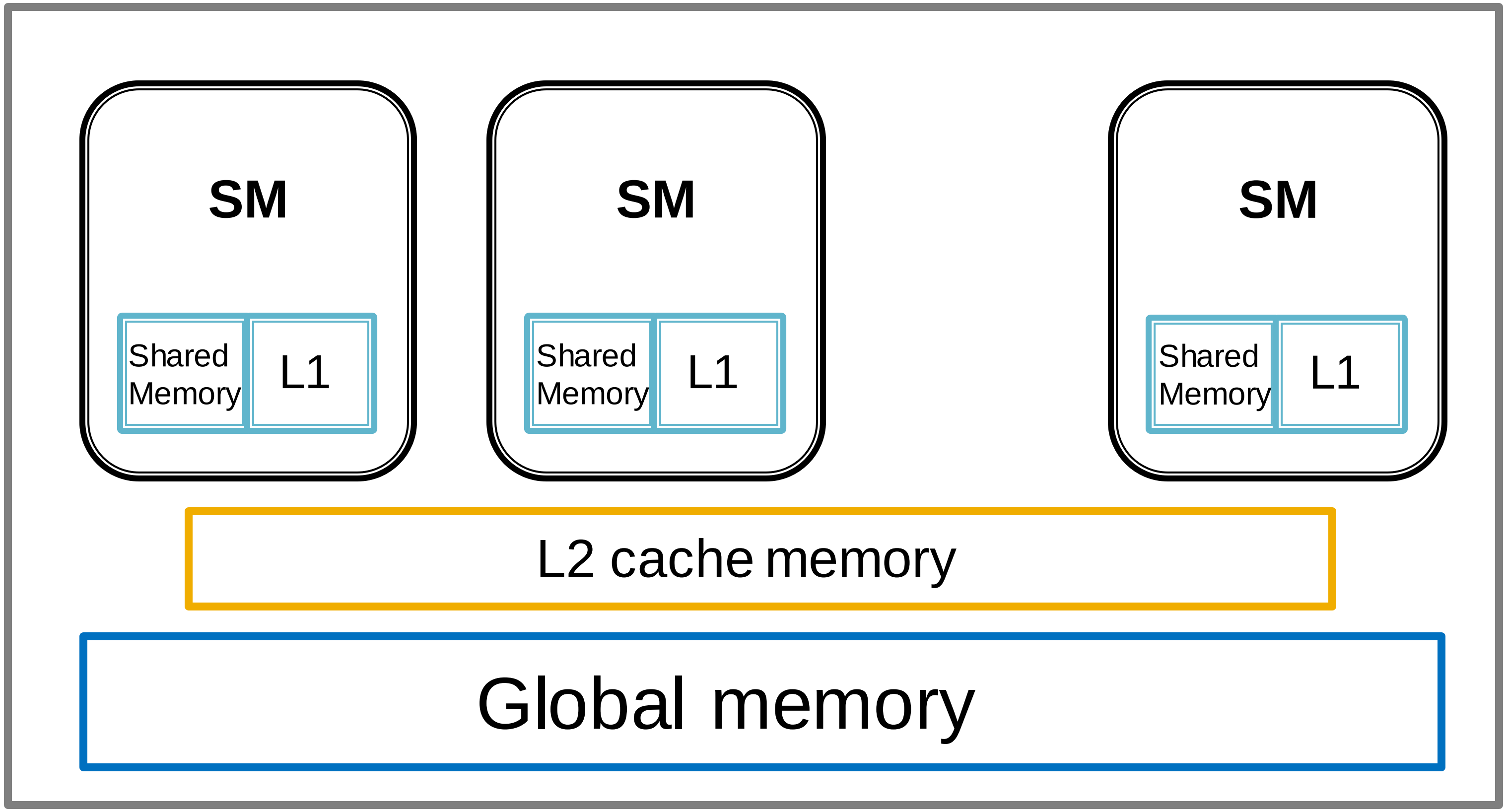
- **How to train large deep learning models?**
- **Memory Optimization Methods**
- **Parallel and distributed training with multiple GPUs**
  - **Model Parallel Training**
  - **Data Parallel Training**



Memory Optimization

Acknowledgements: Tianqi Chen, Deep Learning Systems Course, Carnegie Mellon University

GPU Architecture hierarchy

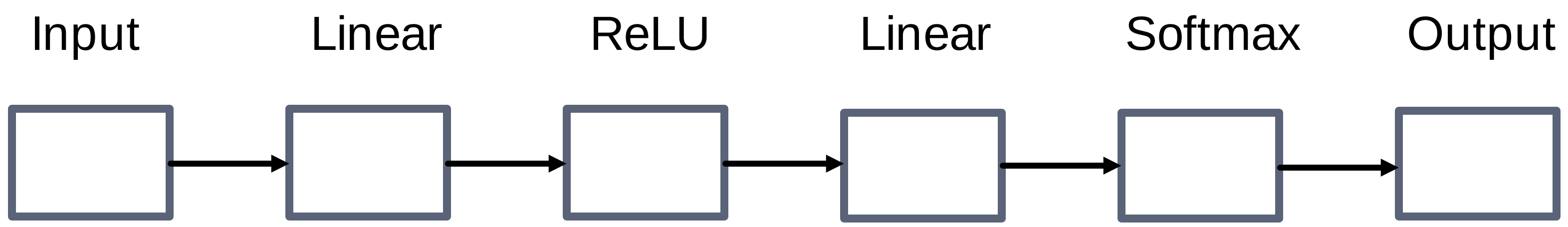


- **RTX 3090: 7GPU clusters, 84 SMs per cluster, 24 GB memory**
- **H100: 8 GPU clusters, 144 SMs per cluster, 80 GB memory**

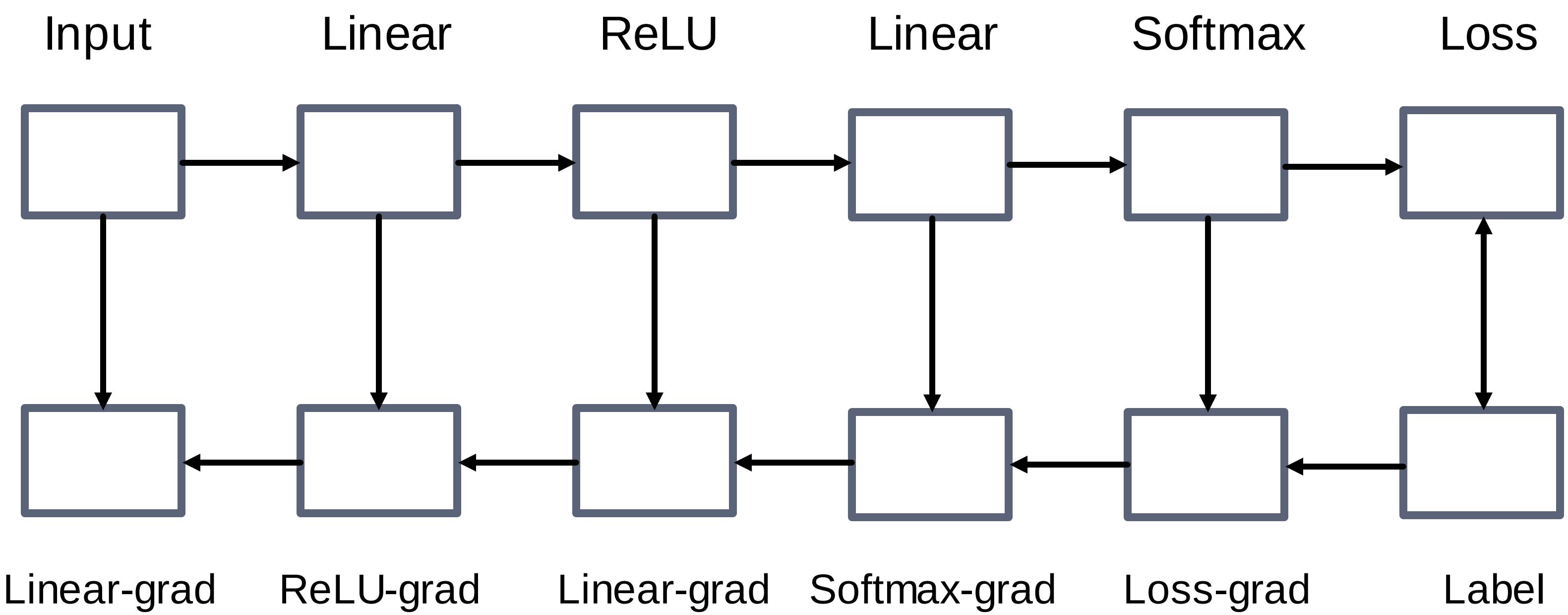
Sources of Memory Consumption

- **Input Data:** sequencies, images, graphs, etc.
- **Trainable parameters**
- **Auxiliary optimization variables**
- **Intermediate activation values and gradients**
- **Training Deep Nets with Sublinear Memory Cost [Chen et al., 2016]**
  - https://arxiv.org/pdf/1604.06174.pdf

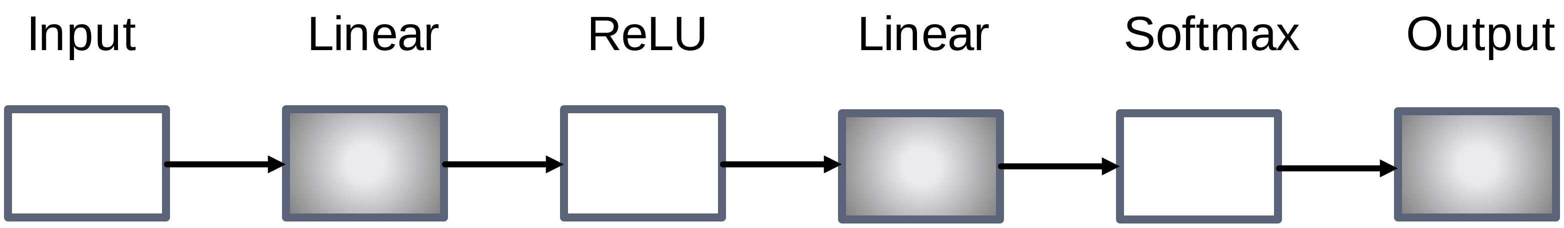
Computation graph



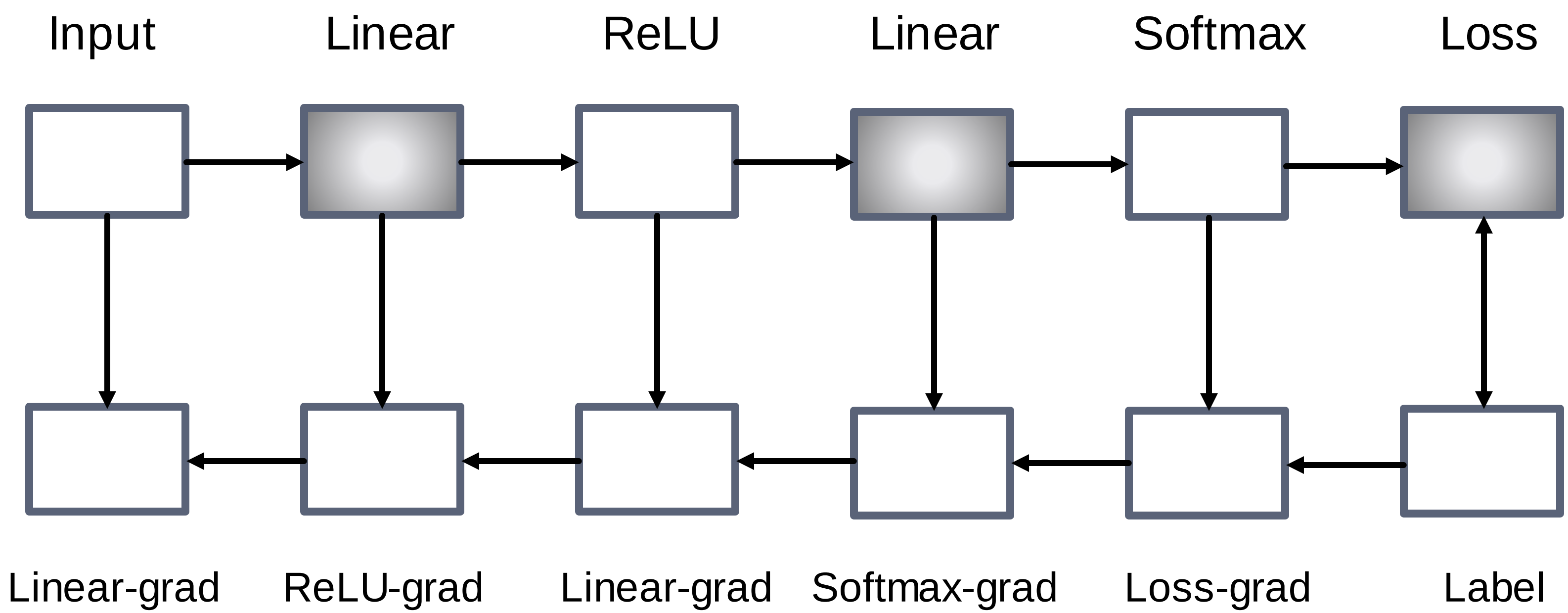
Computing Gradients



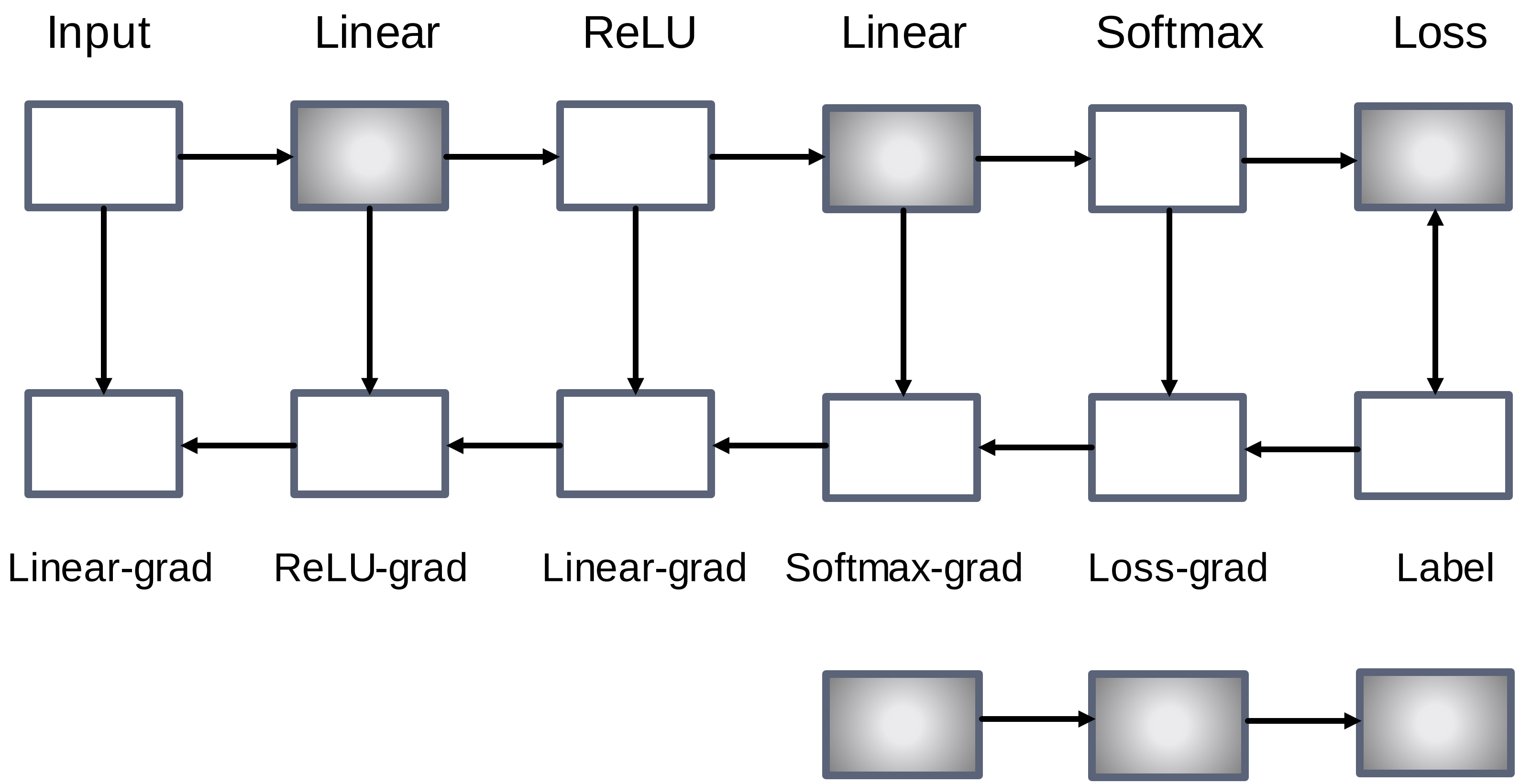
Checkpointing



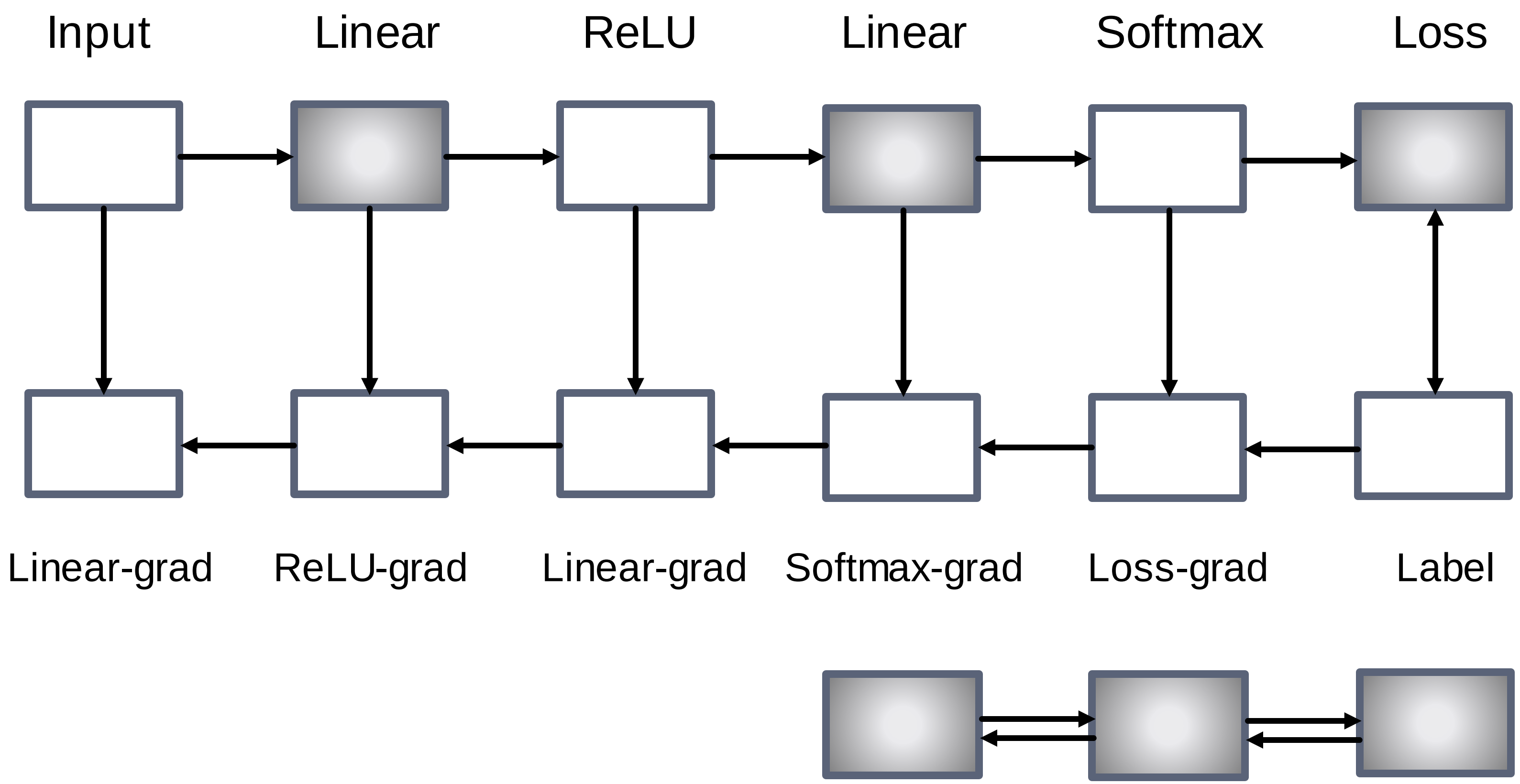
Checkpointing with backprop



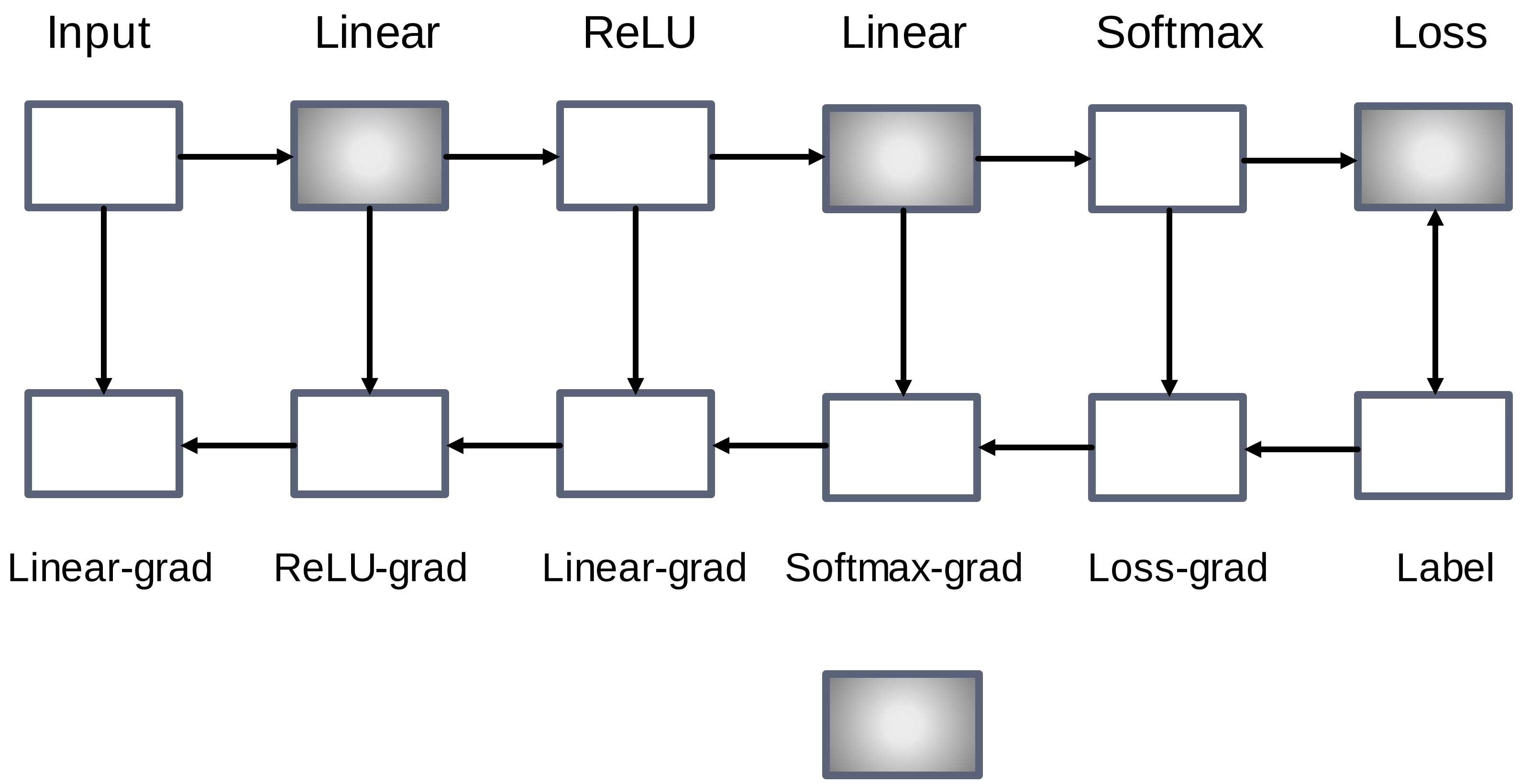
Checkpointing with backprop



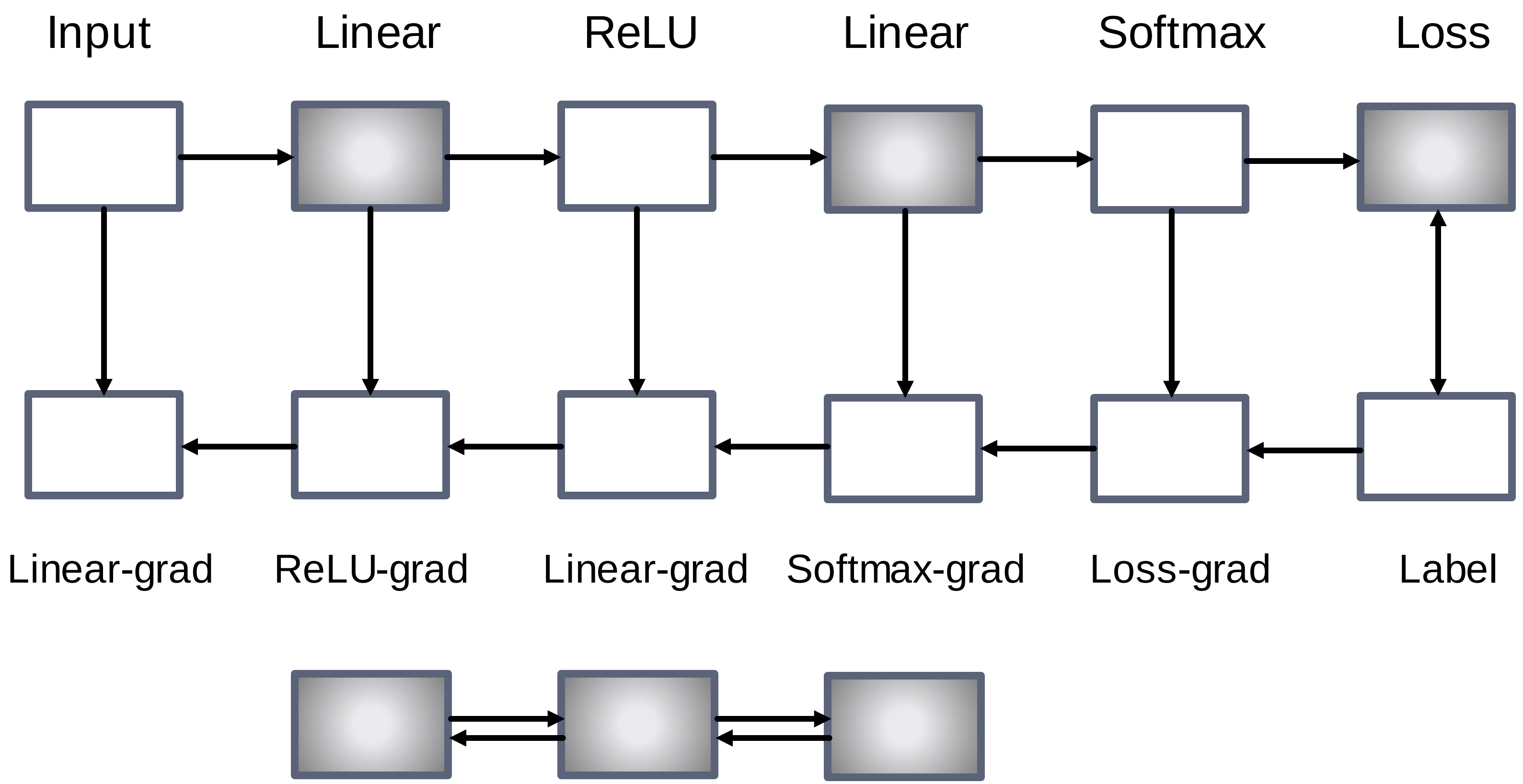
Checkpointing with backprop



Checkpointing with backprop



Checkpointing with backprop



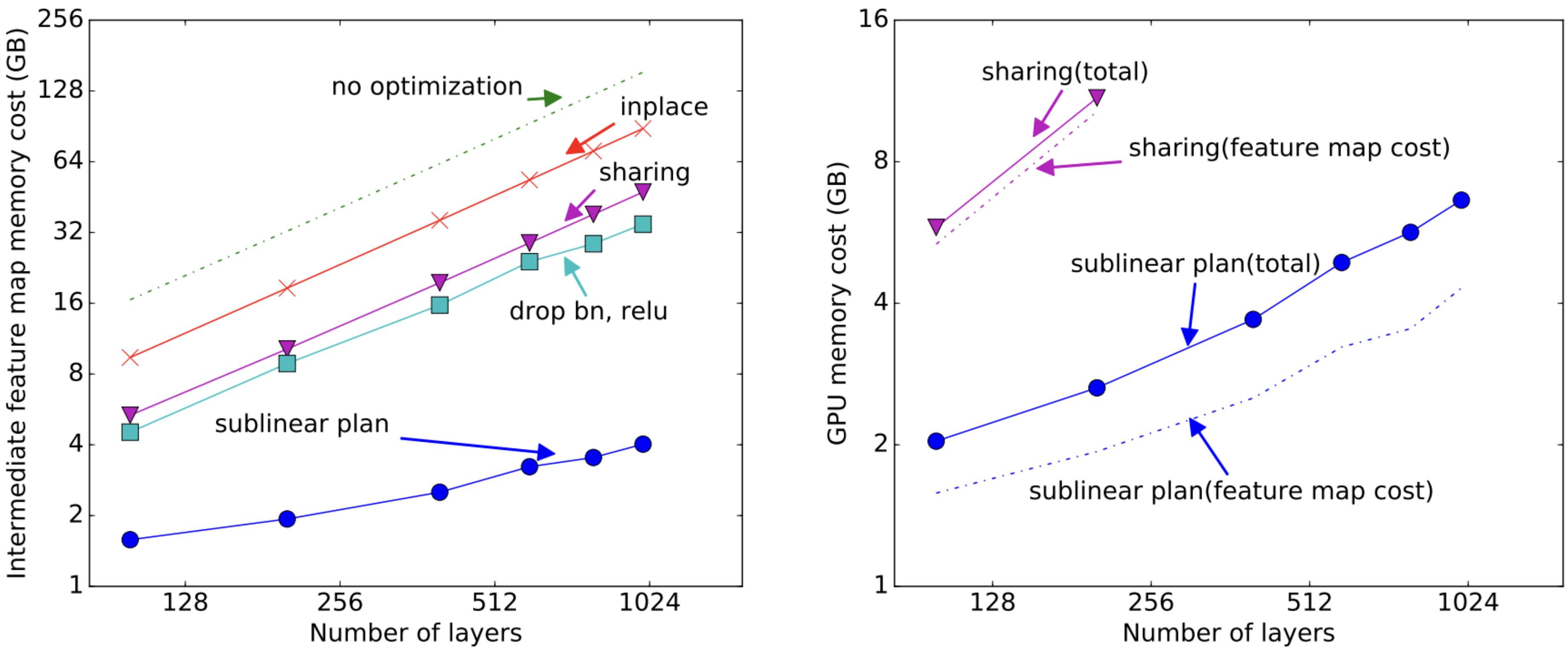
Sublinear Memory Cost

- The memory cost of the previous approach is: $O\left(\frac{N}{2}\right)$ for N Neural Network Layers and $O(1)$ for additional computations.
- If we checkpoint every K layers, the total memory cost is:

$$O\left(\frac{N}{K}\right) + O(K)$$

- **For $K = \sqrt{N}$ we reach sublinear memory cost!**

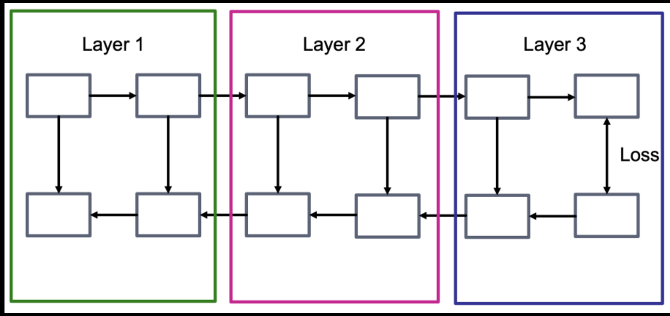
Sublinear Memory Cost



(a) Feature map memory cost estimation

(b) Runtime total memory cost

Source: Training Deep Nets with Sublinear Memory Cost [Chen et al., 2016]

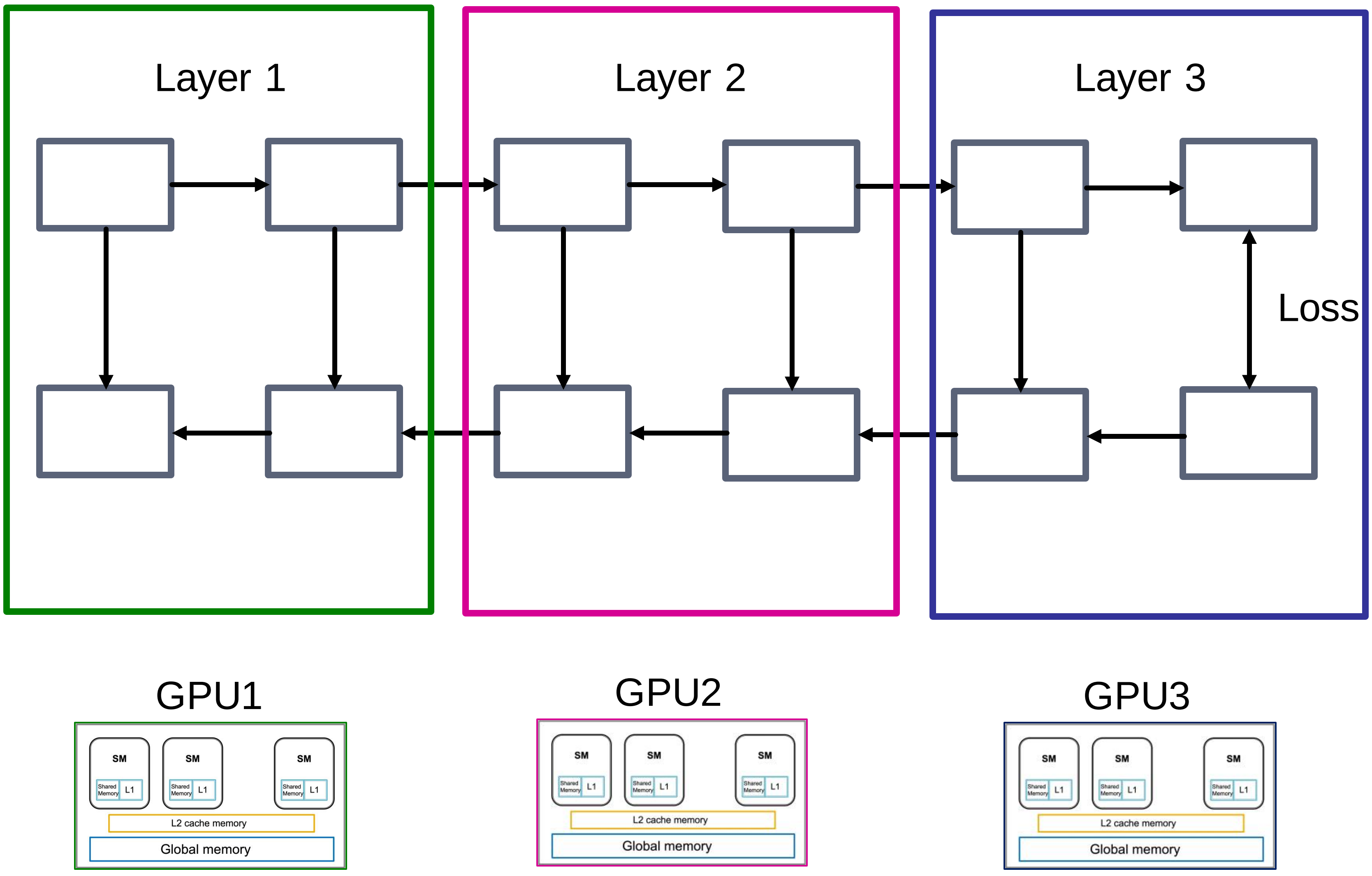


Model Parallel Training

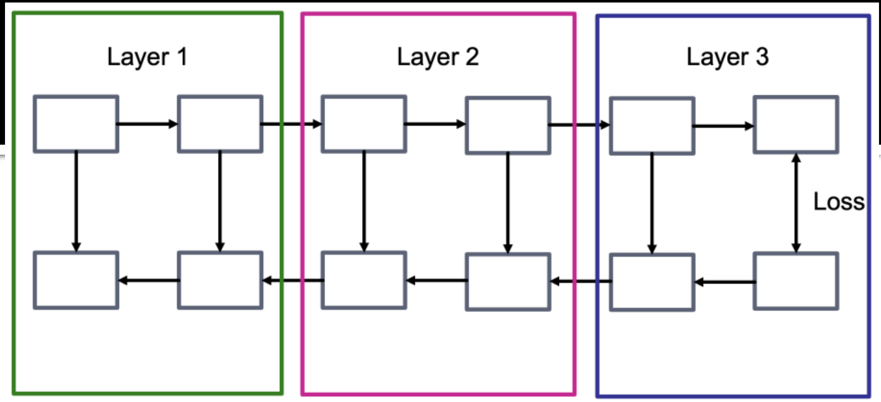
Model Parallelism

- **What if the model does not fit into GPU memory?**
- Idea: Split the model into submodels and fit each submodel into a different GPU

Model Parallel Training



Parallel Training

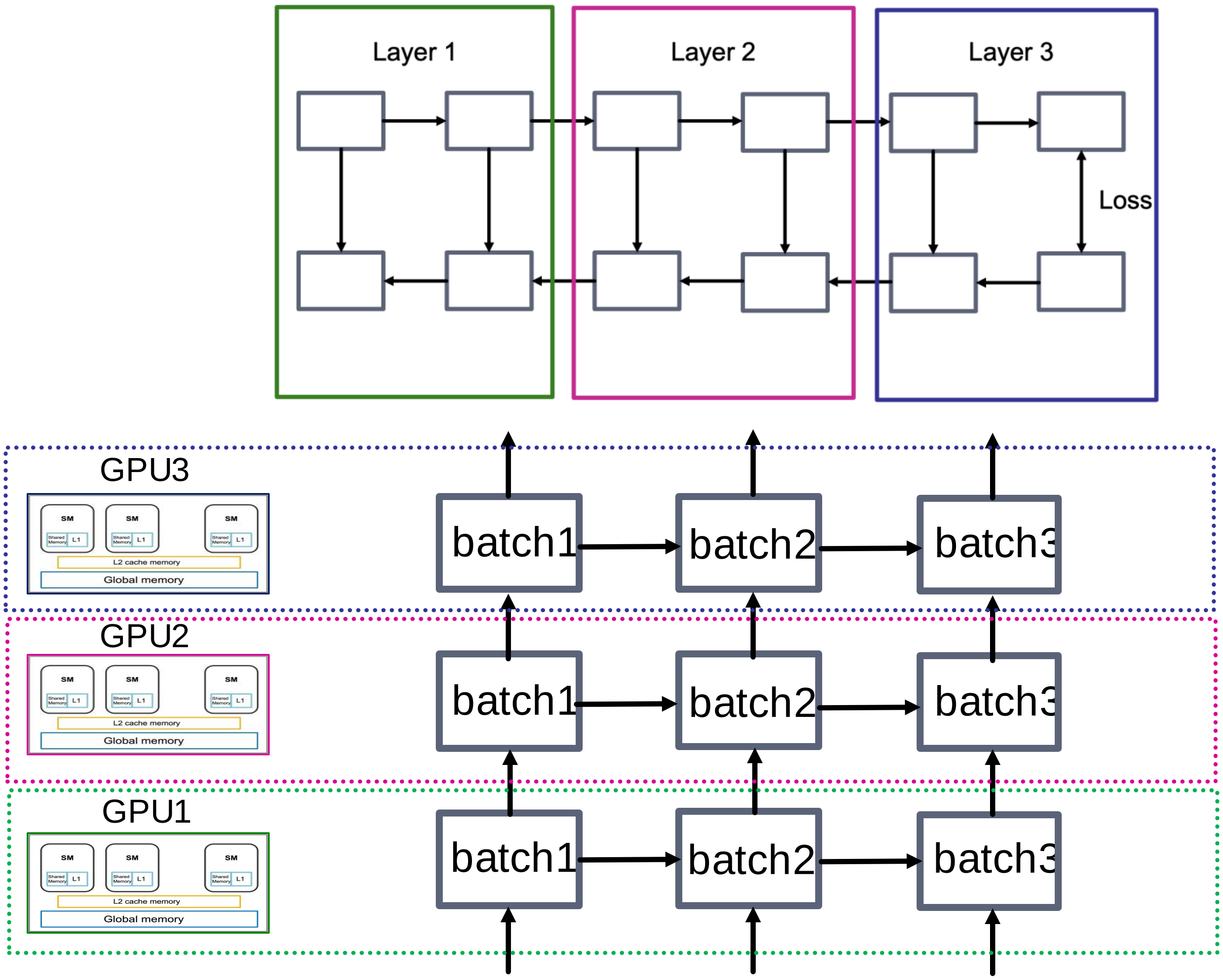


- Move input data GPU1
  - Run forward pass
- Move activations from GPU1 to GPU2
  - Run forward pass
- Move activations from GPU2 to GPU3
  - Run forward pass
  - Compute loss
  - Run backward pass
- Move gradients from GPU3 to GPU2
  - Run backward pass
- Move gradients form GPU2 to GPU1
  - Run backward pass
  - Apply gradient descent step

Challenges

- **How to move activations and gradients?**
  - **Via CPU: bad idea**
  - **Move things between GPUs**
- **How to reconcile for dependencies?**
  - **Pipelining**

Model Parallel Training



Other ways to do model parallelism

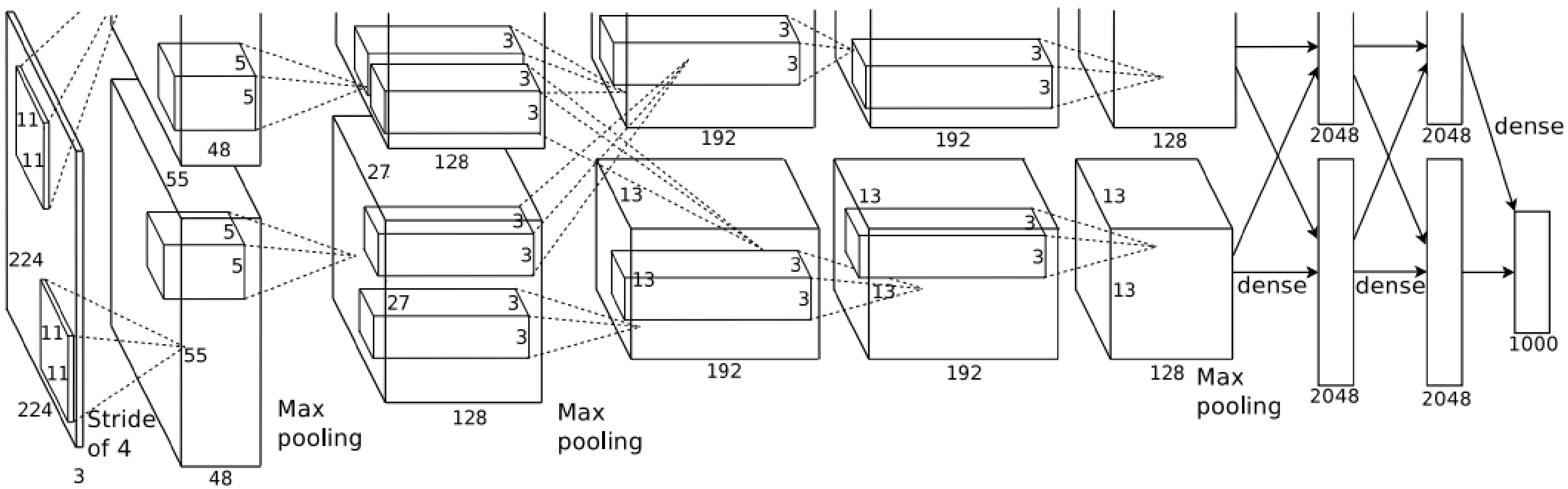
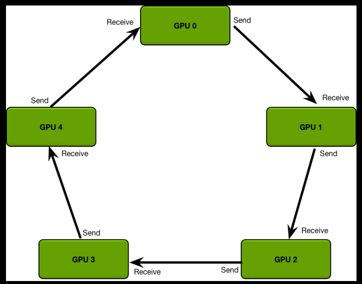


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Source: Imagenet classification with deep convolutional neural networks [Krizhevsky et al., 2012]

Model parallel training recap

- **Works for models that do not fit GPU memory**
- **Requires pipeline design to reconcile for dependencies**
- Pipelining requires synchronization
  - Not always that easy



Data Parallel Training

Acknowledgements: Mohamed Abdelfattah, Machine Learning Hardware and Systems Course, Cornell University

Data Parallelism

- **What if the model fits into GPU memory, but minibatches do not fit?**
- **Use ideas from HPC**
- Idea: Split minibatches into smaller batches and feed each of them into a different GPU

Data Parallelism

- **For each minibatch we need to compute**

$$\theta = \theta - \frac{\alpha}{\mathrm{n}} \sum_{i=1}^{\mathrm{n}} \nabla \mathcal{L}(f_\theta(x_i), y_i)$$

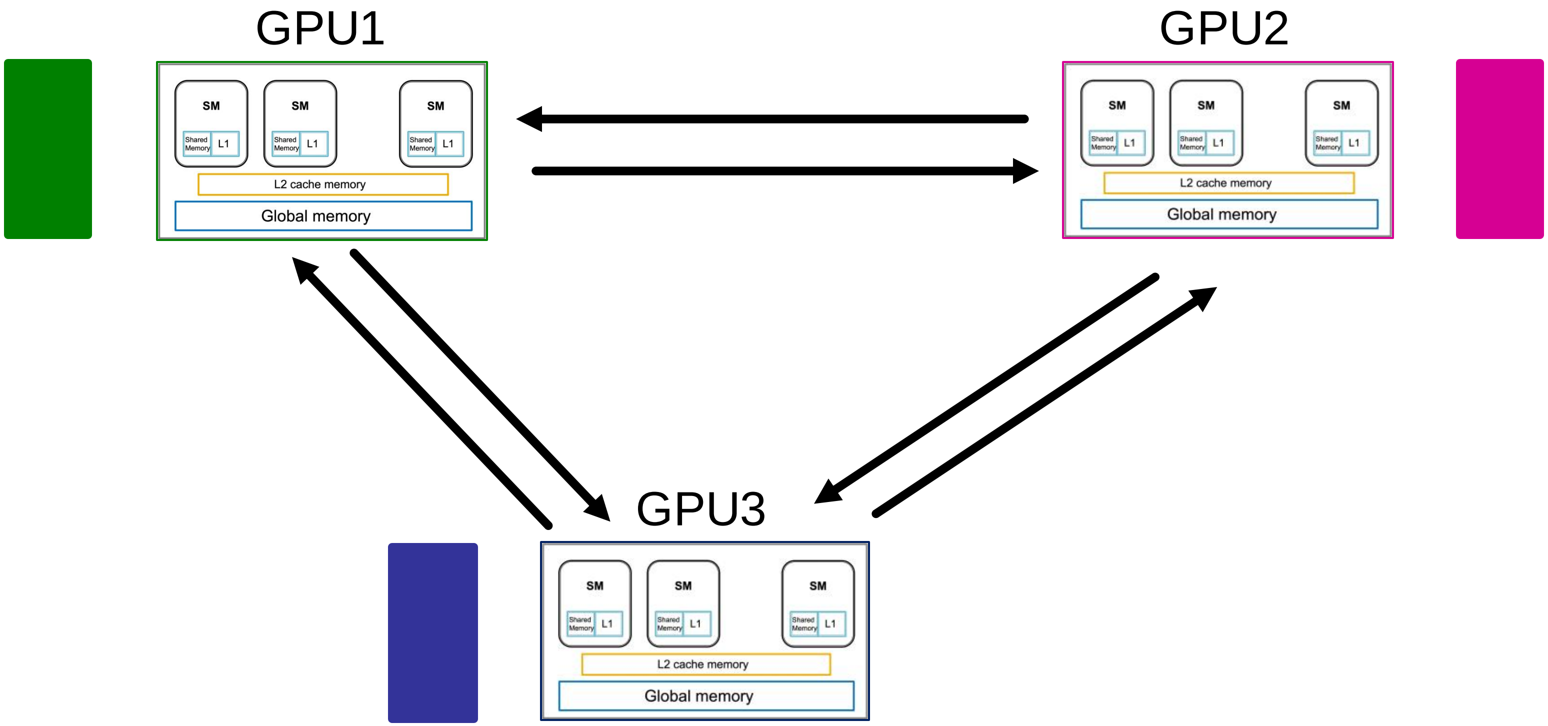
- Split each minibatch to K smaller batches

$$\theta_k = \theta - \frac{\alpha}{\mathrm{n}} \sum_{i=1}^{\mathrm{n}/K} \nabla \mathcal{L}(f_\theta(x_i), y_i)$$

$$\theta = \frac{1}{K} \sum_{\mathrm{k}=1}^{K} \theta_k$$

- First idea: Use the AllReduce framework

AllReduce



- Each GPU communicates d derivatives to K-1 GPUs

AllReduce

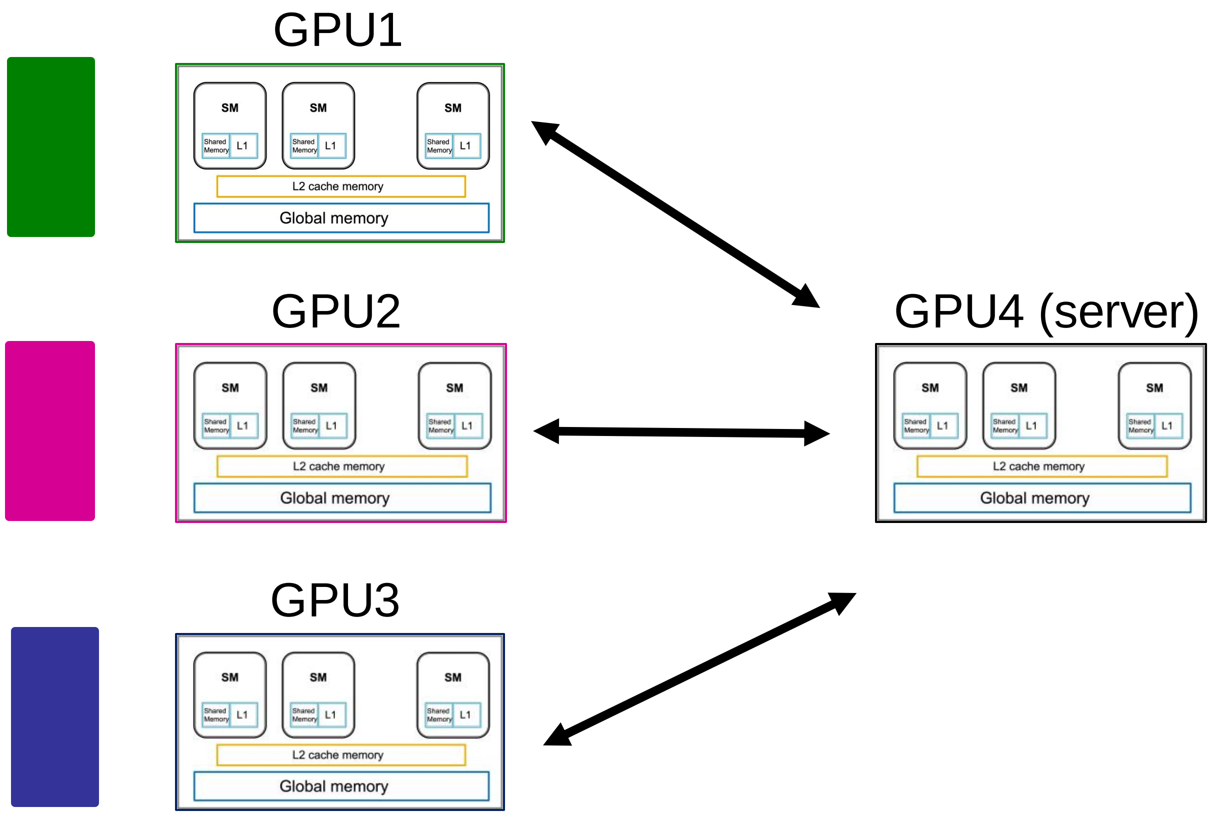
- **Total communication cost per GPU is**

d (K-1)

- Linear scaling with the number of GPUs

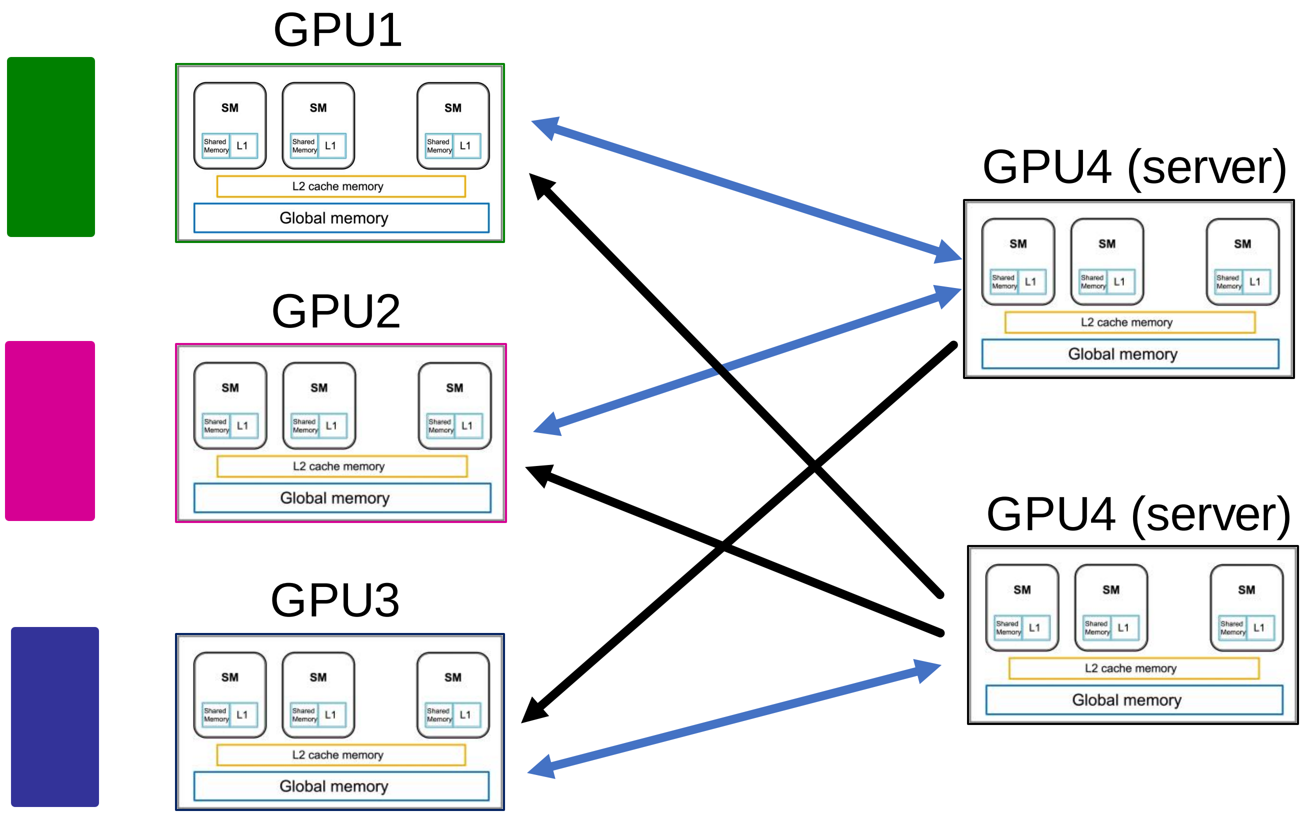
- Communication becomes a bottleneck

Parameter Server



- Each GPU communicates d derivates to 1 server
- Server broadcasts d derivatives to K servers:
- **Communication cost:** d K for the server

Parallel Parameter Server

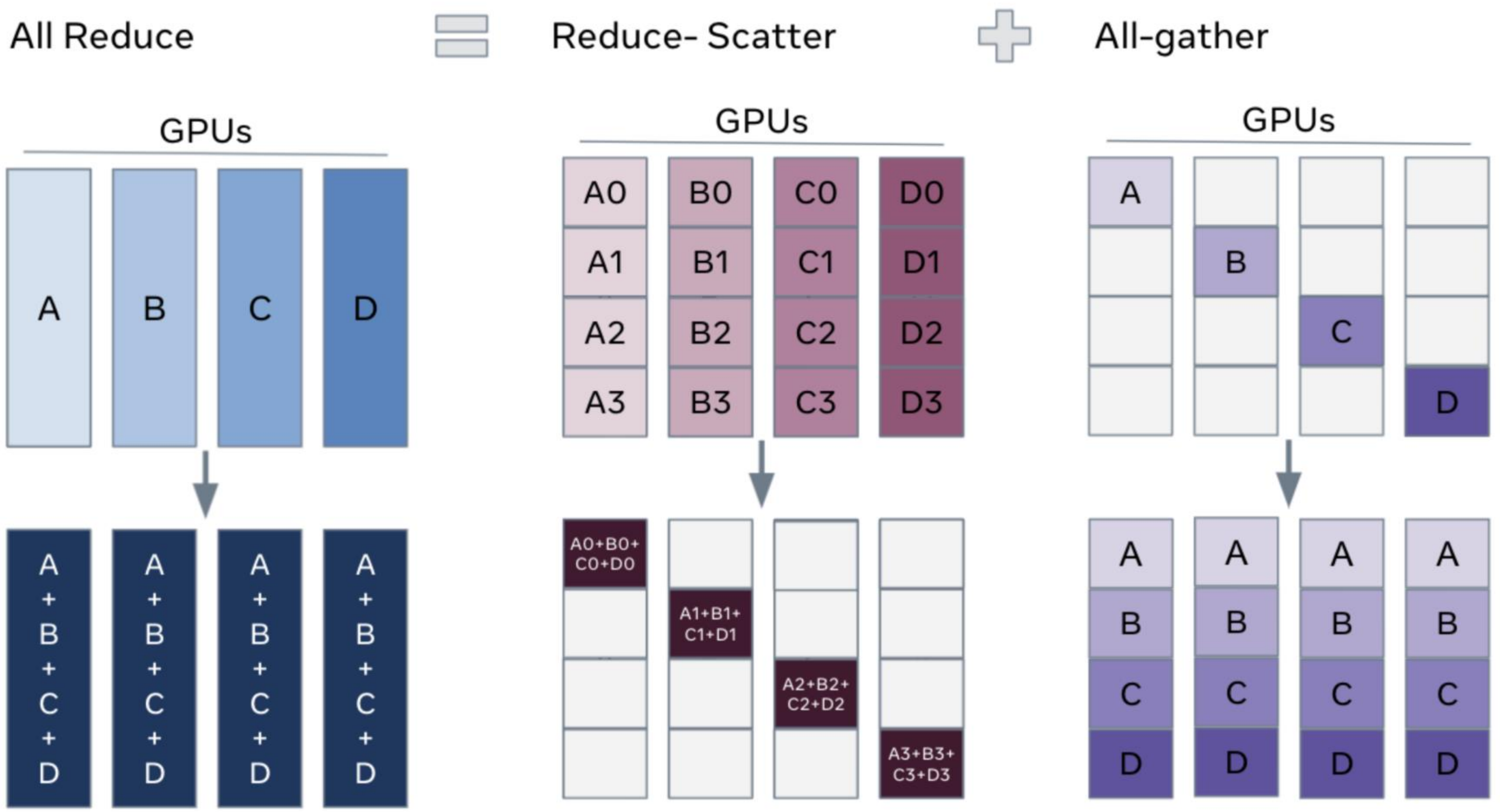


- Each GPU communicates d derivatives to 1 server
- Server broadcasts d derivatives to K / M servers
- **Communication cost:** d K / M for server

Breaking the Limits of Param. Server

- **Can we do better?**
- Idea: Decompose the **all-reduce** operations into separate **reduce-scatter** and **all-gather** operations

Ring-AllReduce



Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

Ring-AllReduce

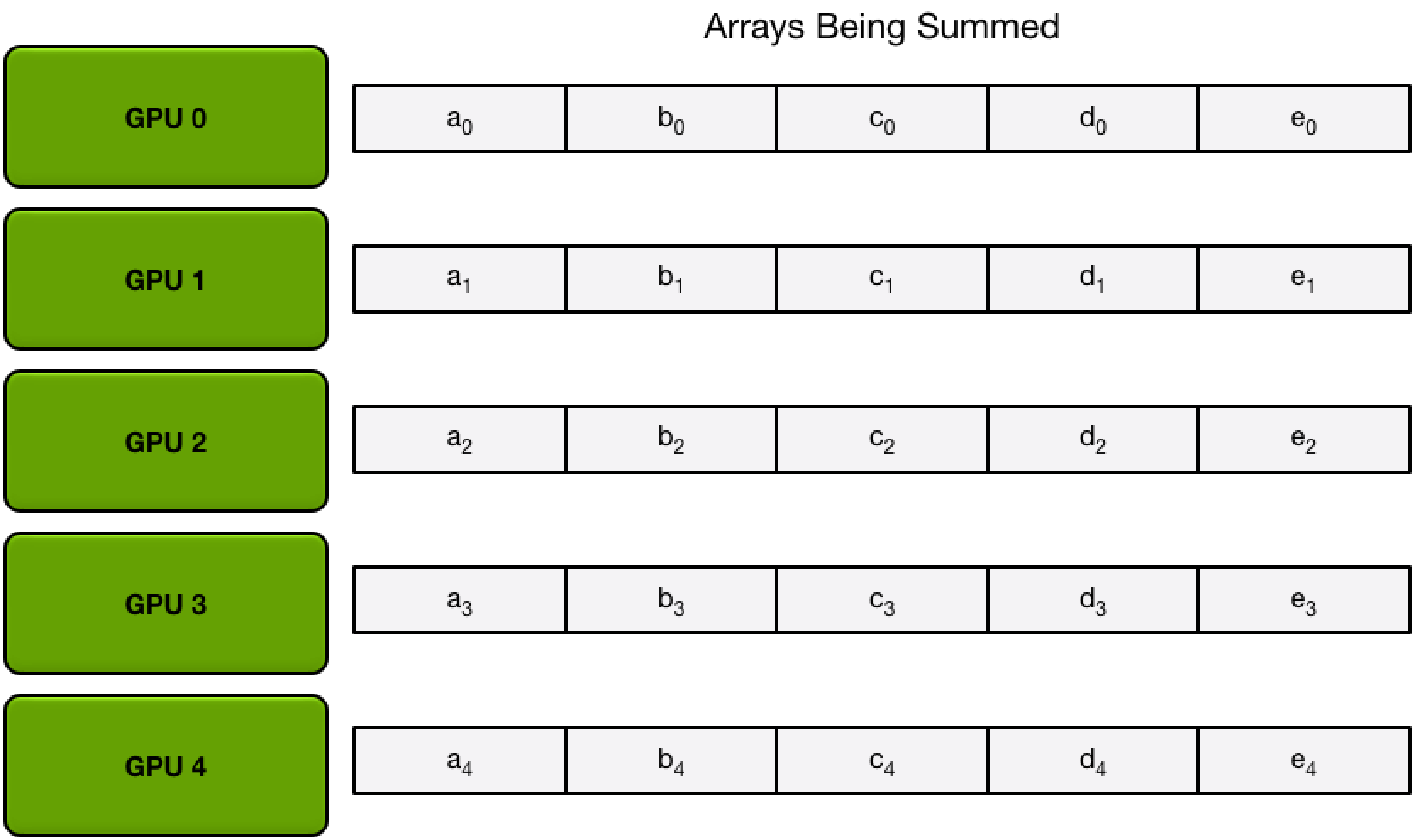
Phase 1: Reduce-scatter

- We divide the array in each GPU into chunks
- The gradients corresponding to the same chunk index are sequentially summed across all GPUs
- Each GPU has a fully aggregated gradient for one chunk

Phase 2: All-gather

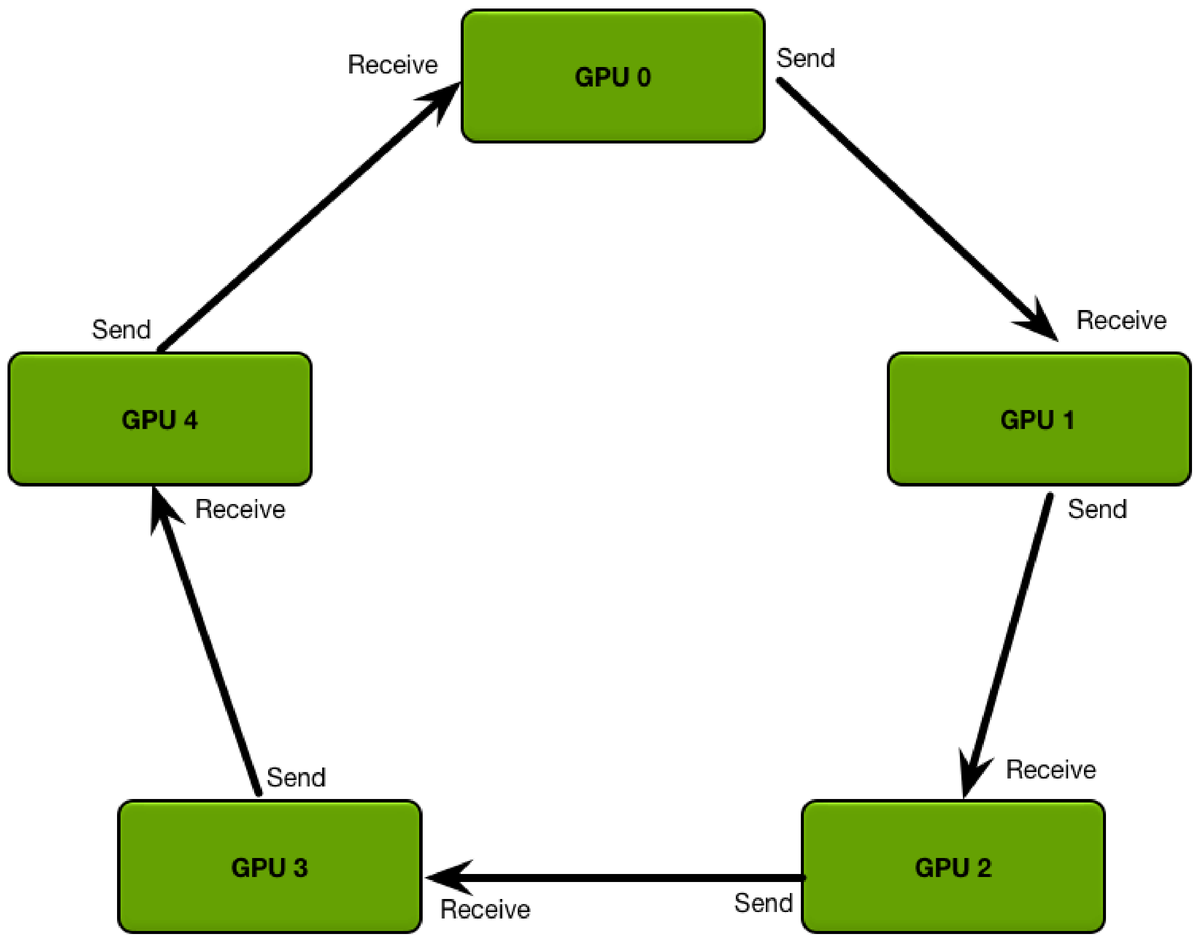
- The fully aggregated gradients on each GPU are made available to all GPUs.

Ring-AllReduce



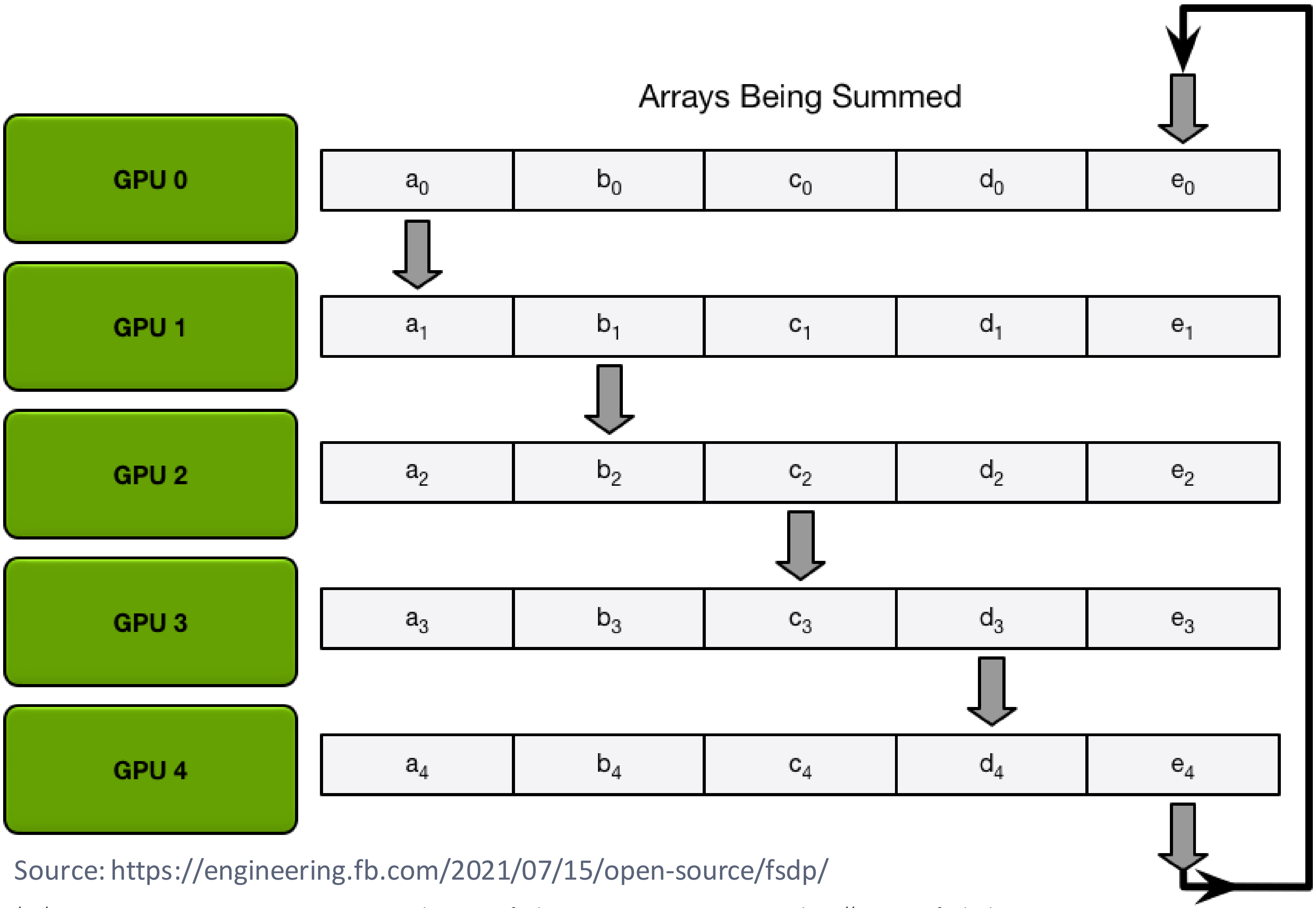
Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

Ring-AllReduce



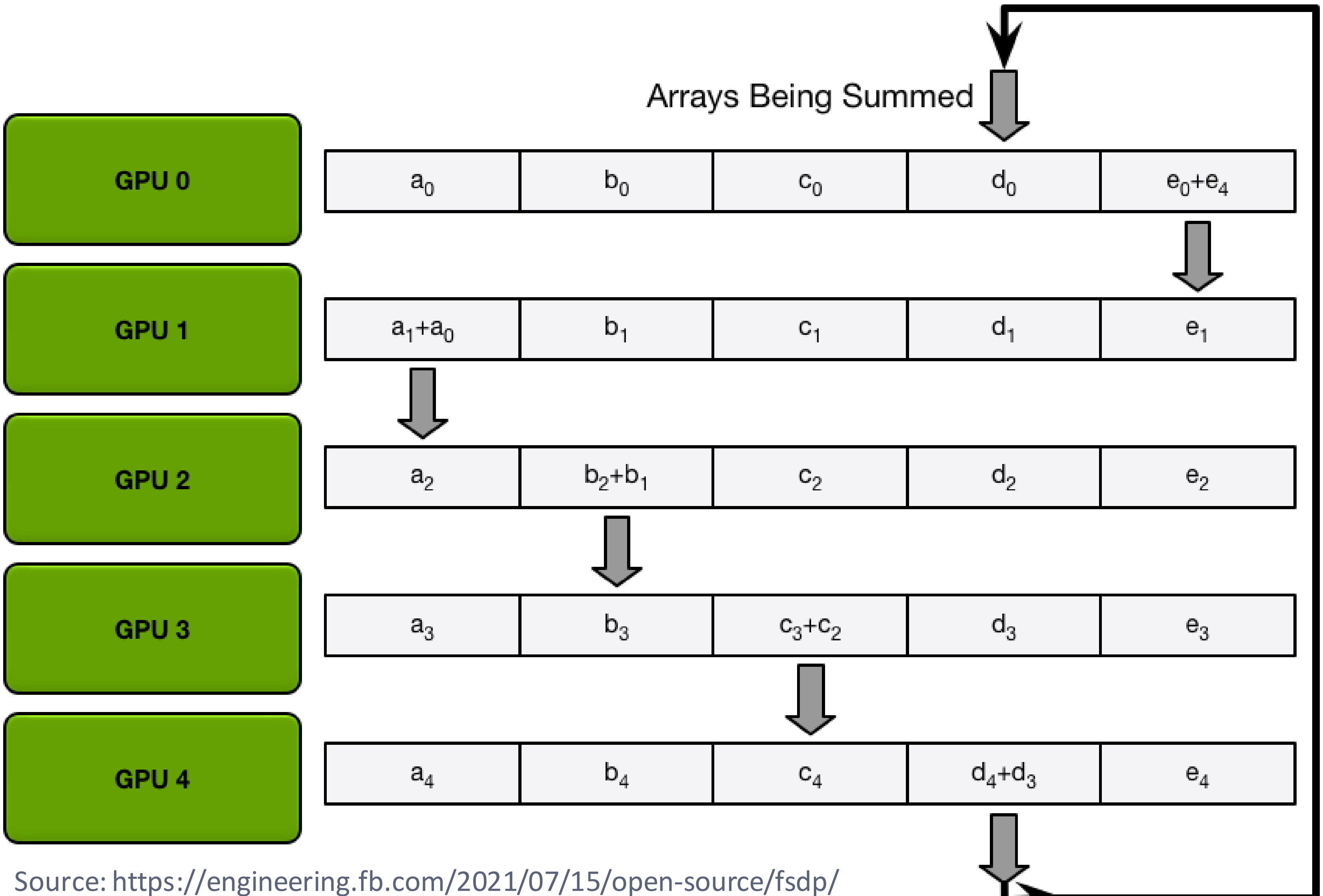
Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

Reduce-Scatter



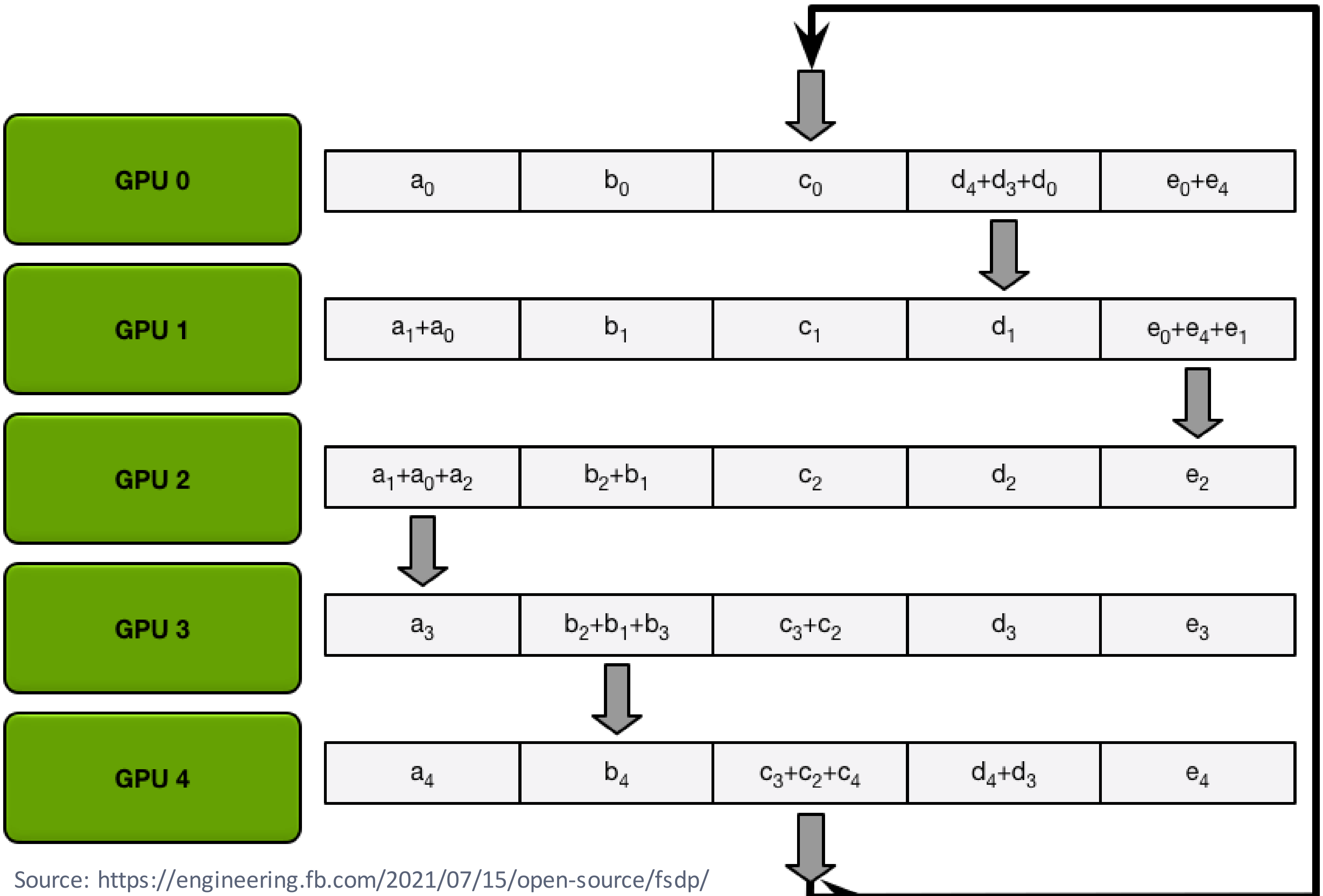
Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

Reduce-Scatter



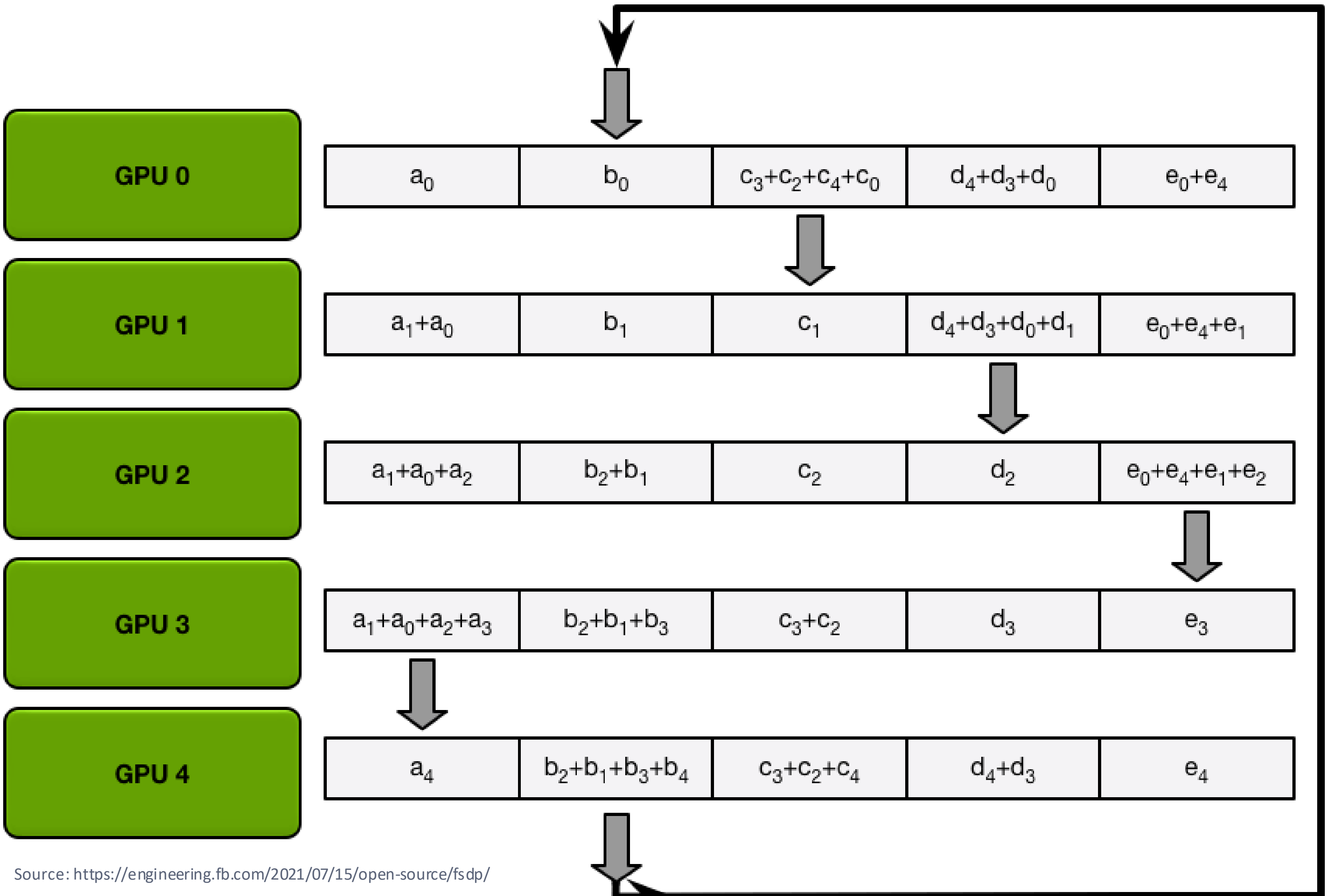
Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

Reduce-Scatter



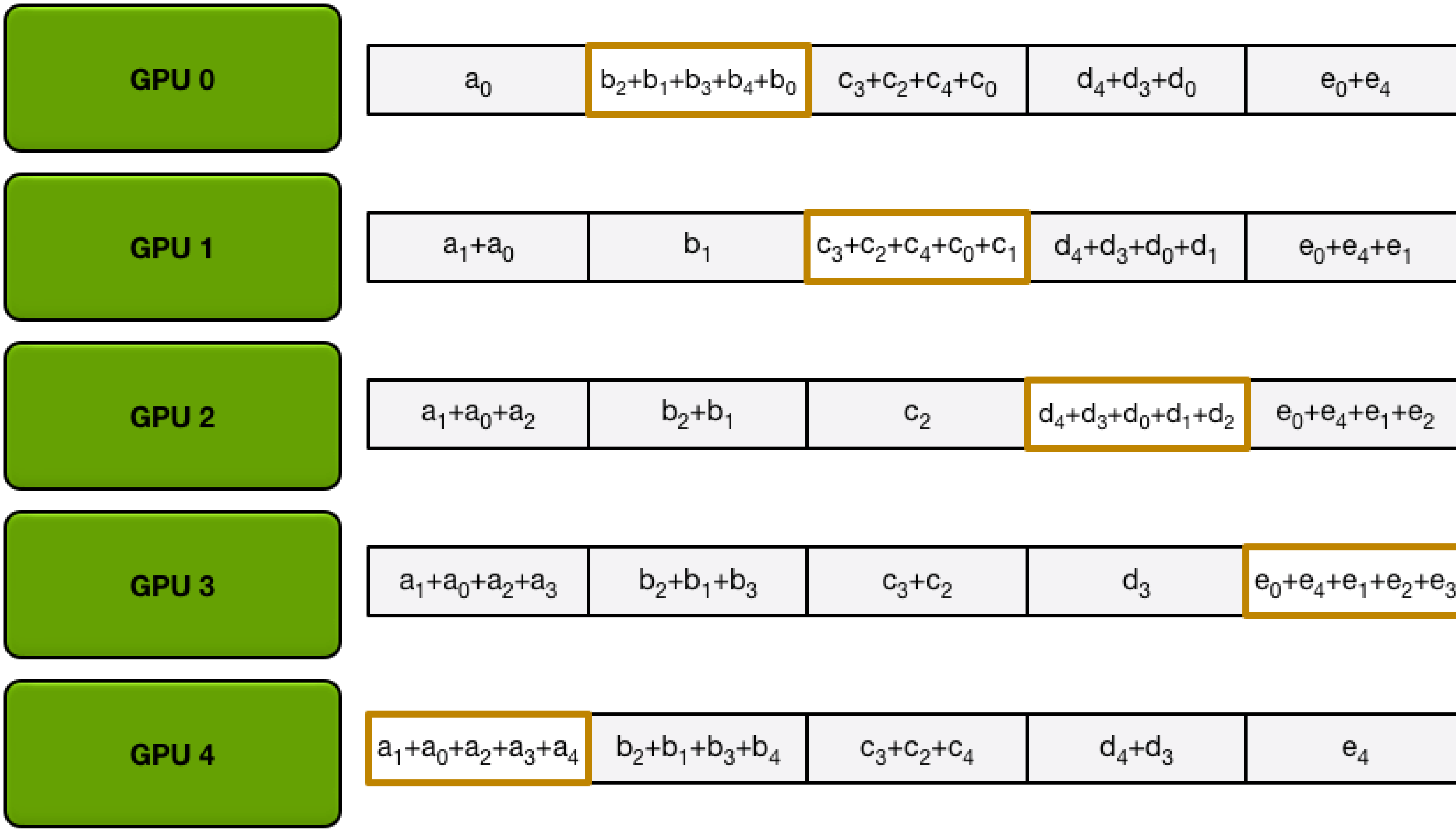
Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

Reduce-Scatter



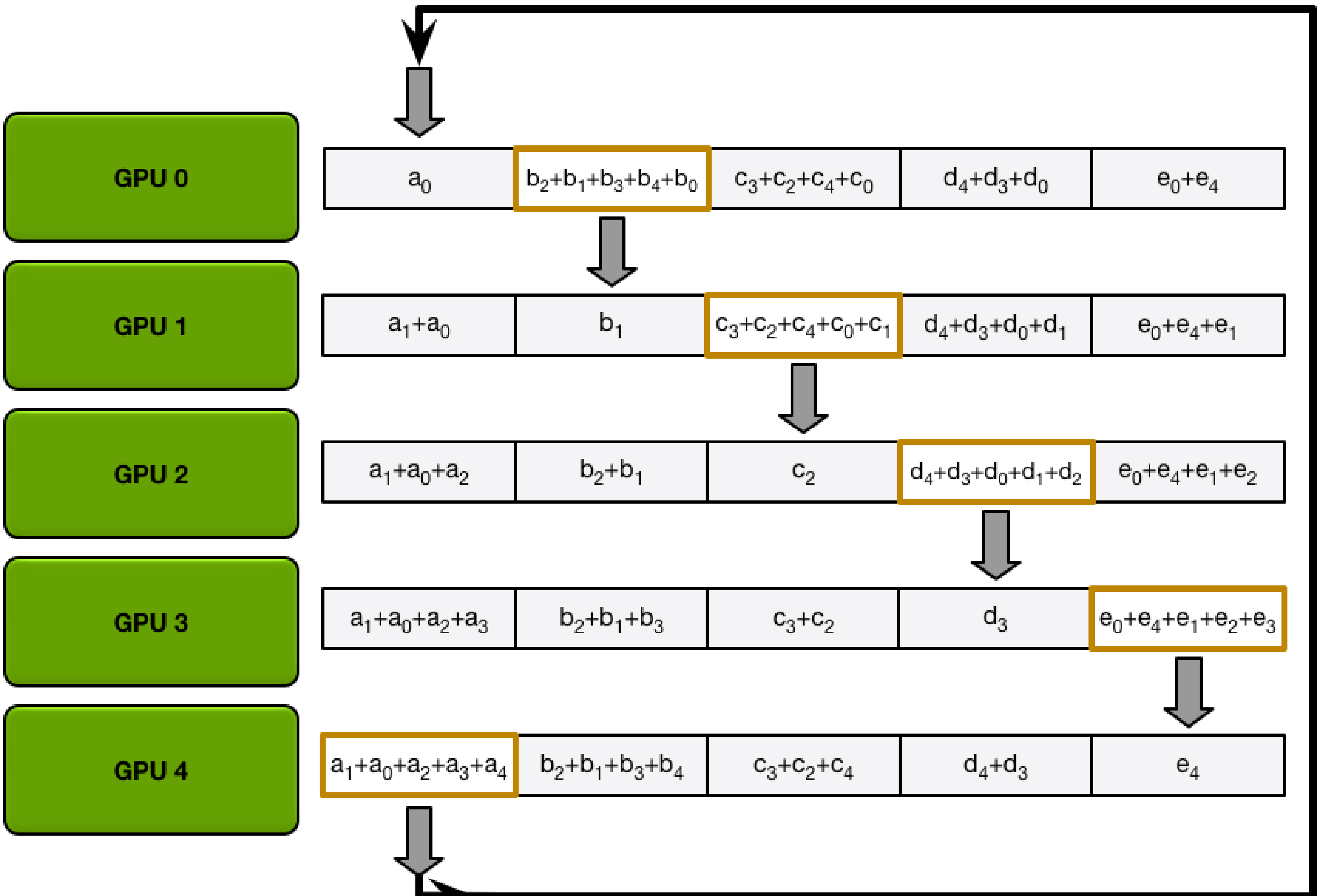
Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

Reduce-Scatter



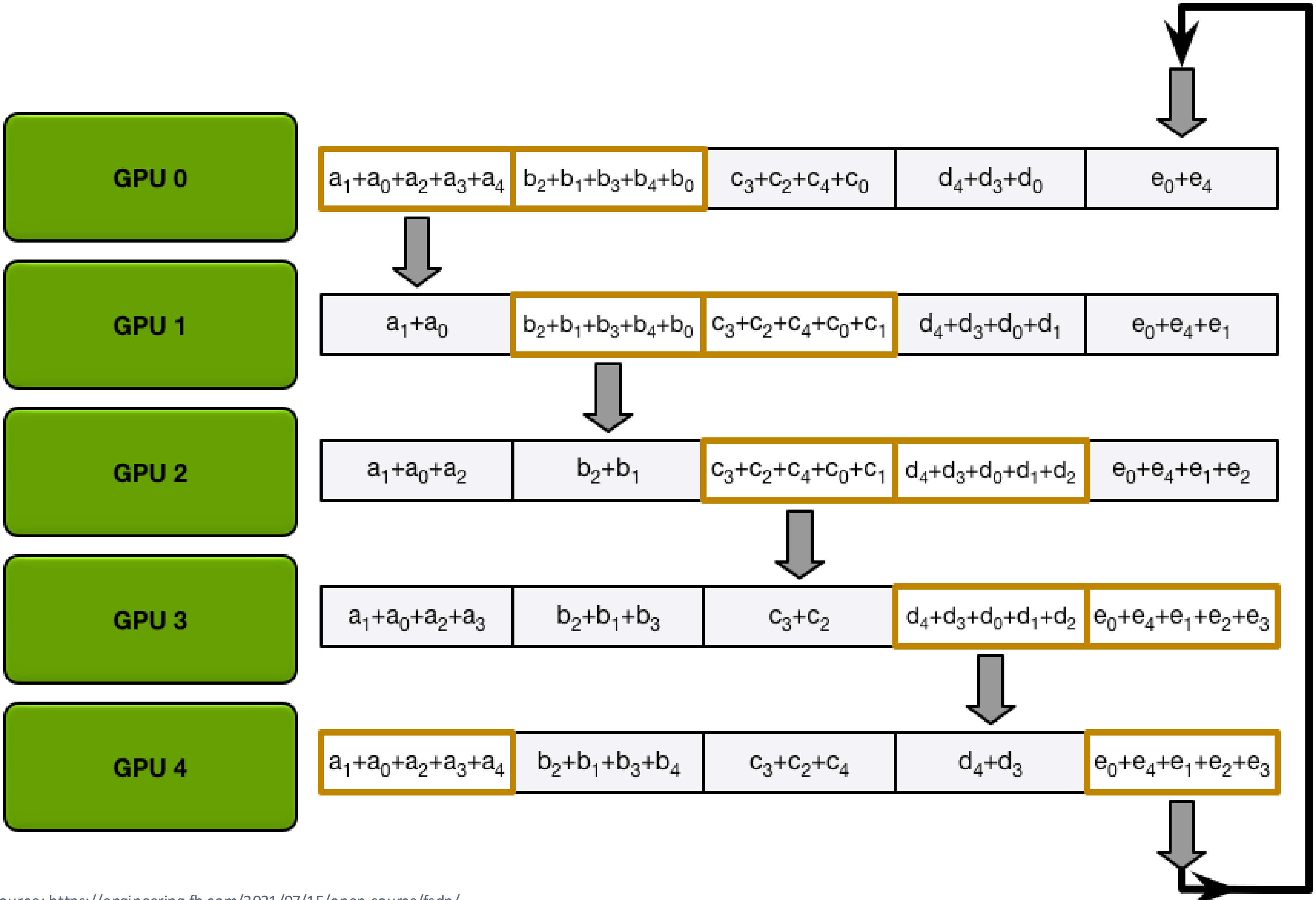
Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

All-gather



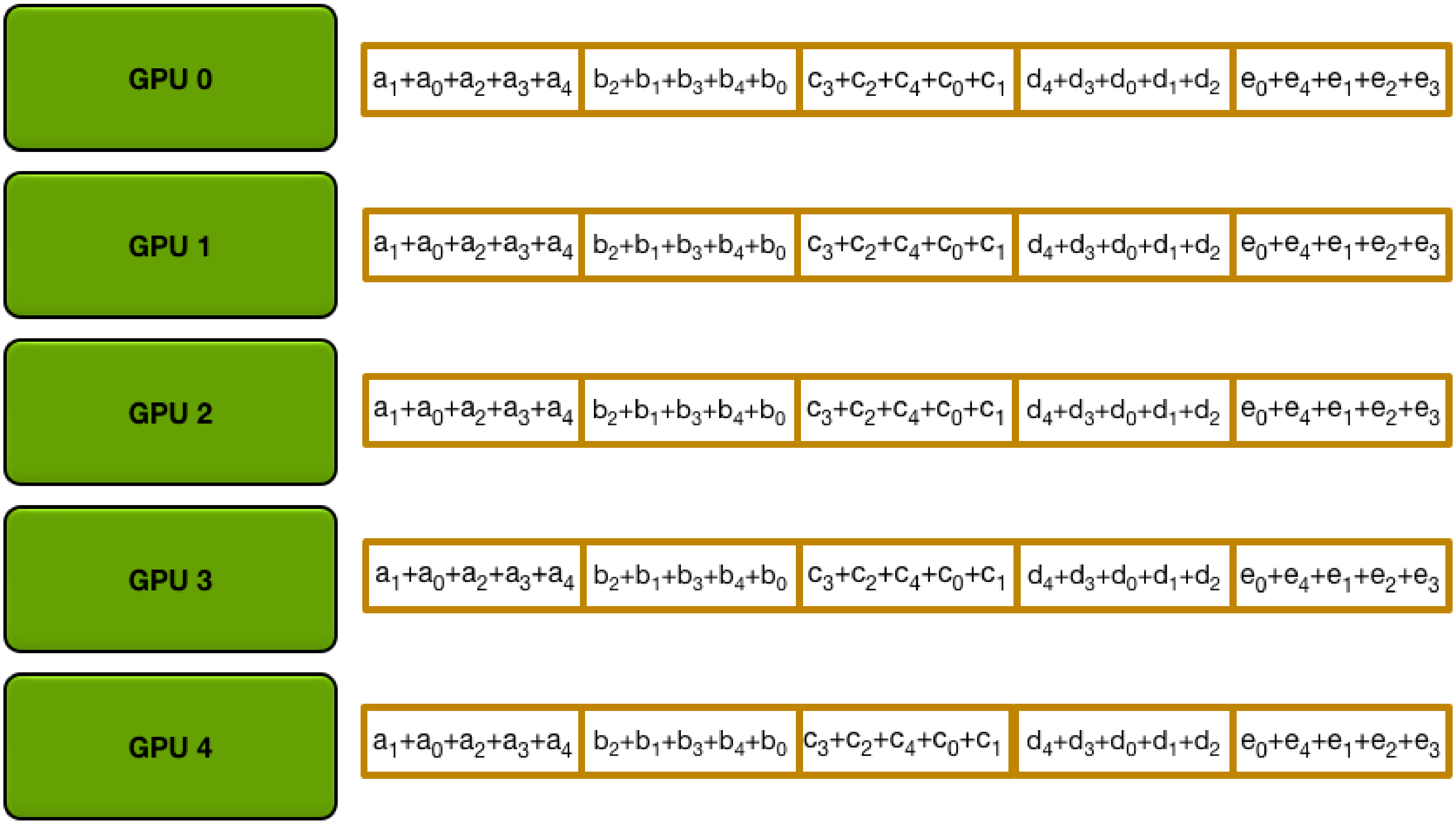
Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

All-gather



Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

All-gather

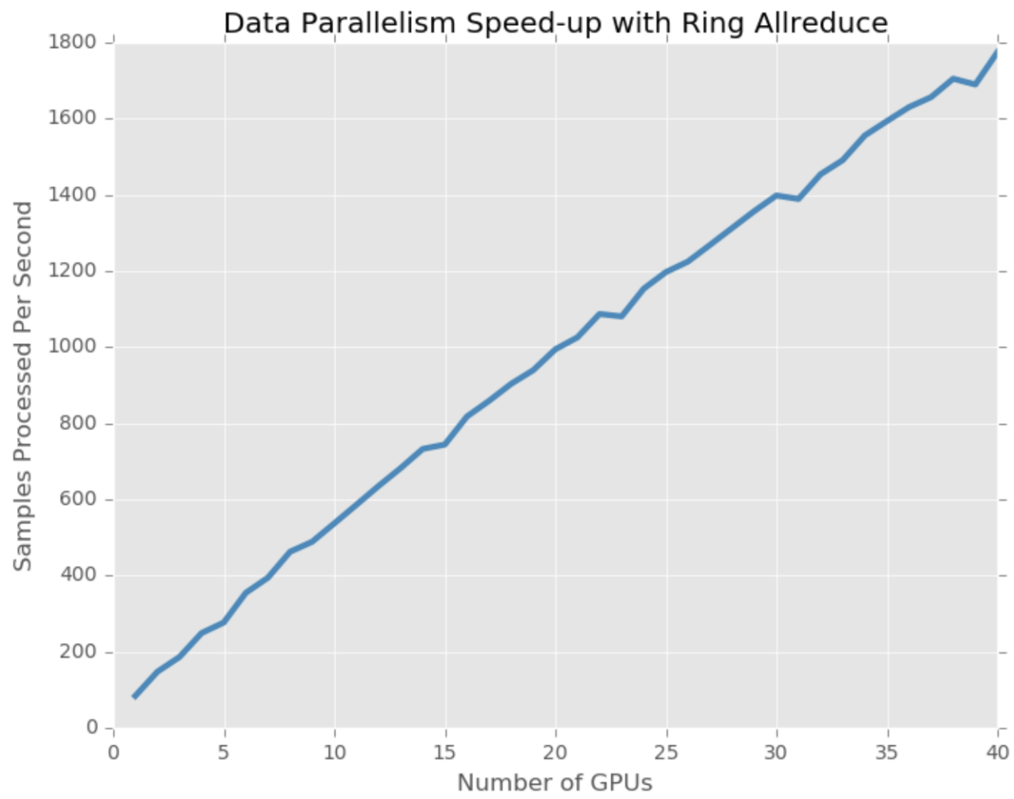


Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

Cost of Ring-AllReduce

- Each GPU sends and receives values K-1 times for reduce-scatter, and K-1 times for the all-gather.
- Each time, the GPUs will send d / K values
- The total cost for every GPU is $\mathbf{2d\,\frac{K-1}{K}}$

Ring-AllReduce in practice



The number of samples processed per second with a 300-million parameter language model scales linearly with the number of GPUs concurrently doing synchronous training.

Source: https://engineering.fb.com/2021/07/15/open-source/fsdp/

Data parallelism recap

- **AllReduce**
  - Cost per GPU: $\boldsymbol{d\,(K-1)}$
- **Parameter Server**
  - Cost per GPU: $\boldsymbol{d}$
  - Cost per Server: $\boldsymbol{d\,(K-1)/M}$
- **Ring-AllReduce**
  - Cost per GPU: $\boldsymbol{2d\,\frac{K-1}{K}}$