🌍 https://hao-ai-lab.github.io/dsc291-s24/

DSC 291: ML Systems
Spring 2024

LLMs

Parallelization

Single-device Optimization

Basics

Recap of Last Week: Memory Optimization

- Checkpointing and rematerialization
  - Limitations: for activations, trade flops
- CPU Swapping
  - Limitations: restricted by dram -> hbm bandwidth
- Quantization and Mixed precision
  - Potential accuracy (ML performance) loss
  - Kernel support cannot catch up

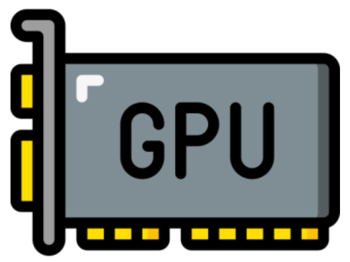
Next 2 weeks: Large-Scale Distributed ML

- **Motivation**
- History
- Parallelism Overview
- Data parallelism
- Model parallelism
  - Inter and intra-op parallelism
- Auto-parallelization

Big Model: The Core Computational Challenge

BERT
340M parameters (680 MB)

<



Device Memory
16 - 40 GB

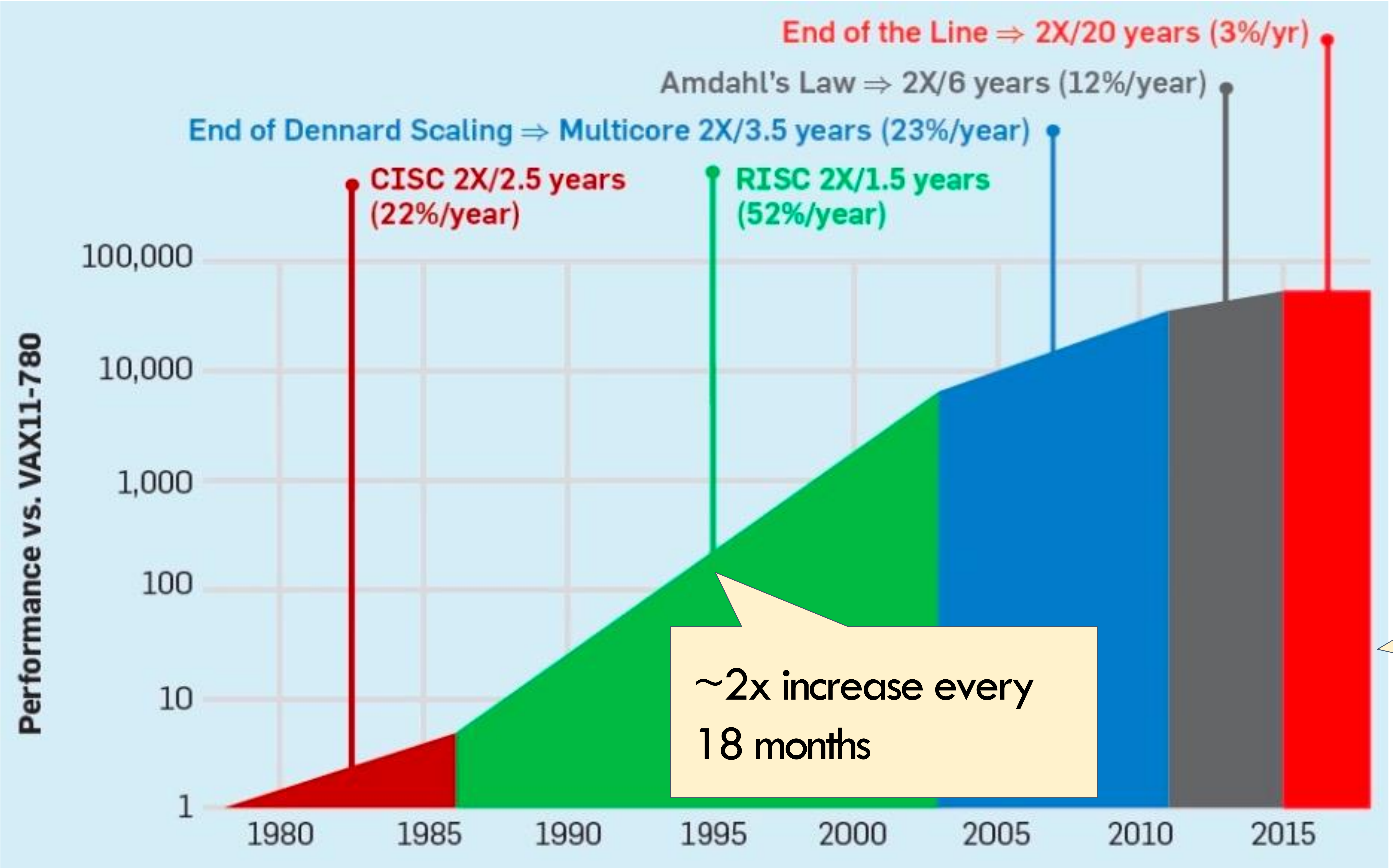
<<

GPT-3
175B parameters (350 GB)

How to train and serve big models?

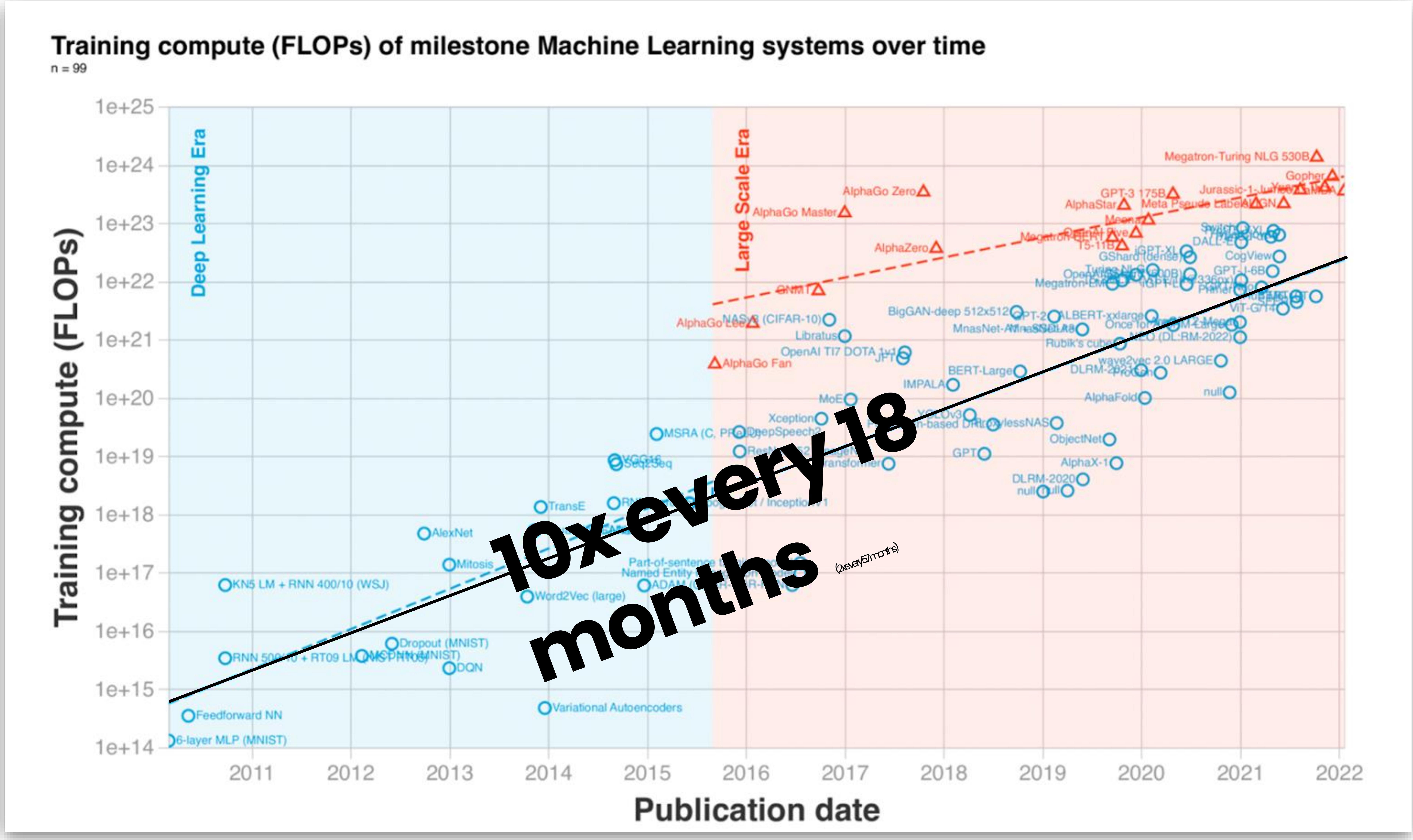
Using parallelization.

Moore's Law coming to an end



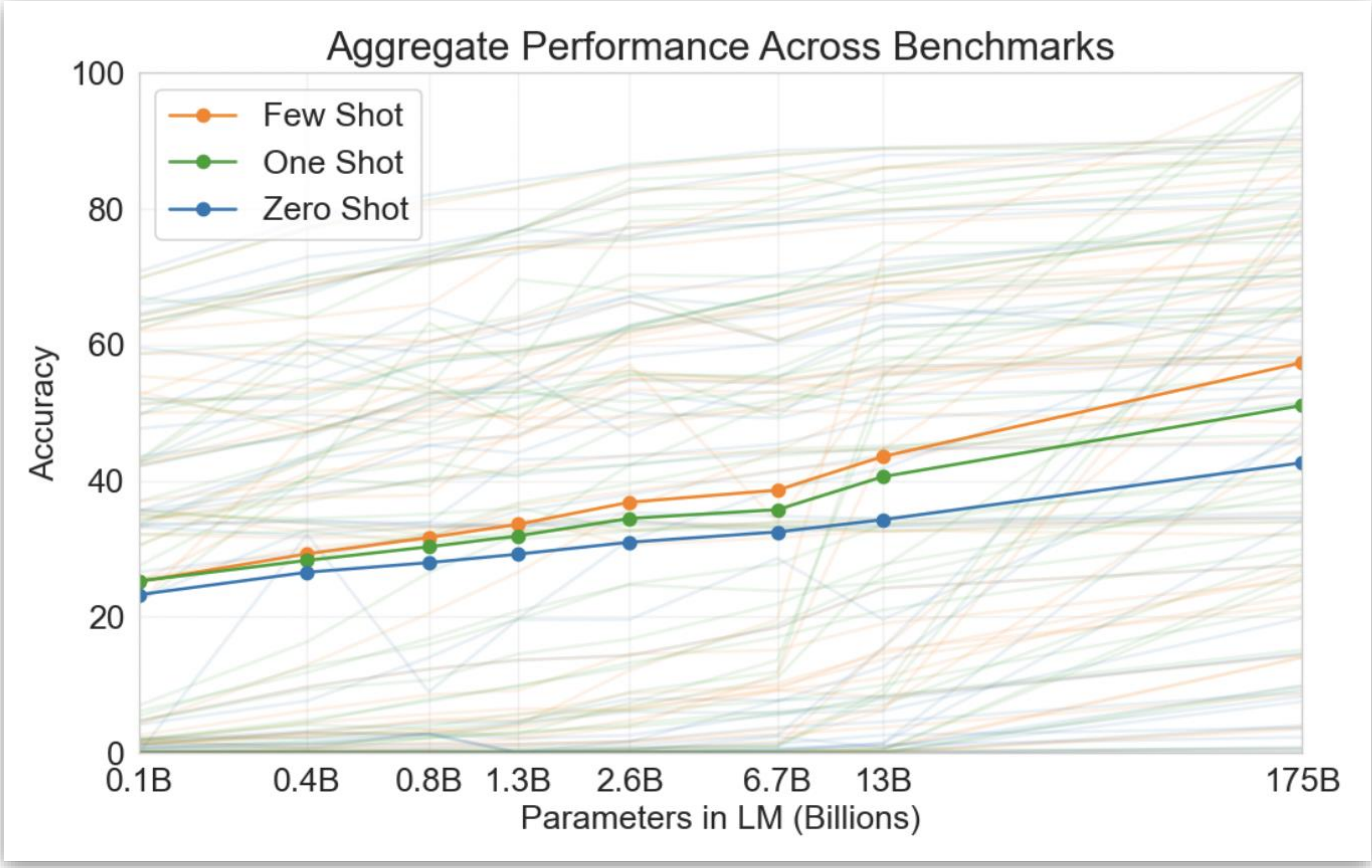
~1.05x increase every 18 months

Meanwhile…. ML demands are exploding



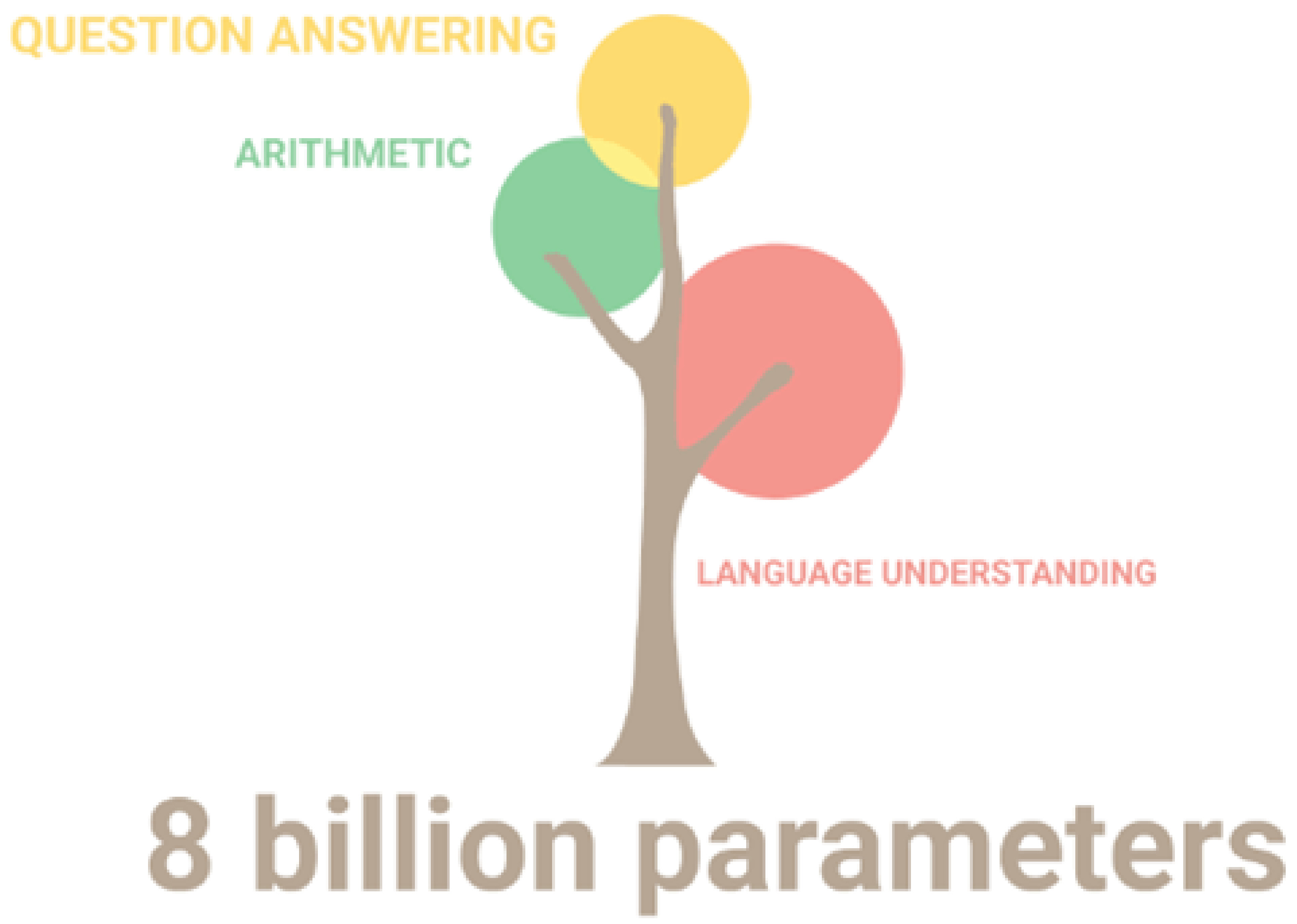
“Compute trends across three eras of machine learning”, J. Sevilla, https://ar5iv.labs.arxiv.org/html/2202.05924

Why? Bigger model, better accuracy



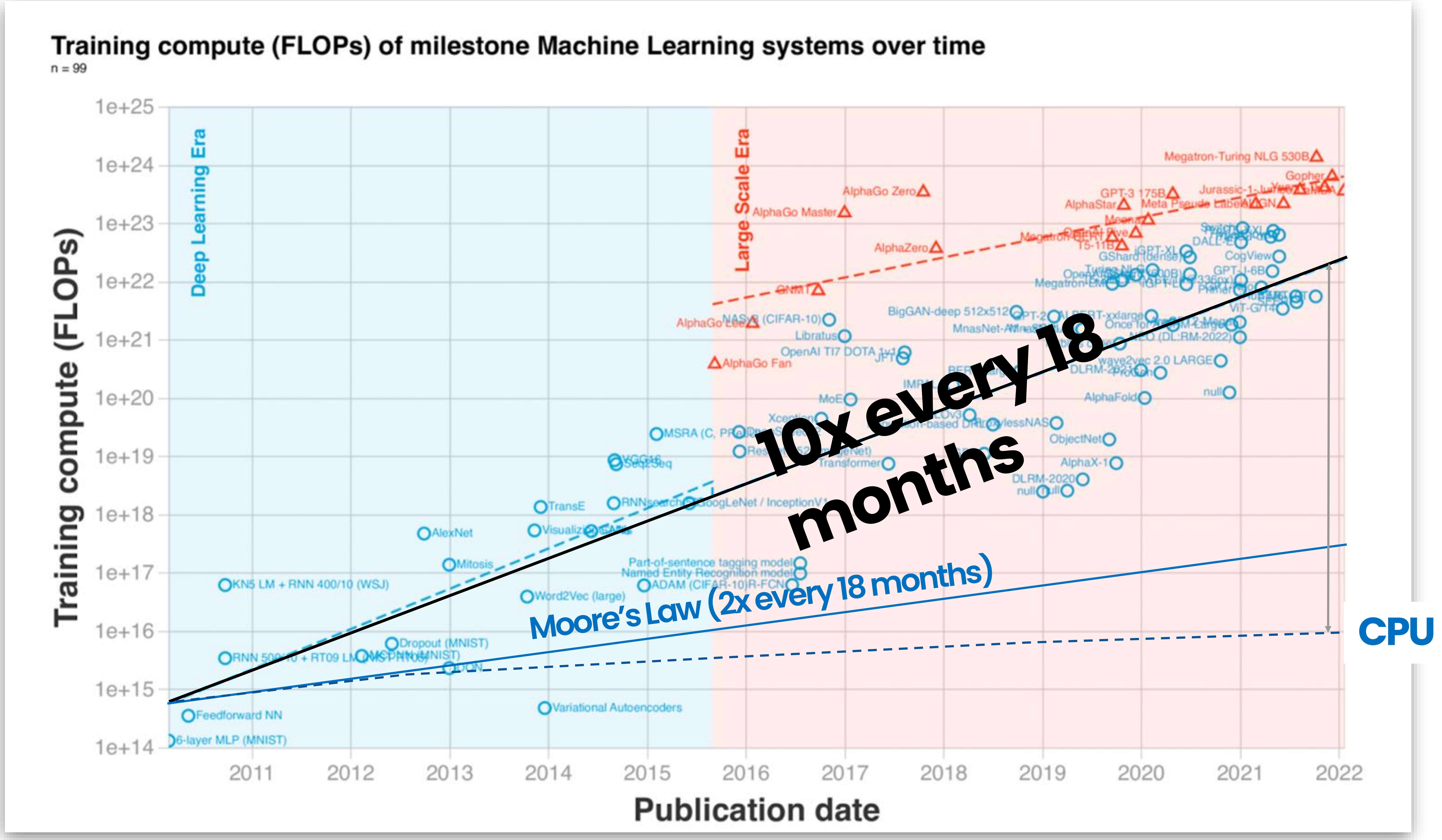
"Language Models are Few-Shot Learners", T. B. Brown et al., https://arxiv.org/pdf/2005.14165.pdf

Why? Emergence of foundation models



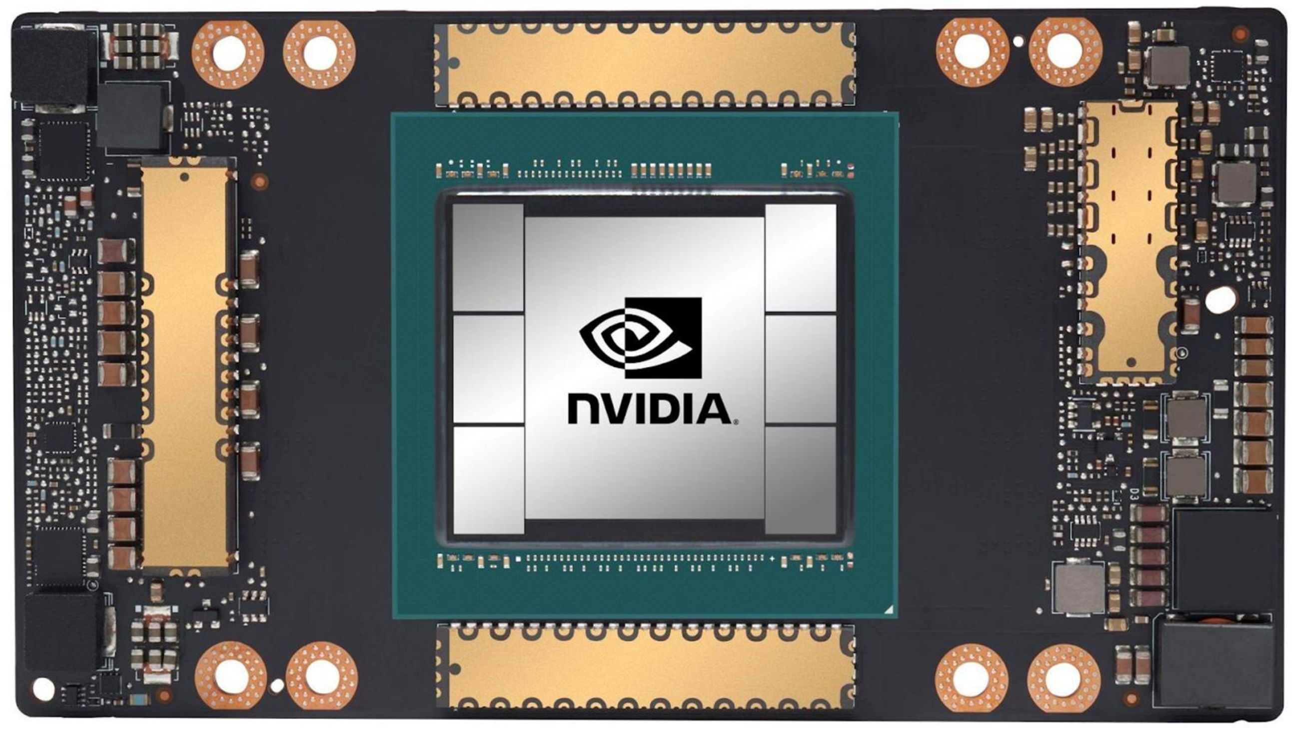
"Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance",
S Narang, A Chowdhery et al, https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html

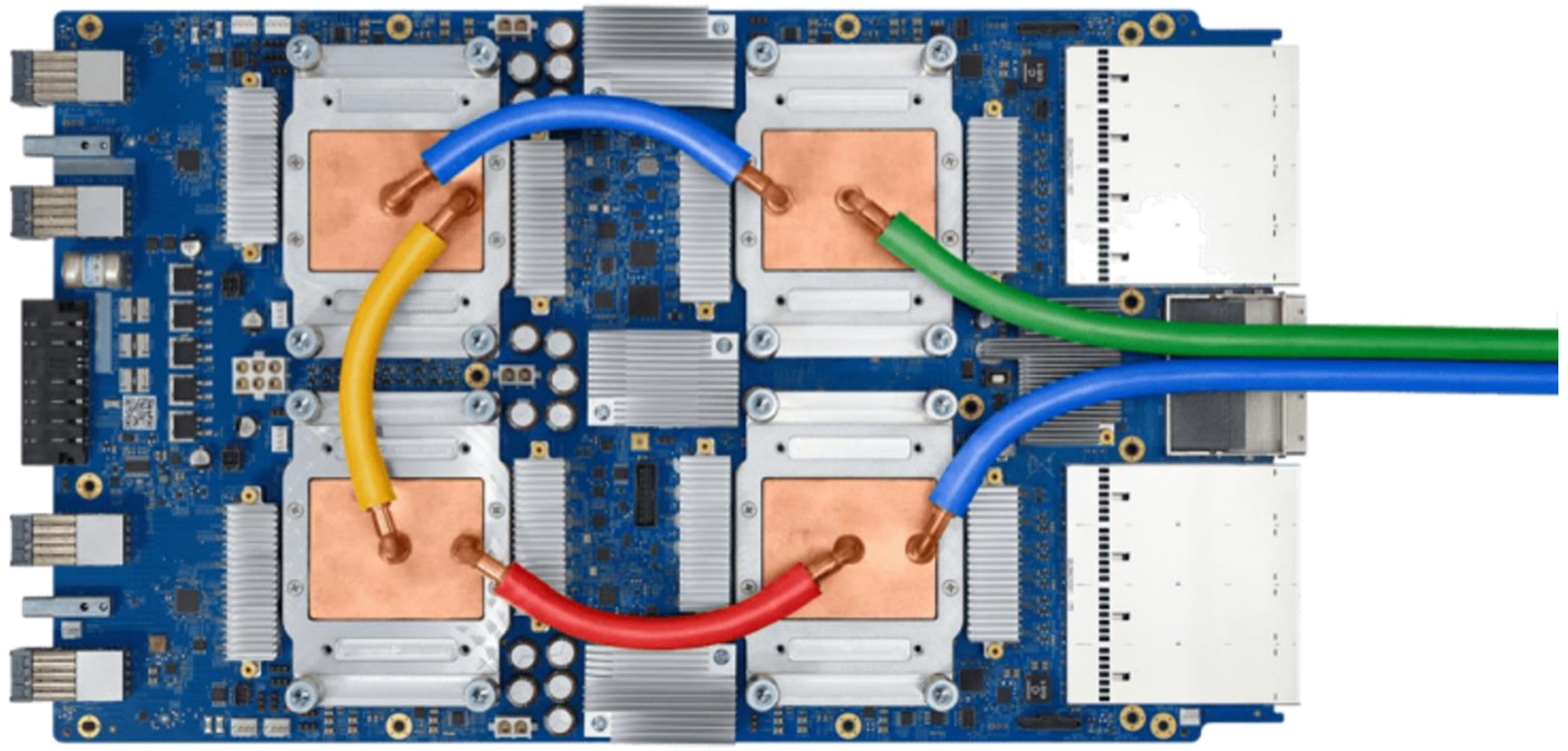
Growing gap between demand and supply

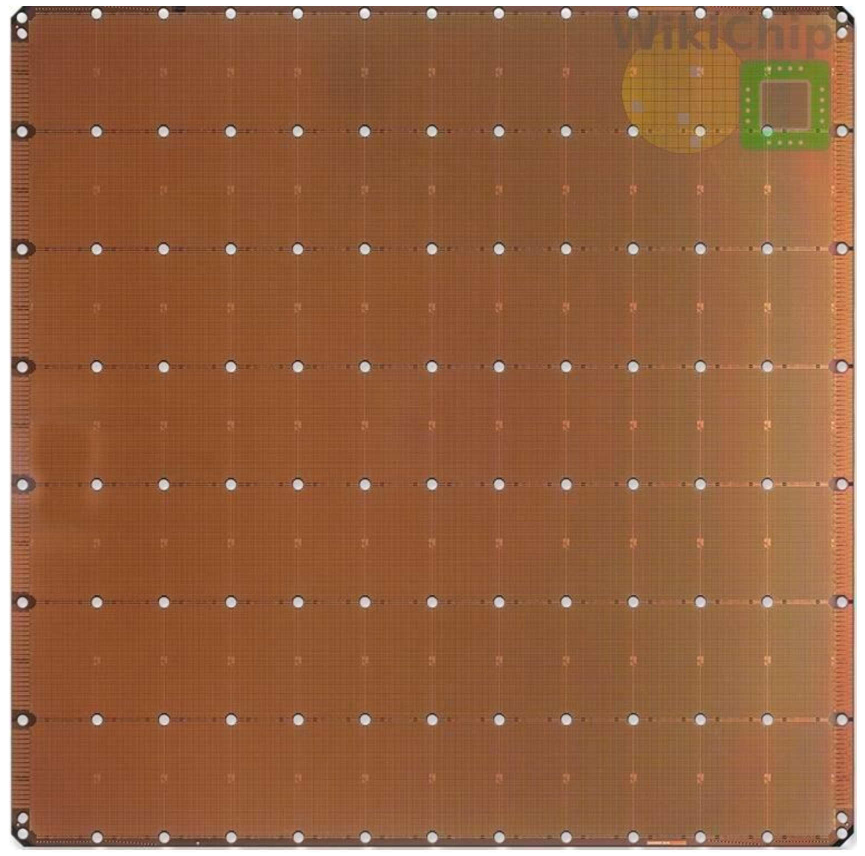


"Compute trends across three eras of machine learning", J. Sevilla, https://ar5iv.labs.arxiv.org/html/2202.05924

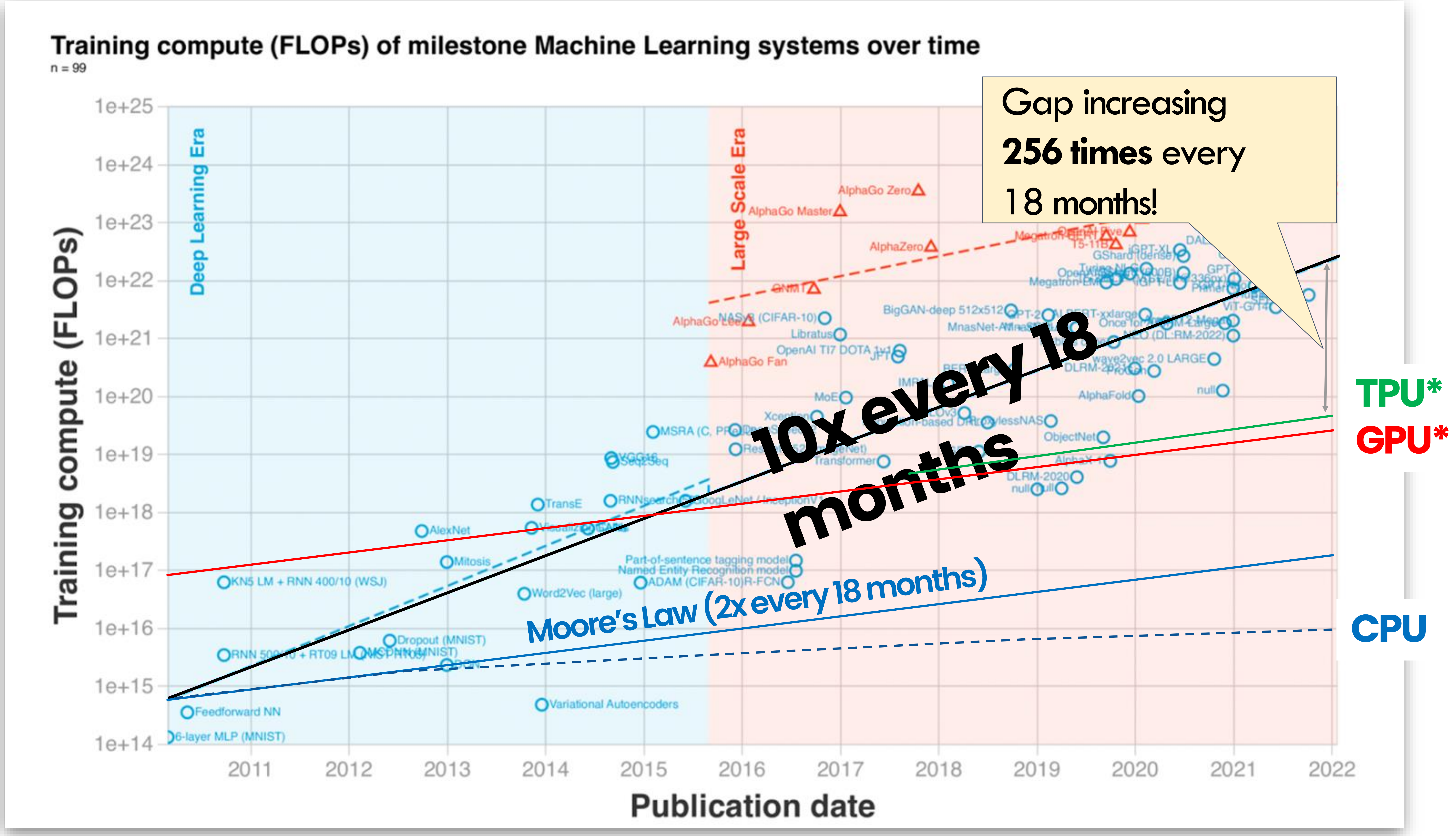
What about specialized hardware?







Specialized hardware not good enough



"Compute trends across three eras of machine learning", J. Sevilla, https://ar5iv.labs.arxiv.org/html/2202.05924

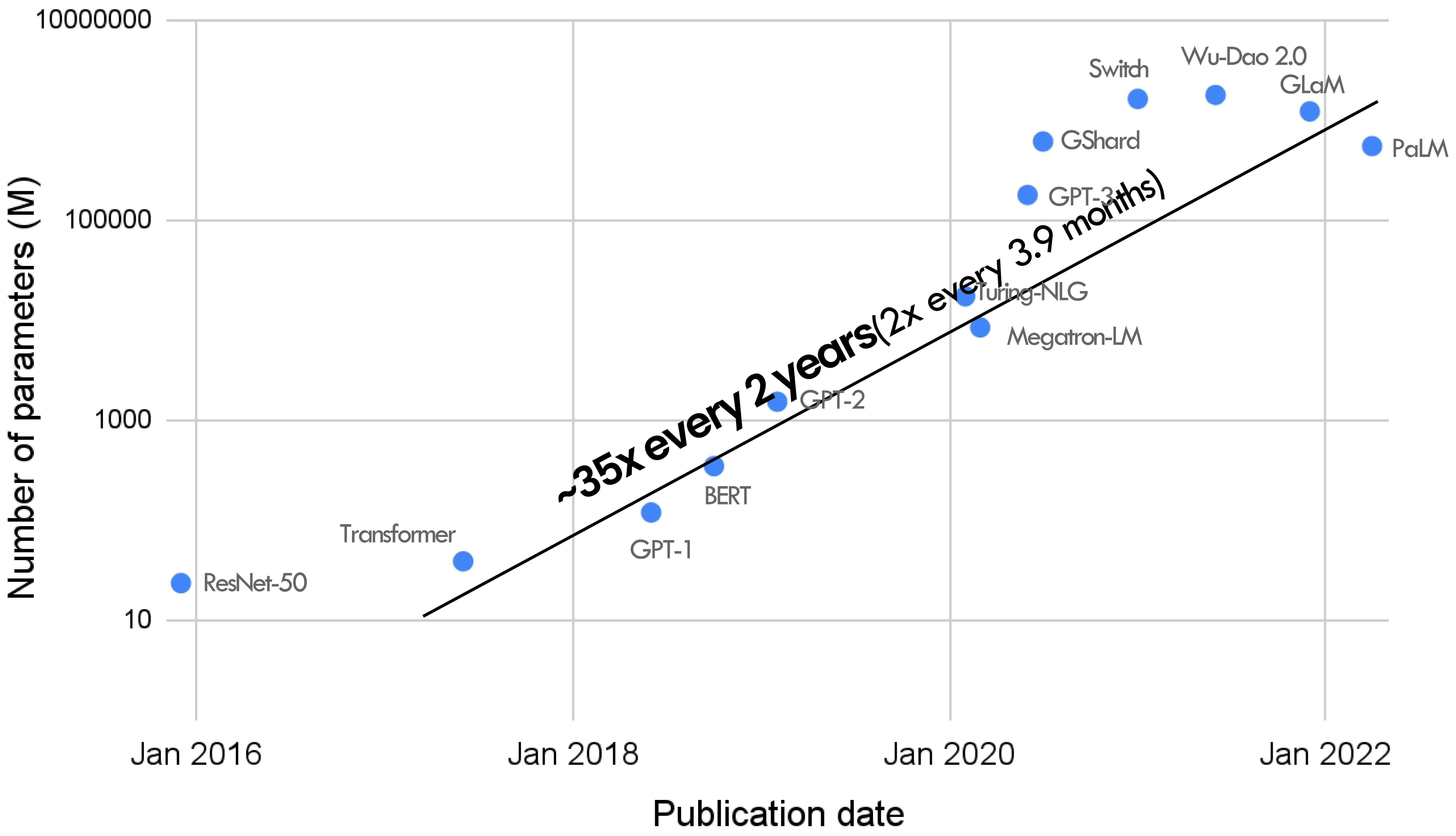
Even if model sizes would stop growing…

… it would take decades for specialized hardware to catch up!

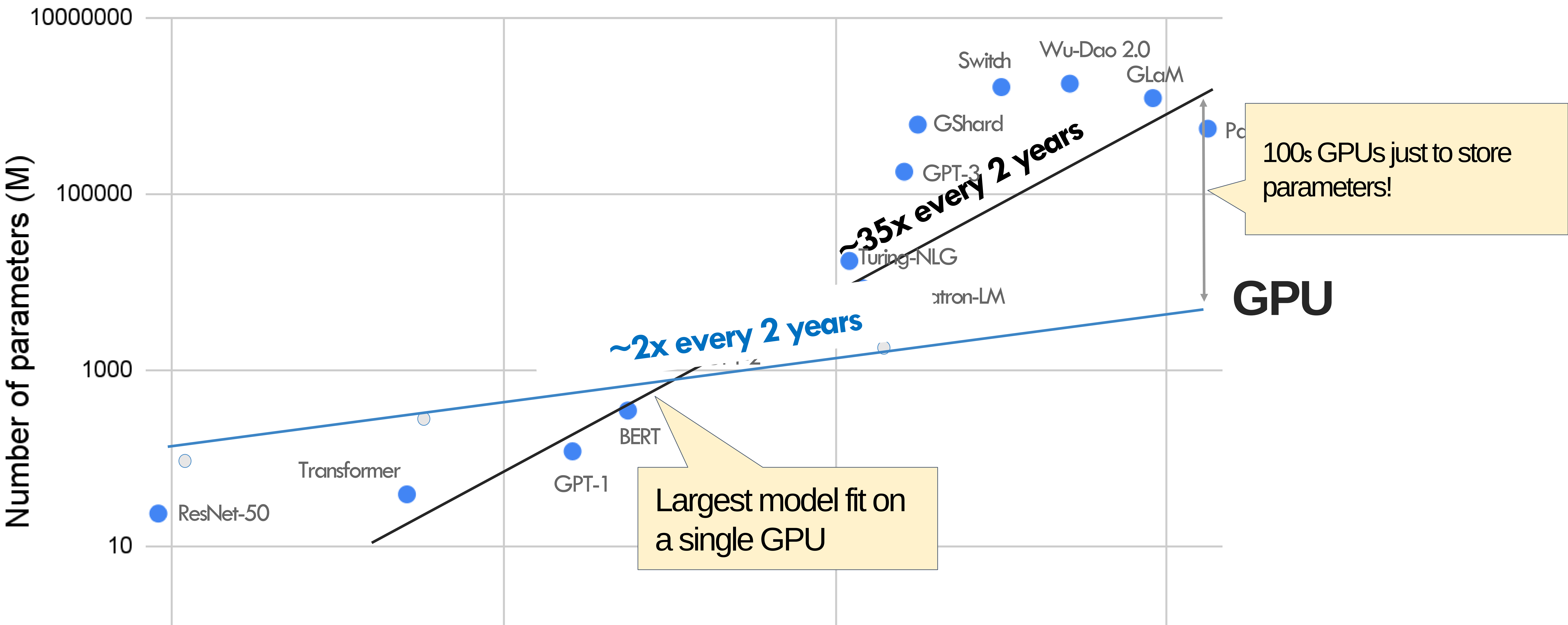
Example:

- Google's PaLM takes 6144 TPU v4 to train
- Assuming doubling performance every 18x month it would take **~19 years** to train it on a single chip

Not only compute, but memory

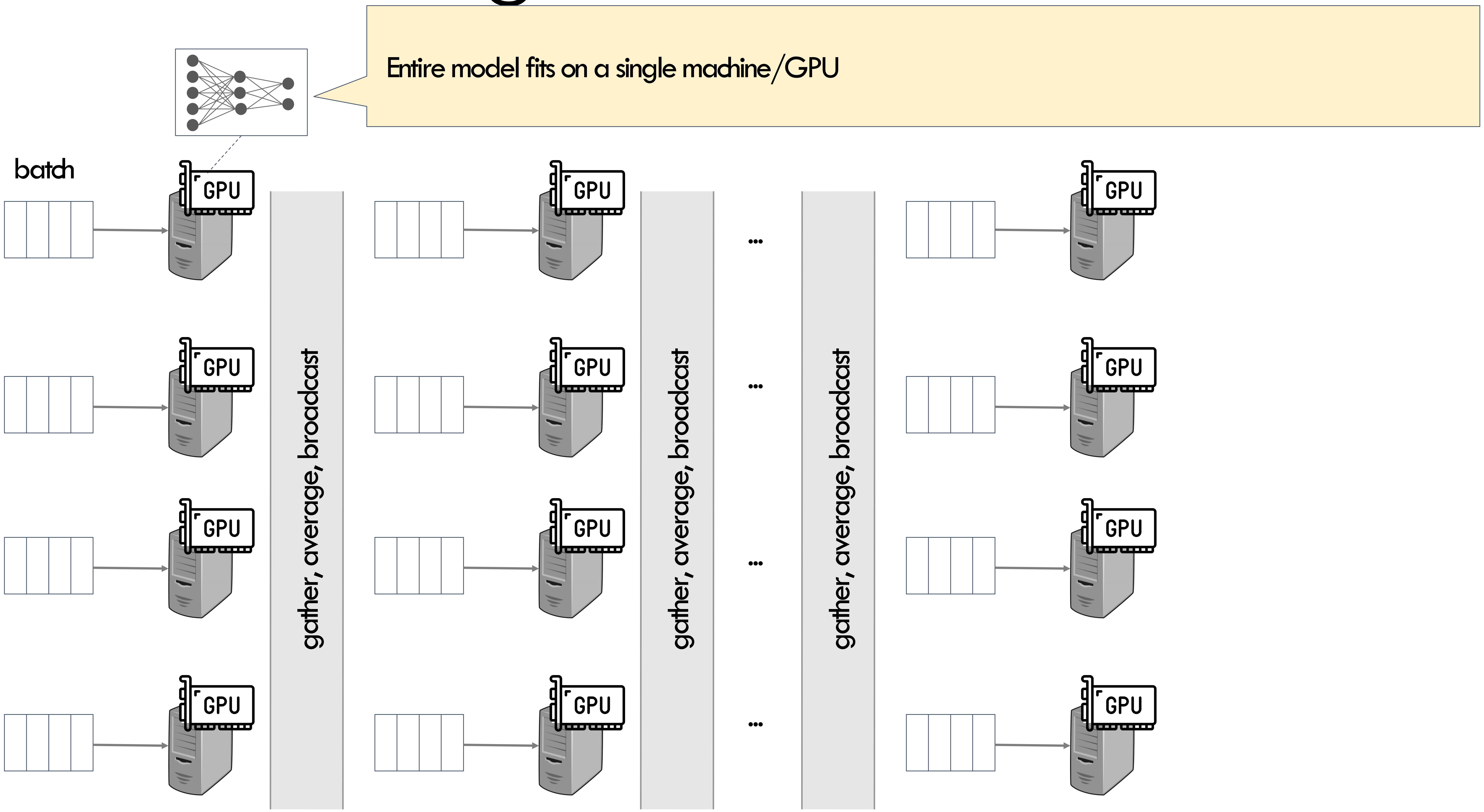


Growing gap between memory demand and supply



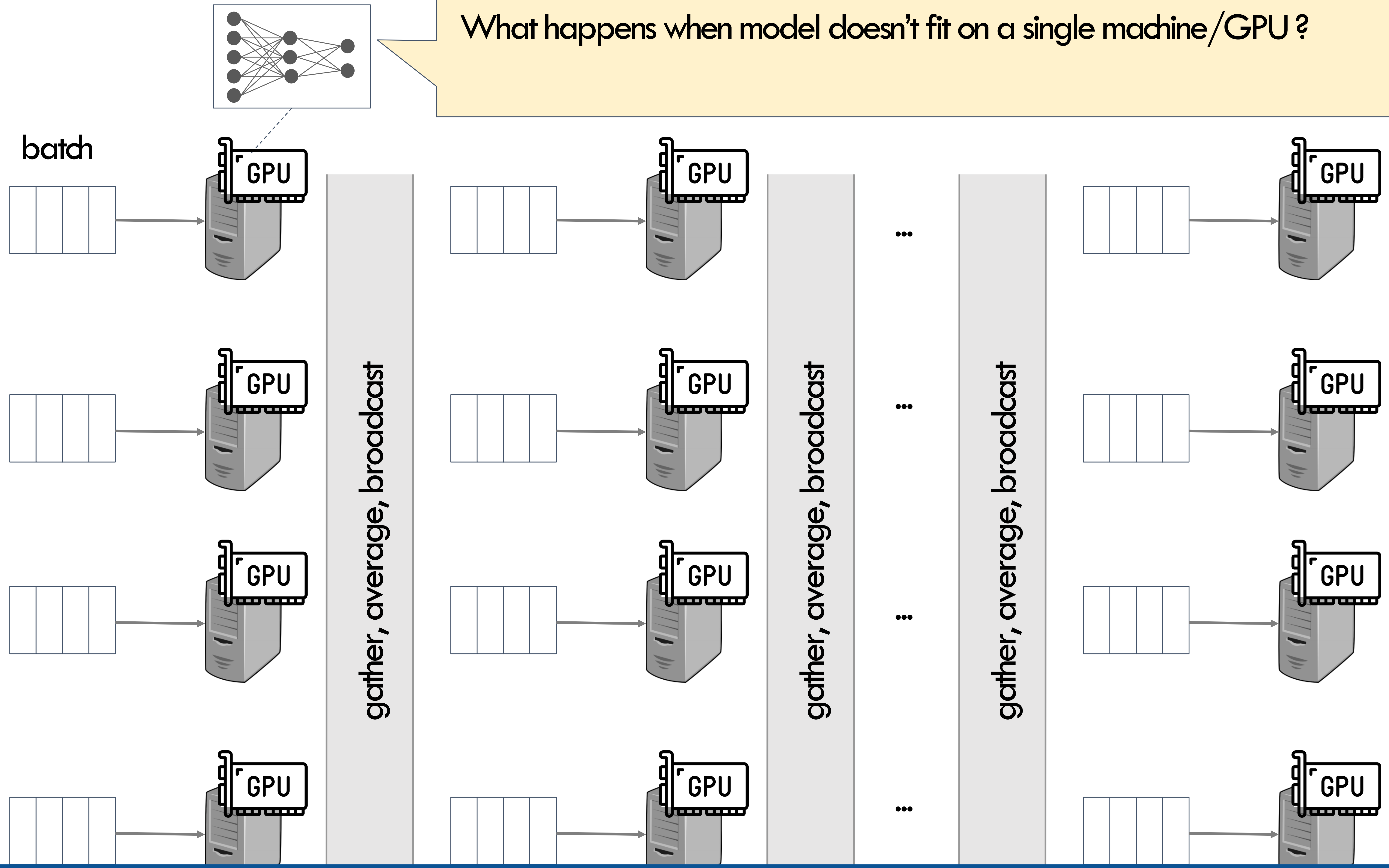
No way out but to parallelize these workloads !

Data Parallel Training



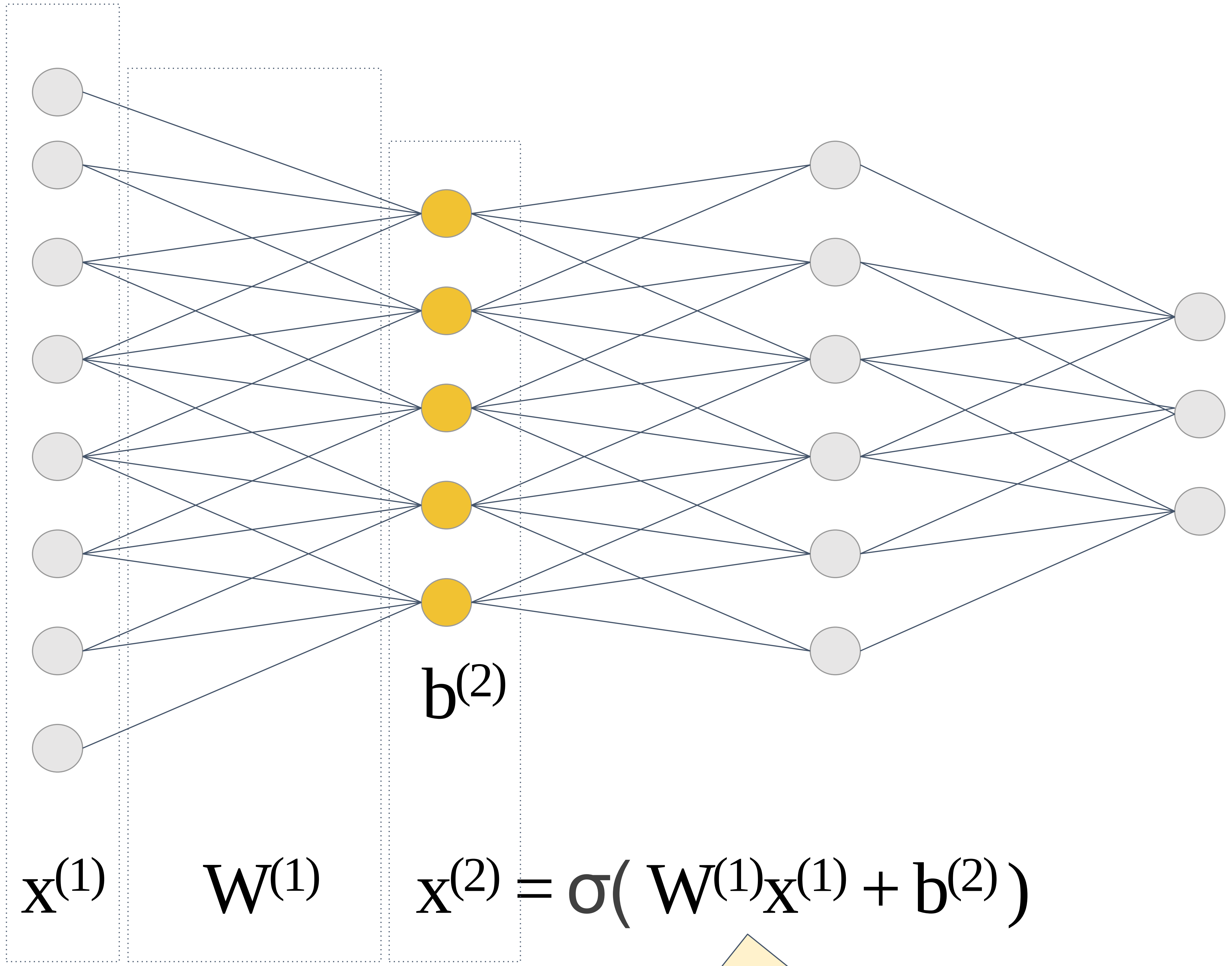
Data Parallel Training

What happens when model doesn't fit on a single machine/GPU ?



Need to parallelize the model itself

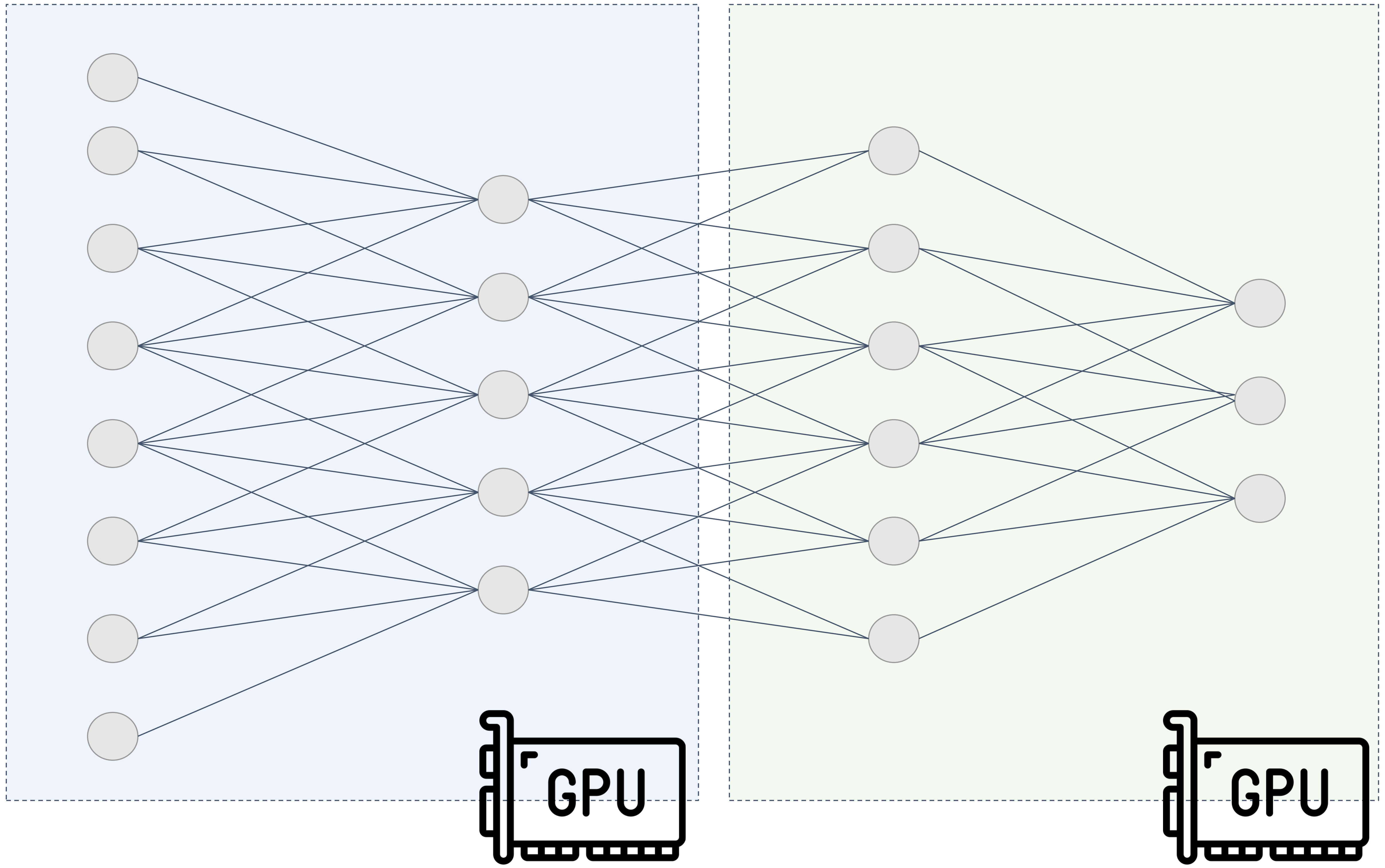
Need do parallelize the model, but how?



$x^{(2)} = \sigma(\, W^{(1)} x^{(1)} + b^{(2)} \,)$

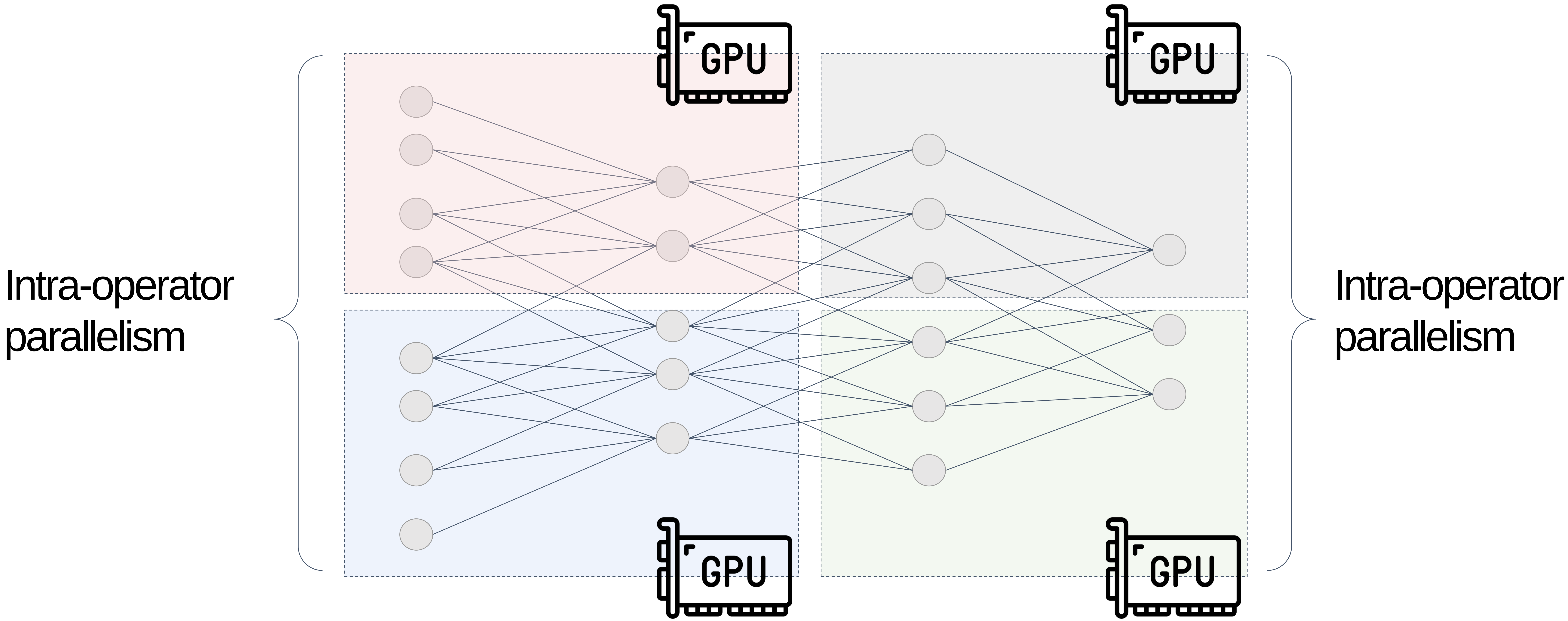
Tensor **operator**

Inter-operator parallelism



- Pipeline execution on both forward and backward paths
- GPUs can be on the same machine or **different** machines

Intra-operator parallelism



Where we are

- Motivation
- **History**
- Parallelism Overview
- Data Parallelism
- Model parallelism
  - Inter and intra-op parallelism
- Auto-parallelization

Distributed DL History in 10 mins

2012

Reflections of DL parallelization in early DL papers

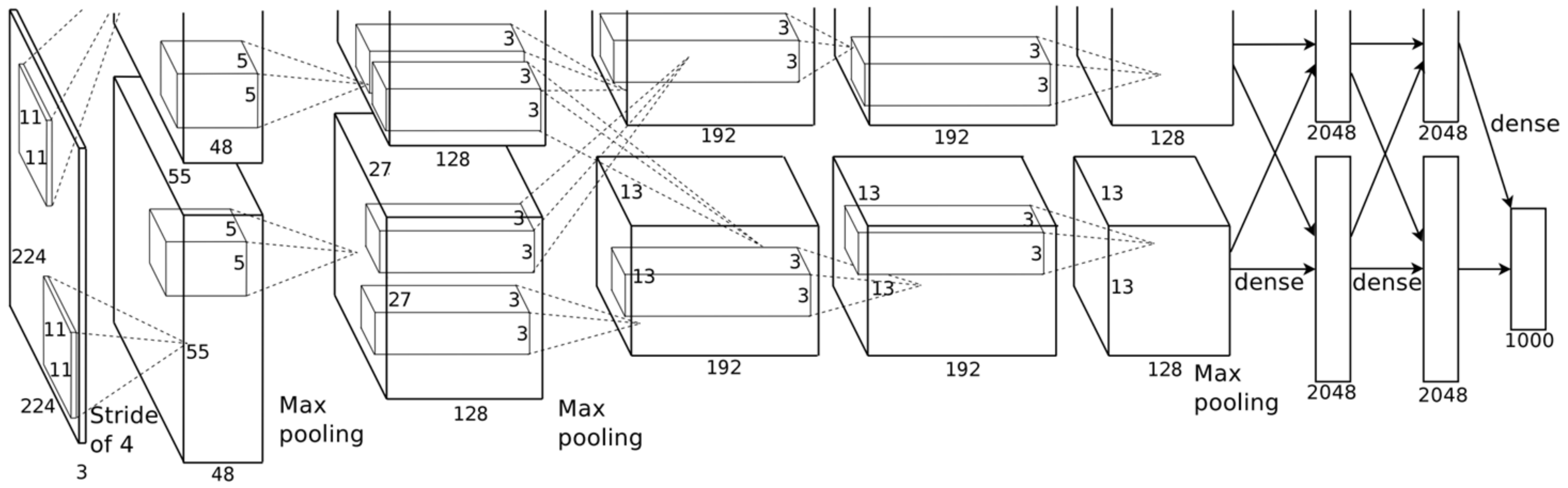


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input 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.

Figure from AlexNet
[Krizhevsky et al., NeurIPS 2012],
[Krizhevsky et al., preprint, 2014]

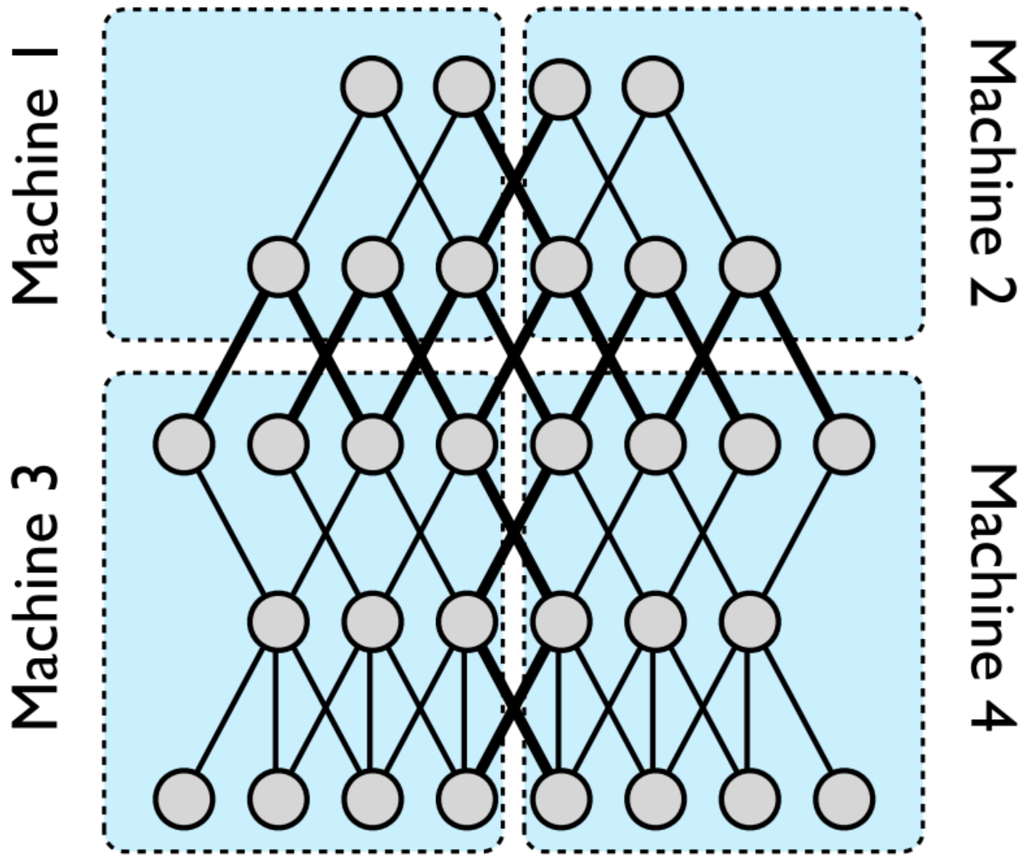


Figure from DistBelief
[Dean et al., NeurIPS 2012]

Data Parallelism with Parameter Server

2012

2016

Focus: Data parallelism with **Parameter Server**

Asynchrony: update every N iters instead of 1

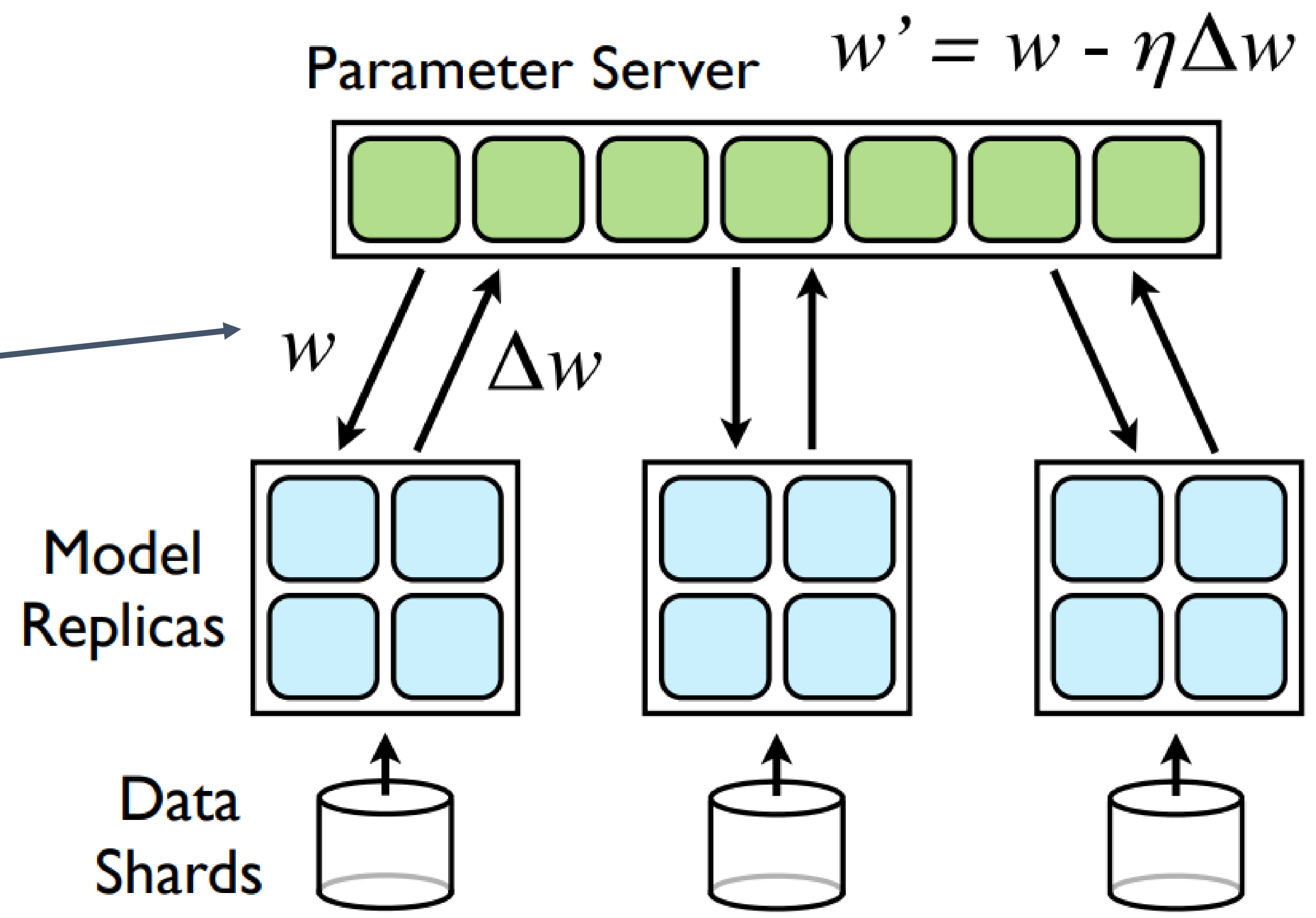


Figure from DistBelief
[Dean et al., NeurIPS 2012]

Various implementations of parameter servers

- DistBelief [Dean et al., NeurIPS 2012]
- Parameter server [Li et al., NeurIPS 2012], [Li et al., OSDI 2014]
- Bosen [Wei et al., SoCC 2015]
- GeePS [Cui et al., Eurosys 2016], Poseidon [Zhang et al., ATC 2017]

Data Parallelism with All-reduce

2012

2016

```python
import torch.nn.parallel as dist
from torch.nn.parallel import DistributedDataParallel as DDP

dist.init_process_group("nccl", rank=rank, world_size=world_size)
ddp_model = DDP(Model(), device_ids=[rank])

for batch in data_loader:
   loss = train_step(ddp_model, batch)
```

Sergeev et al., "Horovod: fast and easy distributed deep learning in TensorFlow". *Preprint 2018.*
Li et al., "PyTorch Distributed: Experiences on Accelerating Data Parallel Training". VLDB 2020.

Data Parallelism with All-reduce

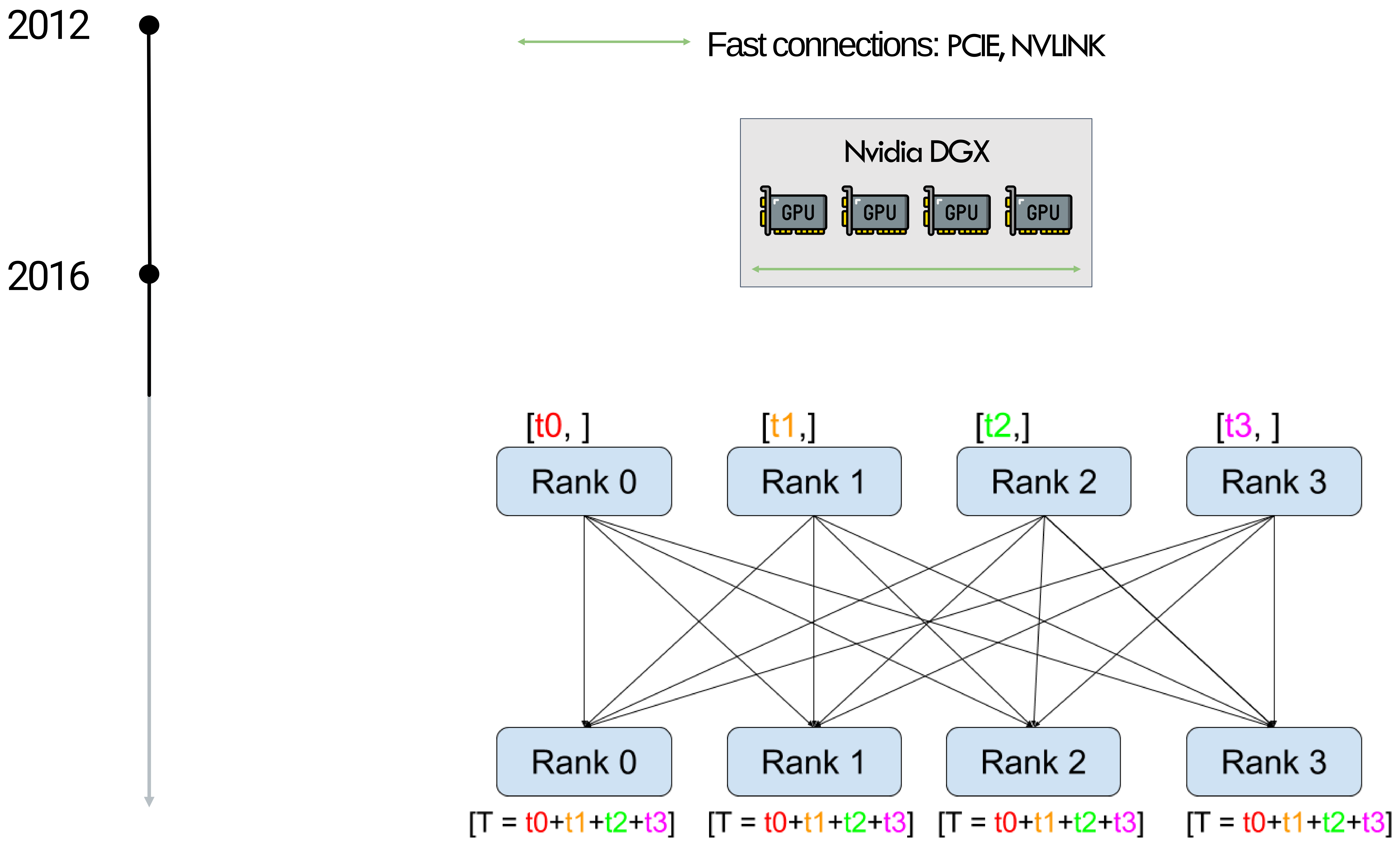


Figure from PyTorch Tutorials

Computational Graph and Placement

2012

2016

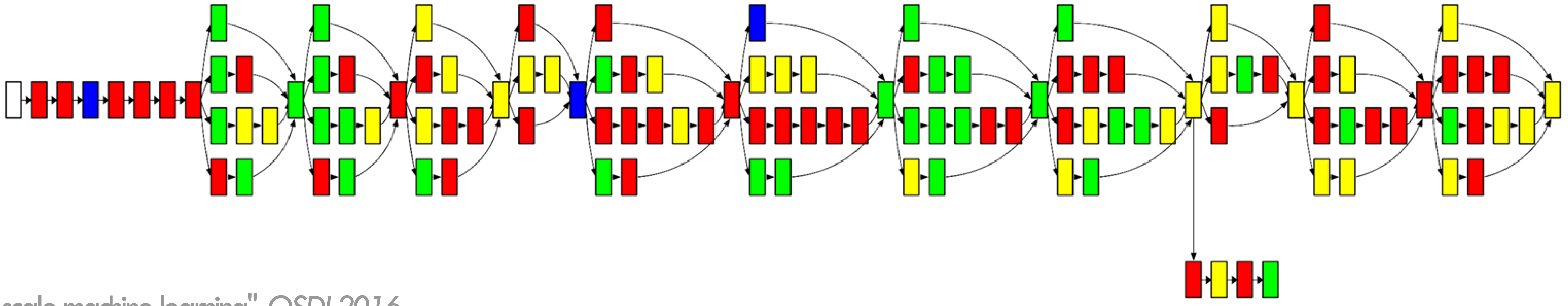
2018

TensorFlow: DL computation as a dataflow graph

```
import tensorflow as tf

c = []
for gpu in gpus:
  with tf.device(gpu.name):
    a = tf.constant([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]])
    b = tf.constant([[1.0, 2.0], [3.0, 4.0], [5.0, 6.0]])
    c.append(tf.matmul(a, b))
with tf.device('/CPU:0'):
  matmul_sum = tf.add_n(c)
```

Figure from [Mirhoseini et al., ICML 2017]



Abadi et al., "TensorFlow: A system for large-scale machine learning". *OSDI 2016.*

Model Parallelism Renaissance

2012

2016

2018

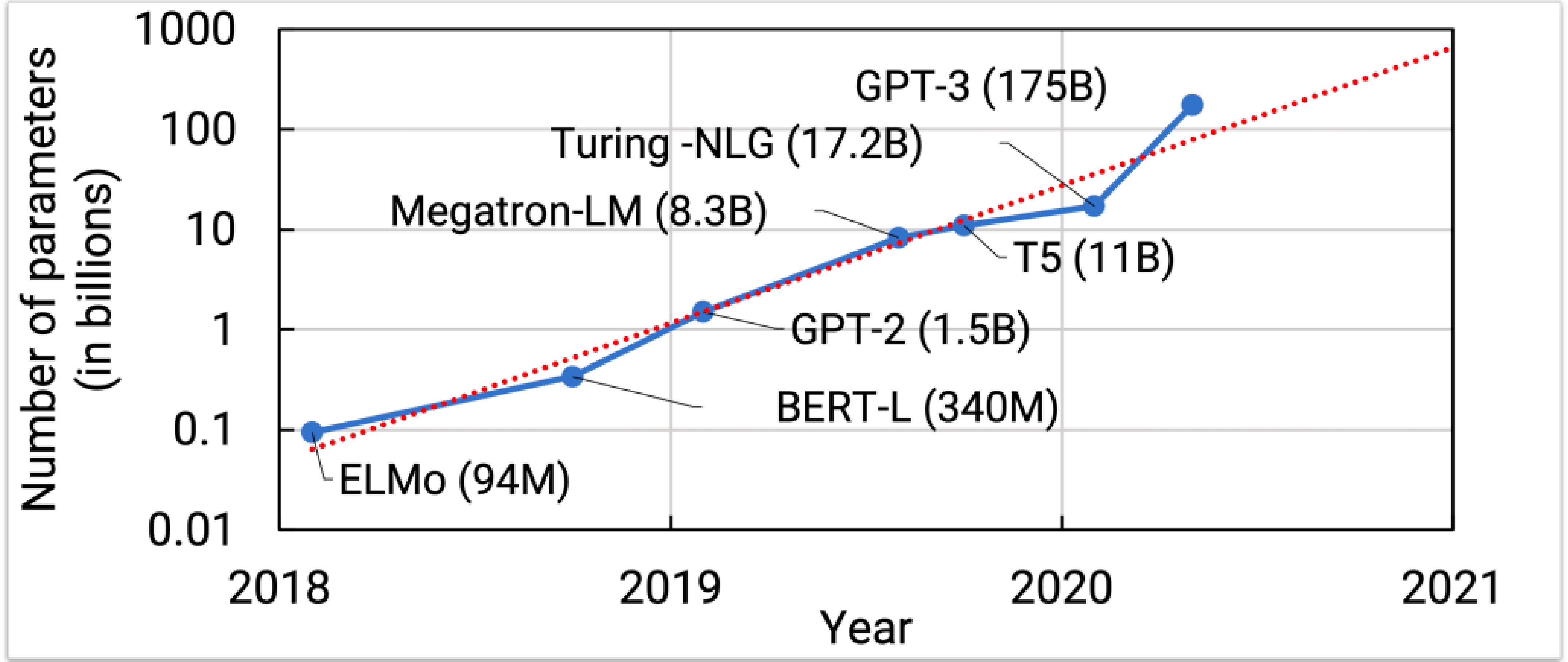
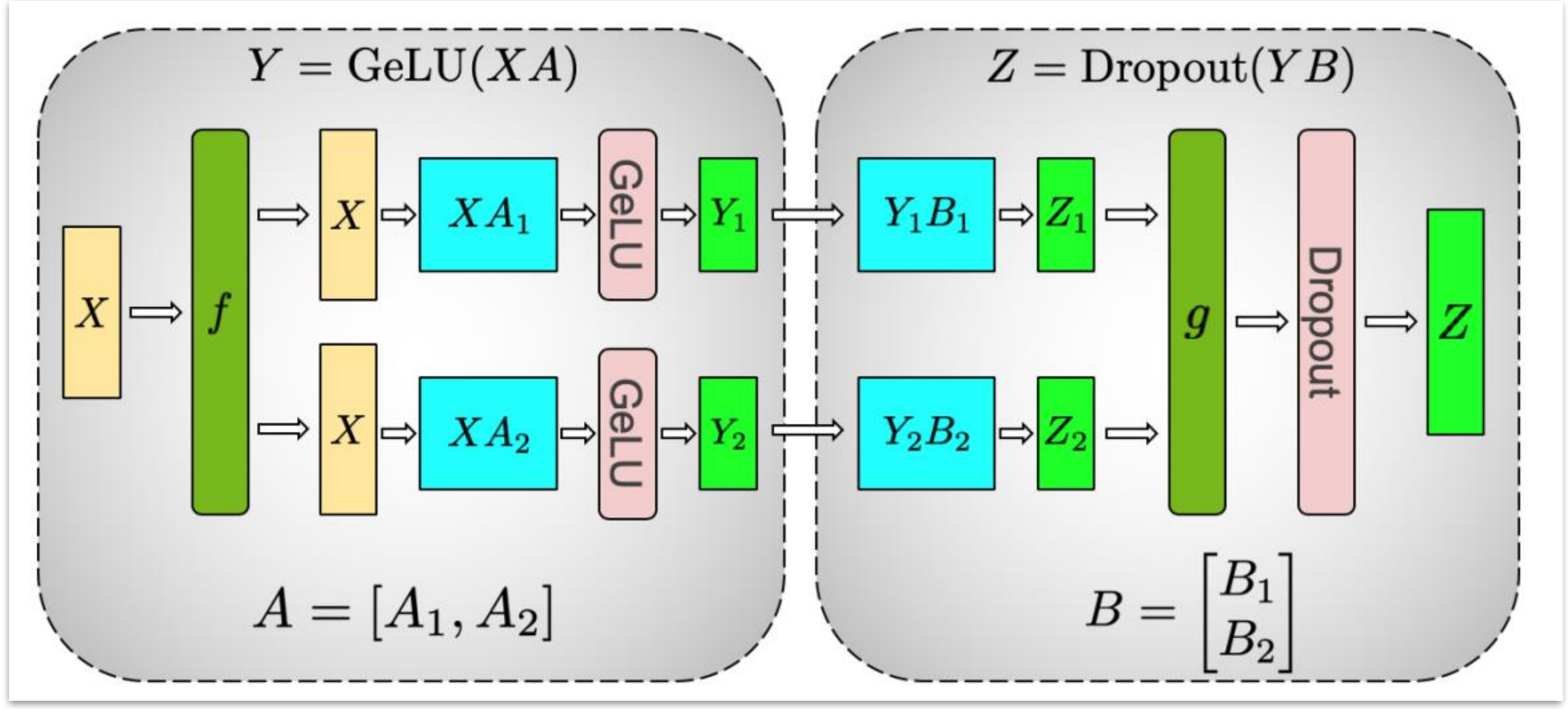
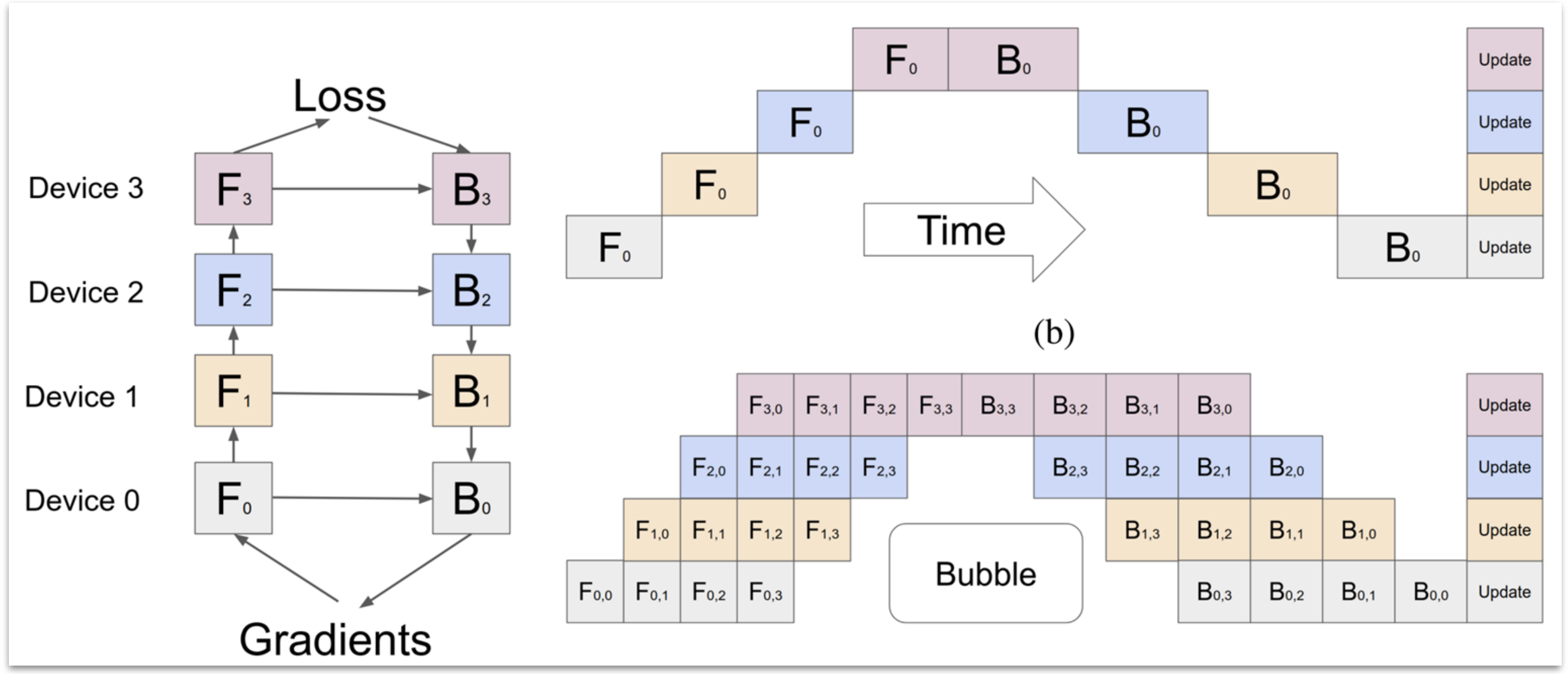


Figure from Nvidia



Matmul partitioning
[Shoeybi et al., ICML 2020]



Pipeline parallelism
[Huang et al., NeurIPS 2019]

GPT-3

2012

2016

2018

2020

GPT-3, trained with massive model parallelisms, enables new ML breakthroughs

AI

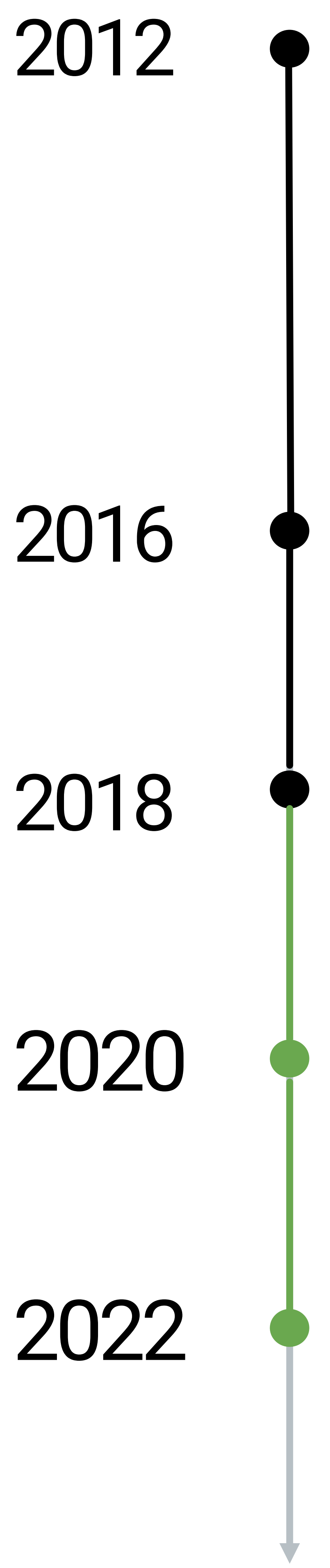
OpenAI debuts gigantic GPT-3 language model with 175 billion parameters

KHARI JOHNSON @KHARIJOHNSON MAY 29, 2020 8:34 AM

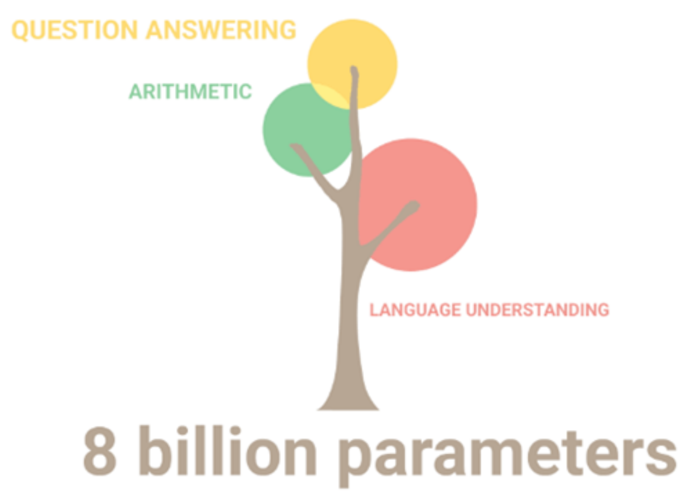
OpenAI booth at NeurIPS 2019 in Vancouver, Canada
Image Credit: Khari Johnson / VentureBeat

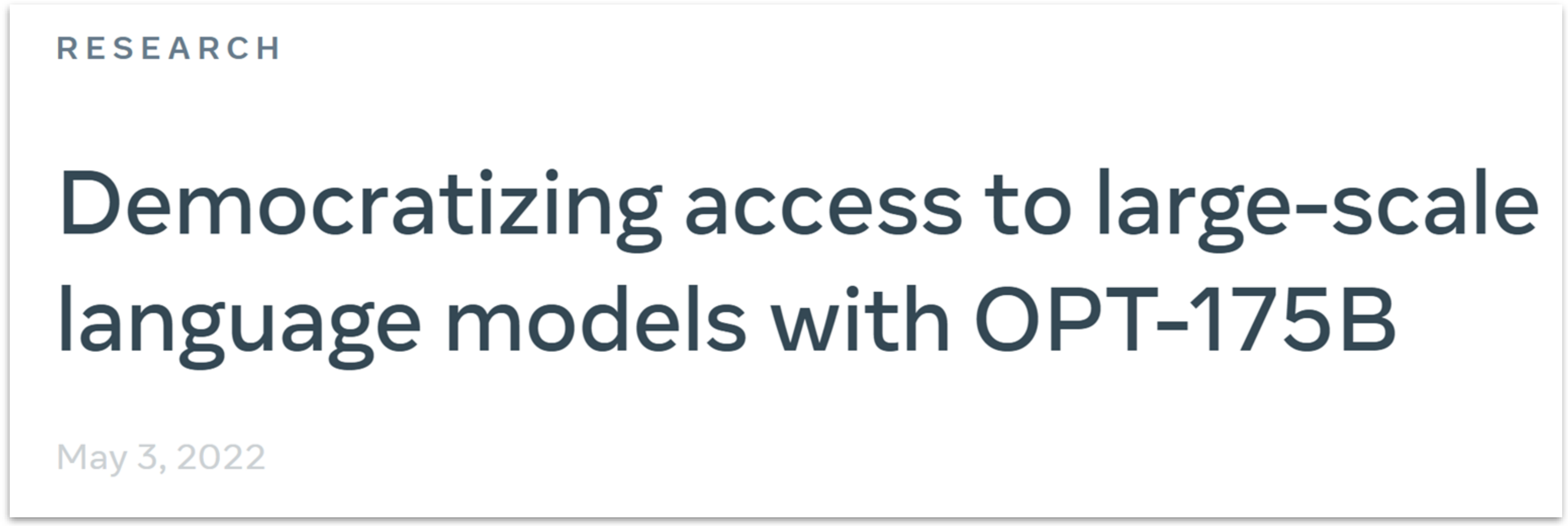
A team of more than 30 OpenAI researchers have released a paper about GPT-3, a language model capable of achieving state-of-the-art results on a set of benchmark and unique natural language processing tasks that range from language translation to generating news articles to answering SAT questions. GPT-3 has a whopping 175 billion parameters. By comparison, the largest version of

Big Model Era



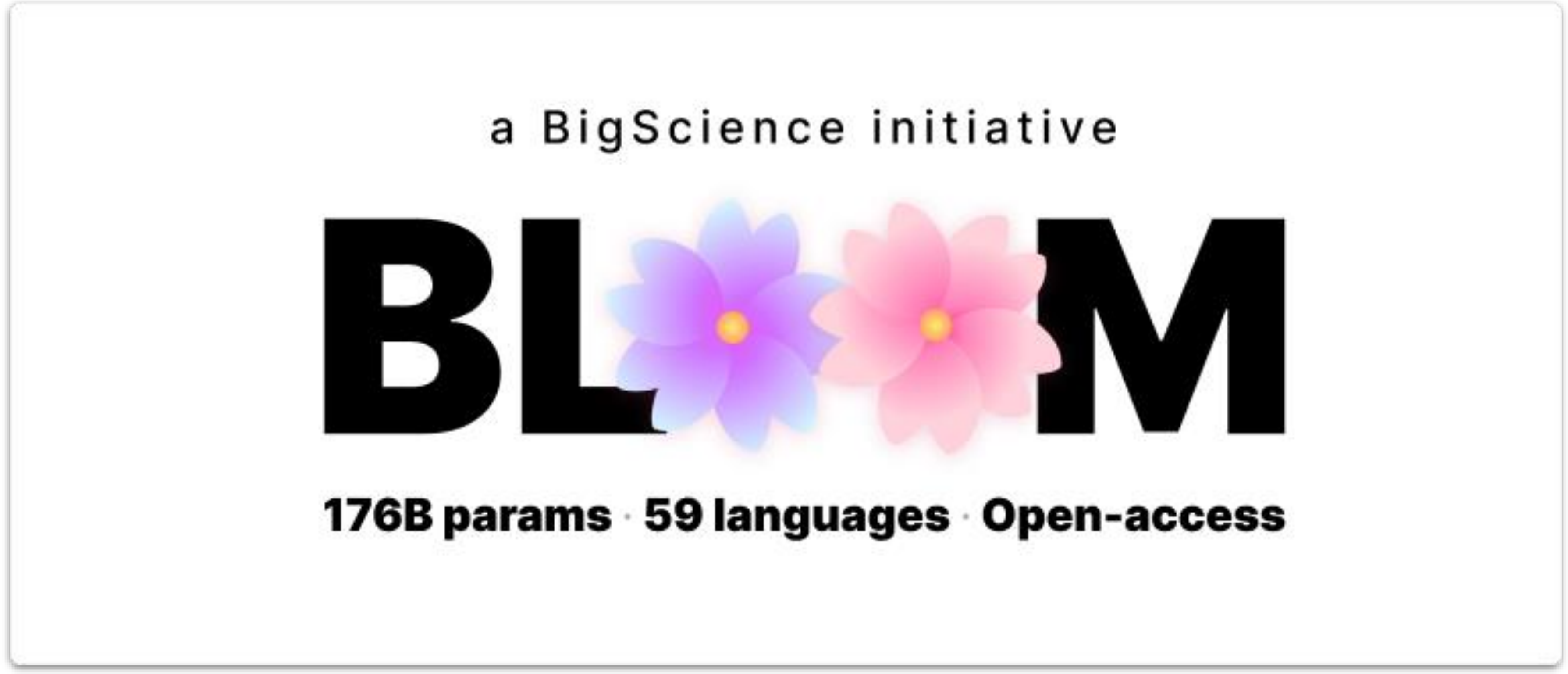
How to embrace big models?




RESEARCH

Democratizing access to large-scale language models with OPT-175B

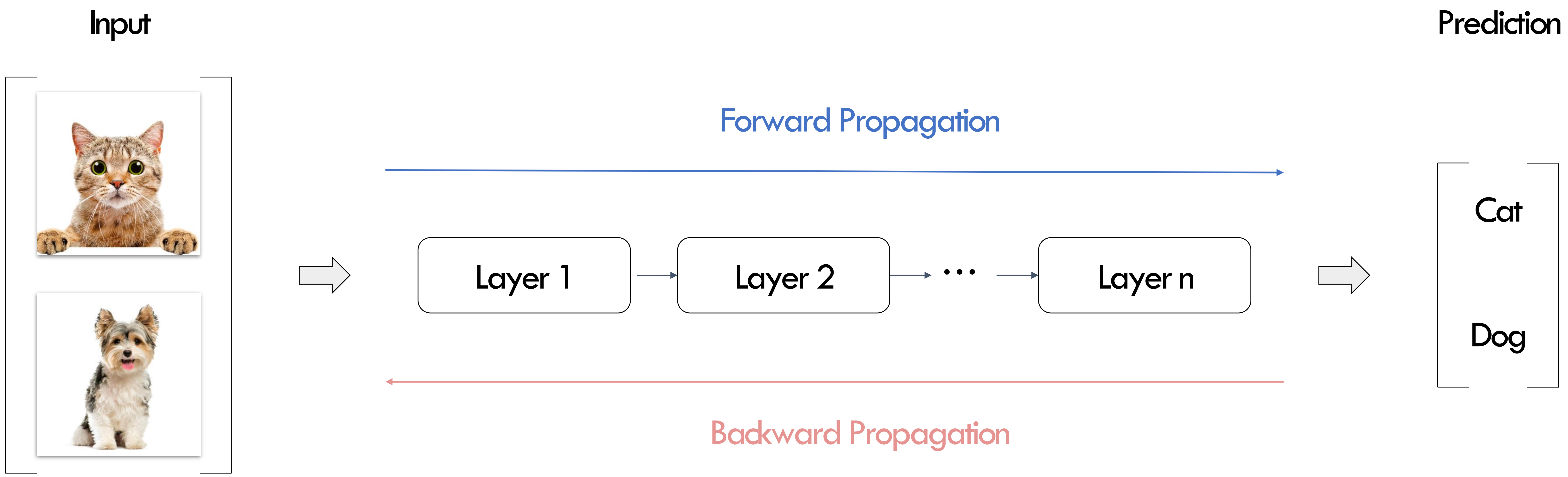
May 3, 2022



Where we are

- Motivation
- History
- **Parallelism Overview**
- Data parallelism
- Model parallelism
  - Inter and intra-op parallelism
- Auto-parallelization

Background: DL Computation



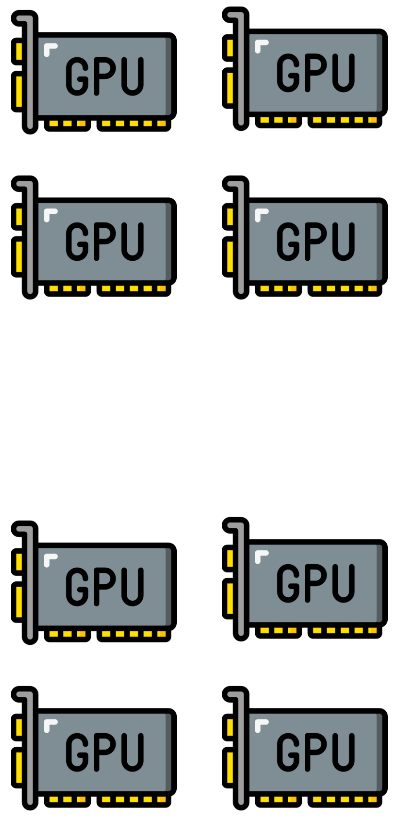
$$\underbrace{\theta^{(t+1)}}_{\text{parameter}} = \underbrace{f}_{\substack{\text{weight update}\\\text{(sgd, adam, etc.)}}}\Big(\theta^{(t)}, \underbrace{\nabla_L}_{\substack{\text{model}\\\text{(CNN, GPT, etc.)}}}\big(\theta^{(t)}, \underbrace{D^{(t)}}_{\text{data}}\big)\Big)$$

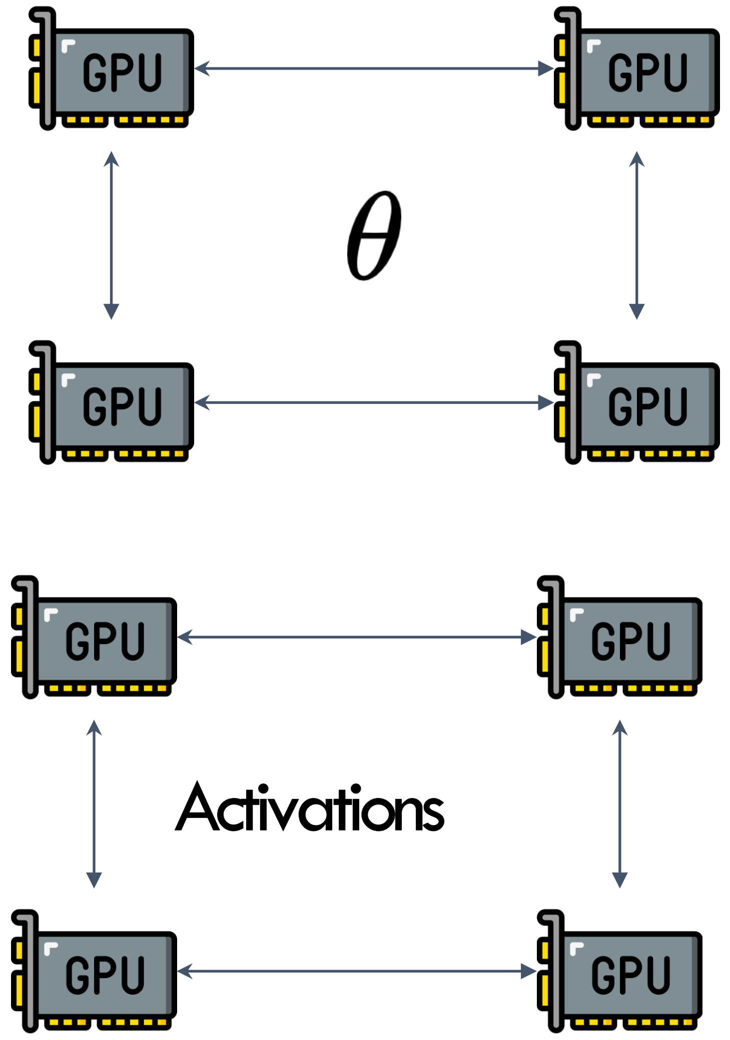
Problem Overview

Computing	**Communication**	**Memory**

$\nabla_L(\cdot)$

$f(\cdot)$

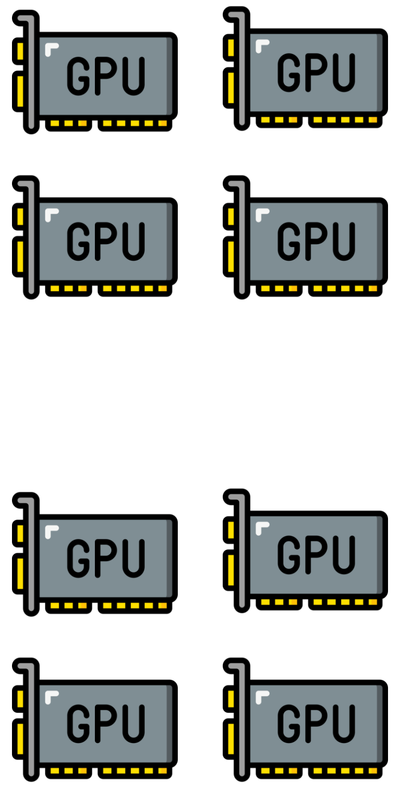




...

D

θ



$$\theta^{(t+1)} = f\big(\theta^{(t)},\ \nabla_L\big(\theta^{(t)},\ D^{(t)}\big)\big)$$

parameter

weight update
(sgd, adam, etc.)

model
(CNN, GPT, etc.)

data

Two Views of ML Parallelisms

Classic view

Data parallelism

Model parallelism

New view (this tutorial)

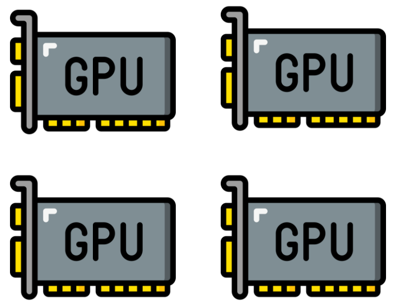
Inter-op parallelism

Intra-op parallelism

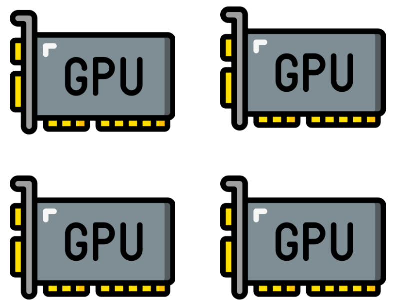
Data and Model Parallelism

Data parallelism

D partition

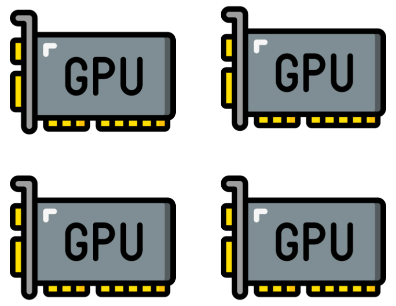


$\theta,\ \nabla_L(\cdot),\ f(\cdot)$ replicate

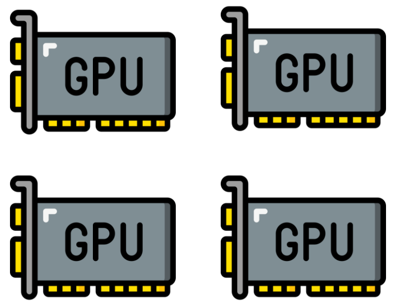


Model parallelism

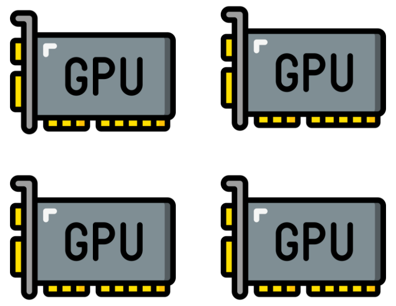
$\theta,\ \nabla_L(\cdot)$ partition



D ?



$f(\cdot)$?



$$\underbrace{\theta^{(t+1)}}_{\text{parameter}} = \underbrace{f}_{\substack{\text{weight update}\\ \text{(sgd, adam, etc.)}}}\Big(\theta^{(t)},\ \underbrace{\nabla_L}_{\substack{\text{model}\\ \text{(CNN, GPT, etc.)}}}\big(\theta^{(t)},\ \underbrace{D^{(t)}}_{\text{data}}\big)\Big)$$

Two Views of ML Parallelisms

Data and model parallelism

- Two pillars: **data** and **model**.
- ✅ "Data parallelism" is general and precise.
- ❓ "Model parallelism" is vague.
- ❓ The view creates ambiguity for methods that neither partitions data nor the model computation.

New: Inter-op and Intra-op parallelism.

- Two pillars: **computational graph** and **device cluster**
- ✅ This view is based on their computing characteristics.
- ✅ This view facilitates the development of new parallelism methods.

DL Computation

$$\theta^{(t+1)} = f\left(\theta^{(t)}, \nabla_L\left(\theta^{(t)}, D^{(t)}\right)\right)$$

$$L = \mathrm{MSE}(w_2 \cdot \mathrm{ReLU}(w_1 x), y) \quad \theta = \{w_1, w_2\}, D = \{(x, y)\}$$

$$f(\theta, \nabla_L) = \theta - \nabla_L$$

Forward	Backward	Weight update
$L(\cdot)$	$\nabla_L(\cdot)$	$f(\cdot)$

Device Cluster

Nvidia DGX with V100

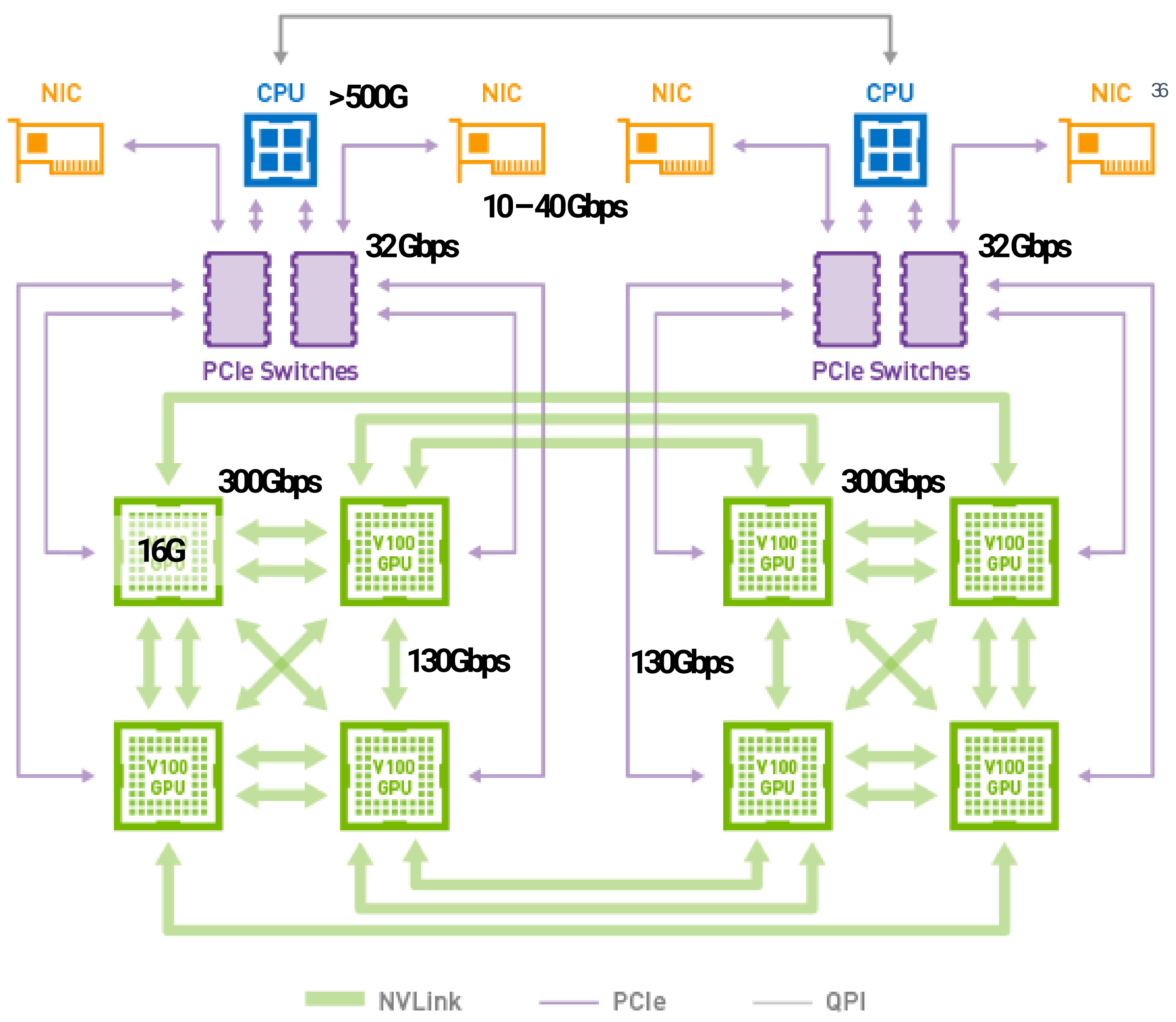
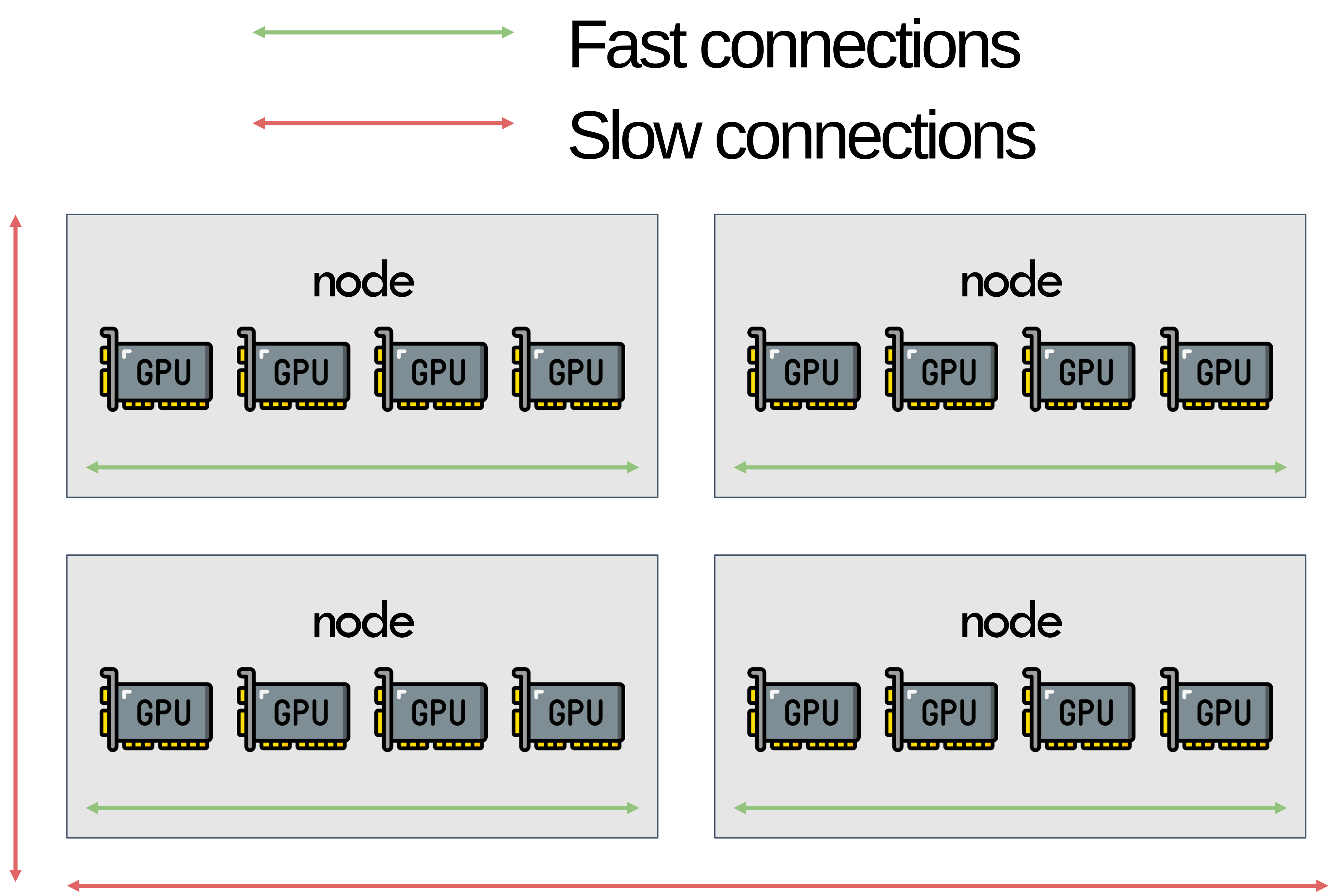


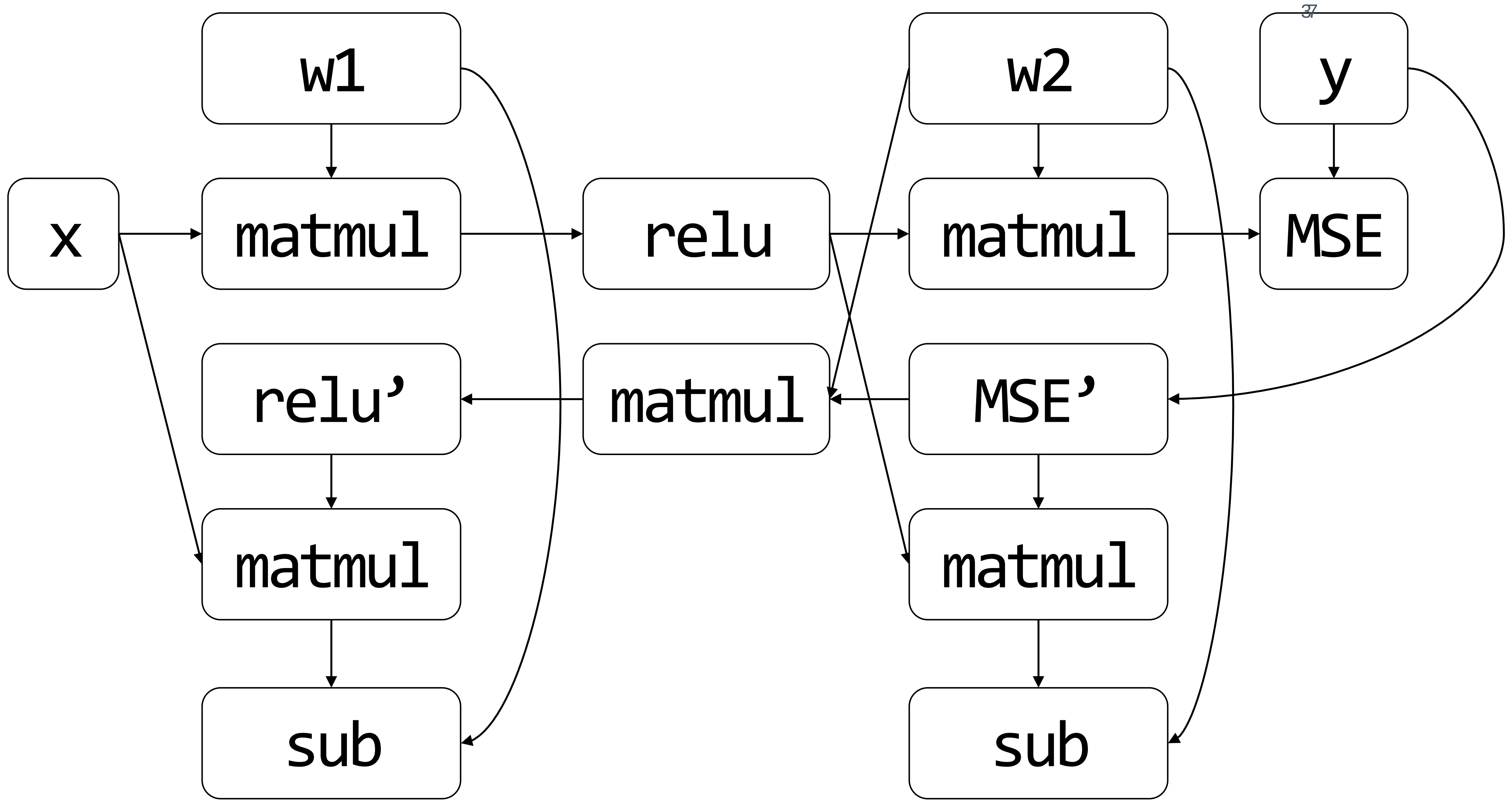
Figure from NVIDIA

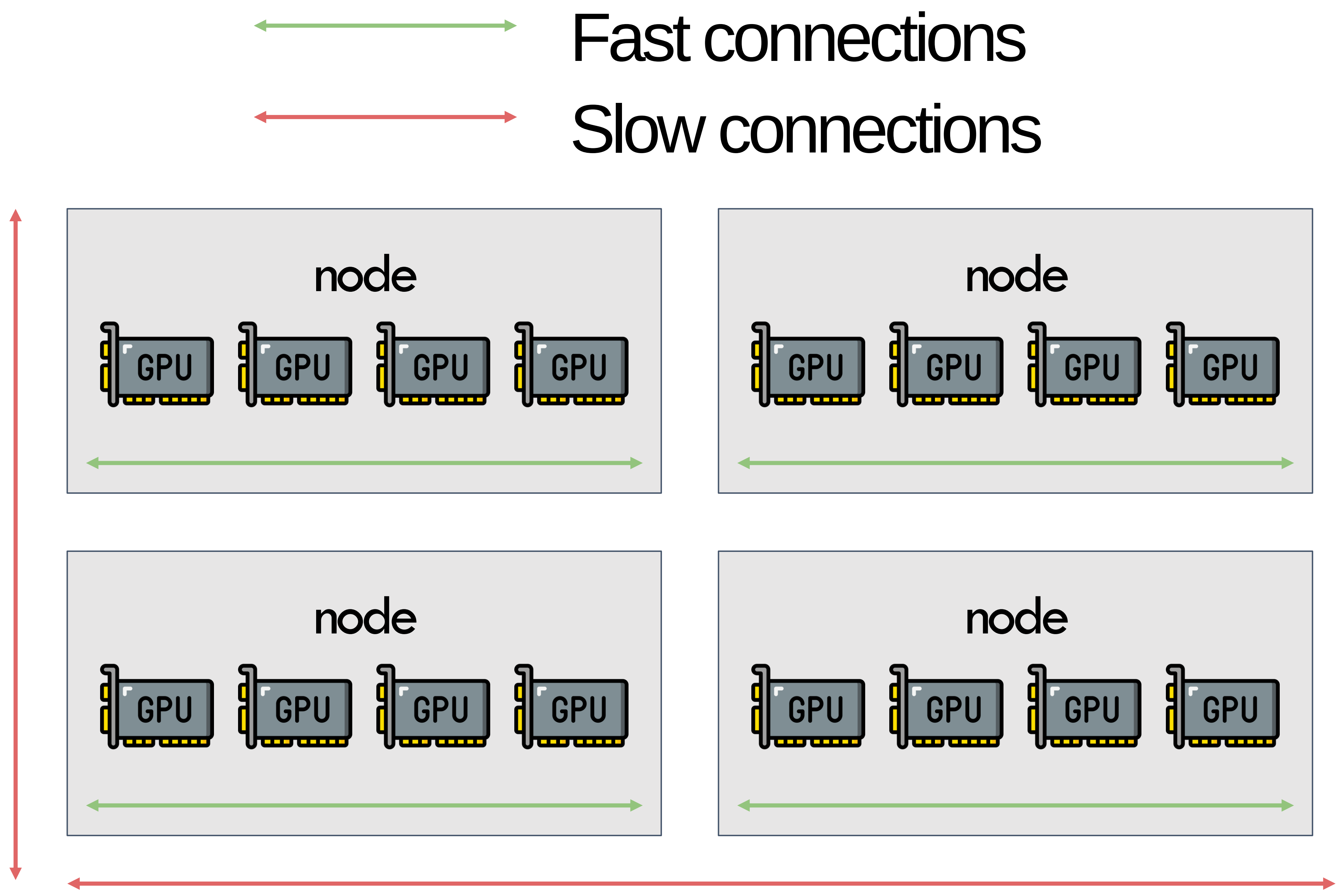
A typical GPU cluster topology



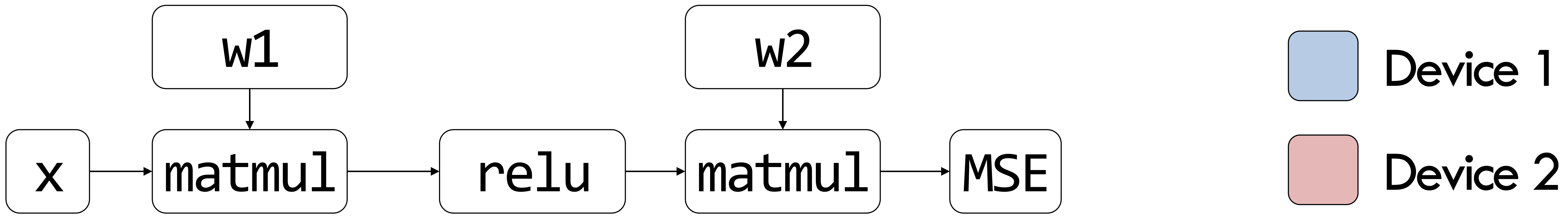
Partitioning Computation Graph on Device Cluster

How to partition the computational graph on the device cluster?

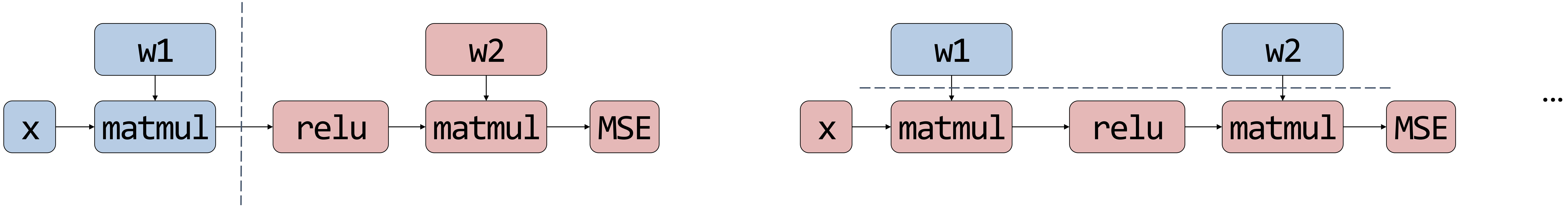




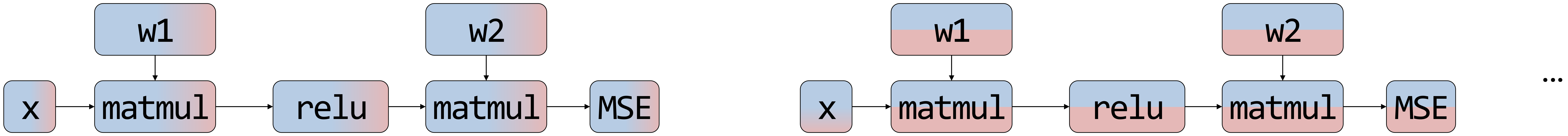
Partitioning Computation Graph



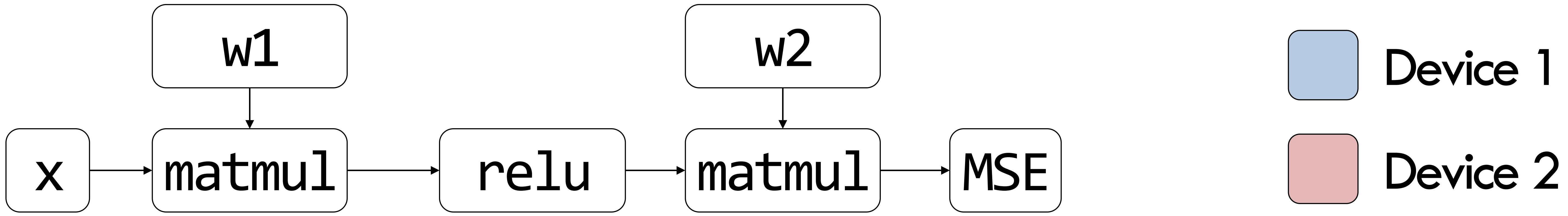
Strategy 1



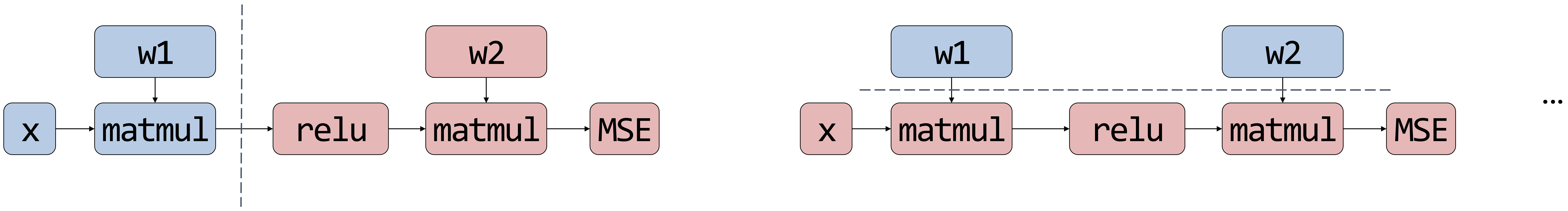
Strategy 2



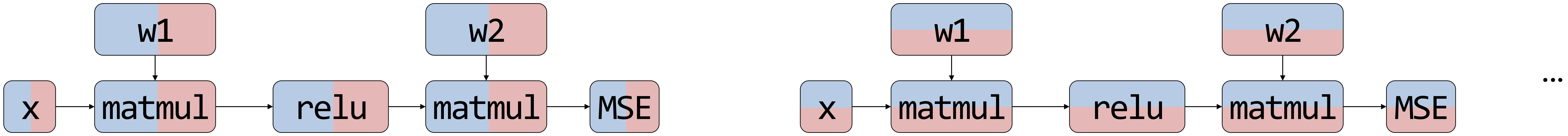
Partitioning Computation Graph



Strategy 1: Inter-operator Parallelism



Strategy 2: Intra-operator Parallelism

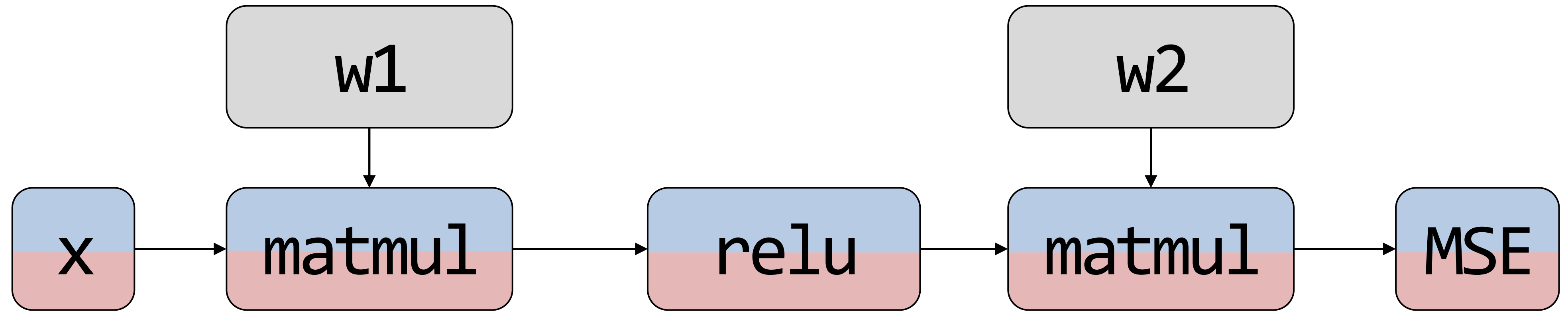


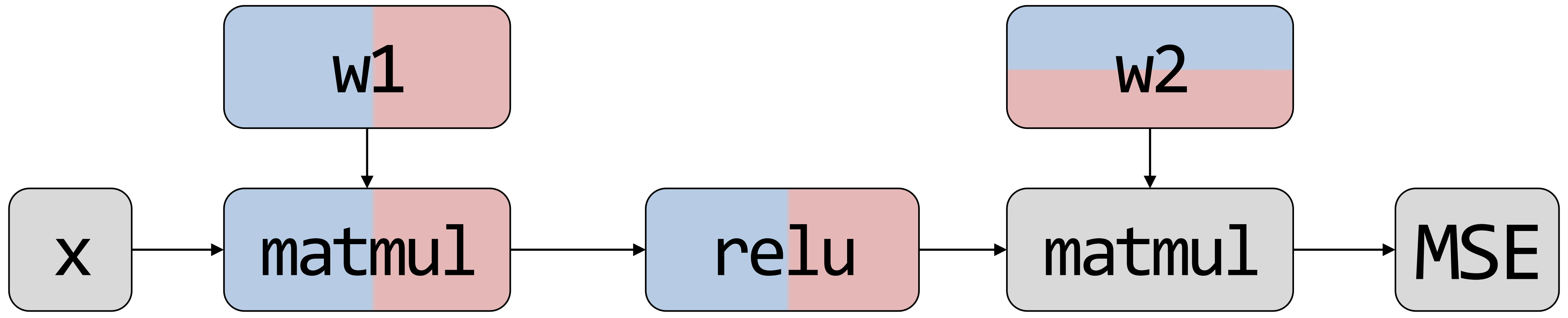
More Parallelisms…

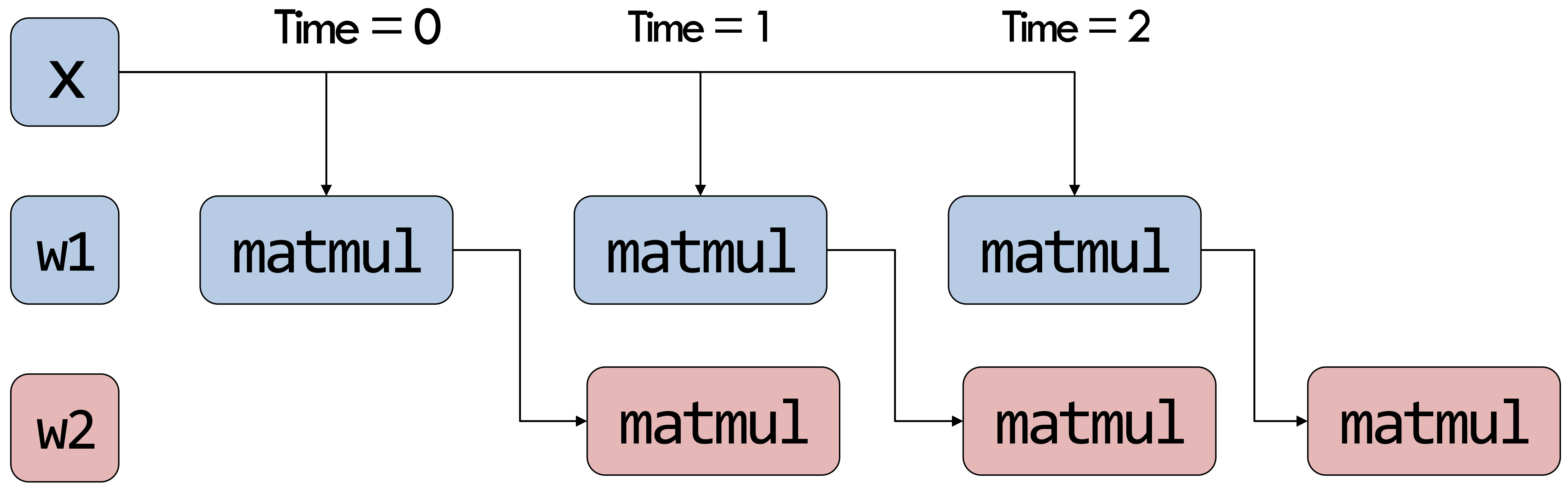
Multiple intra-op strategies for a single node

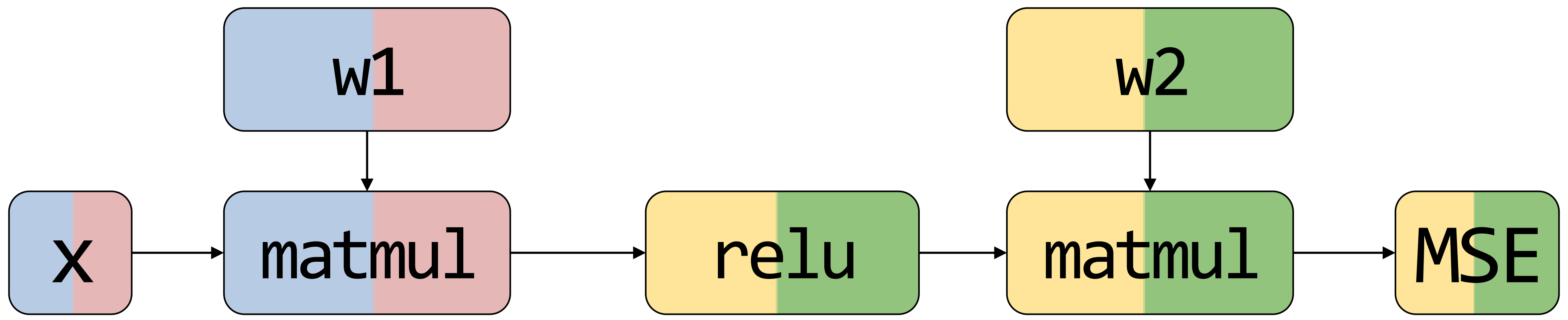
Row-partitioned Column-partitioned Replicated Device 3 Device 4

More strategies

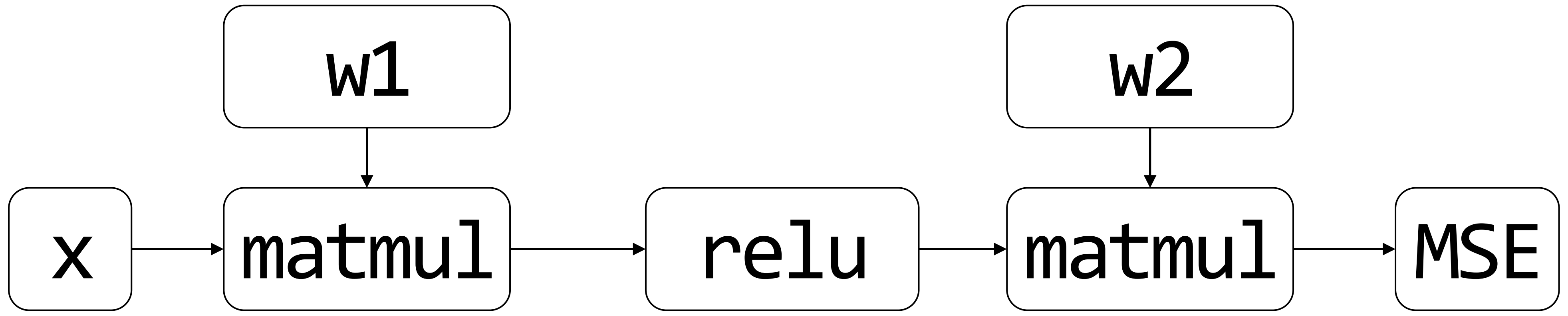


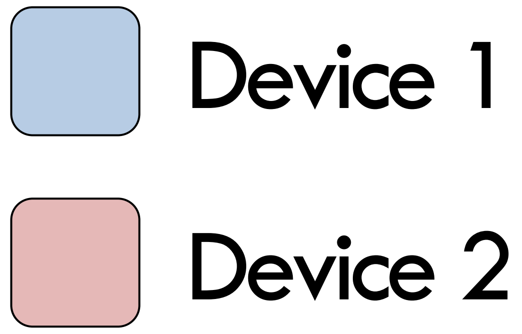




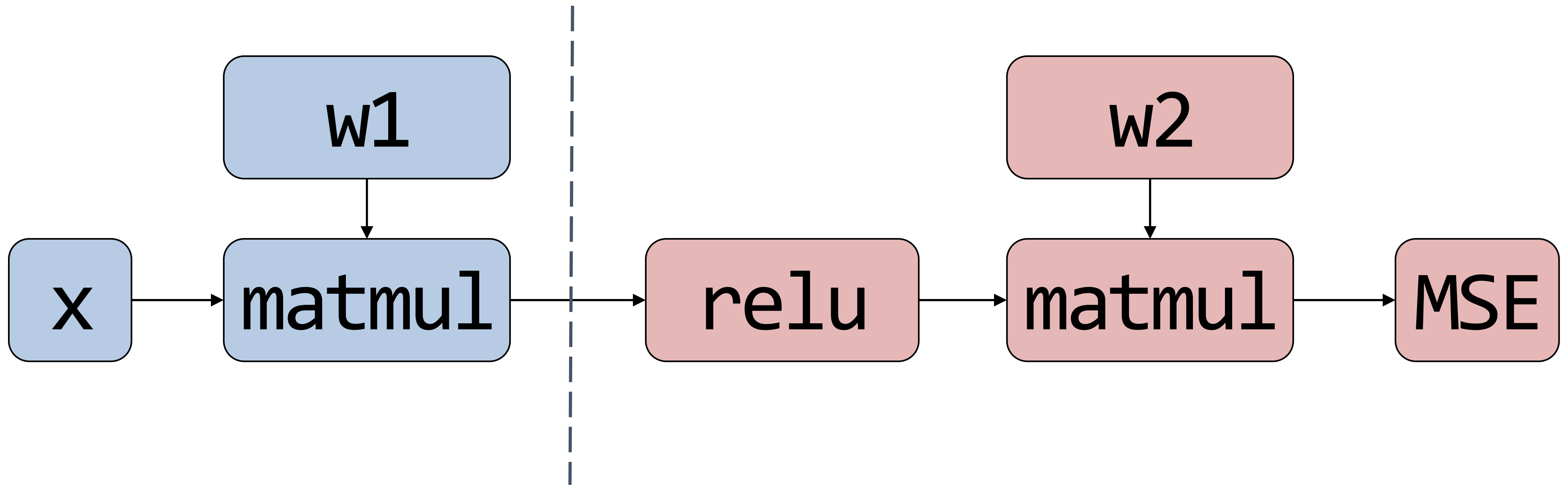


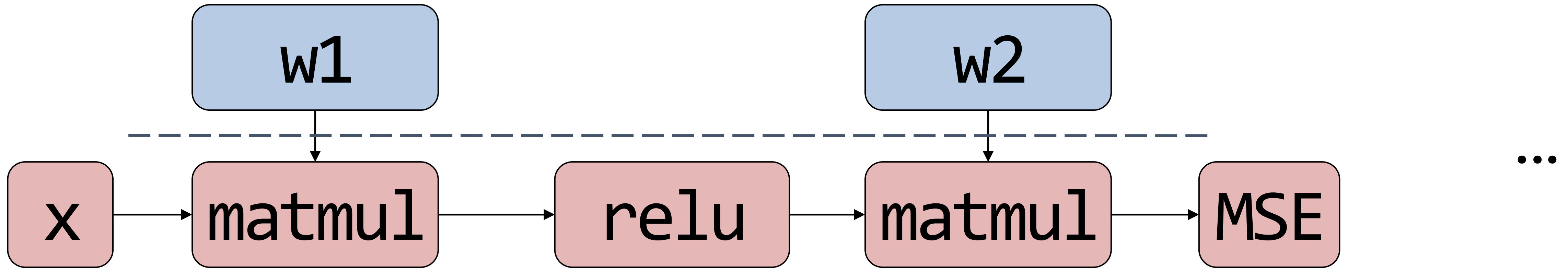
Summary: Inter-op and Intra-op Parallelisms



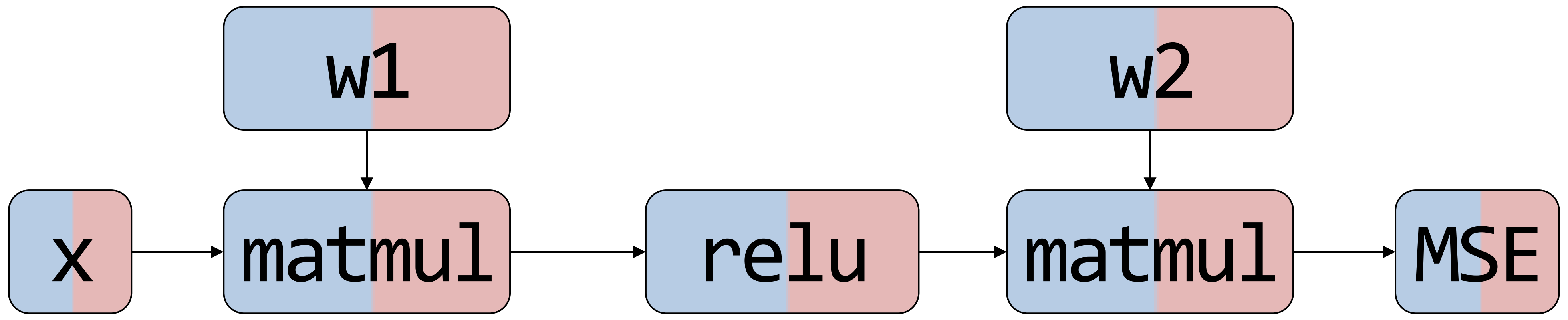


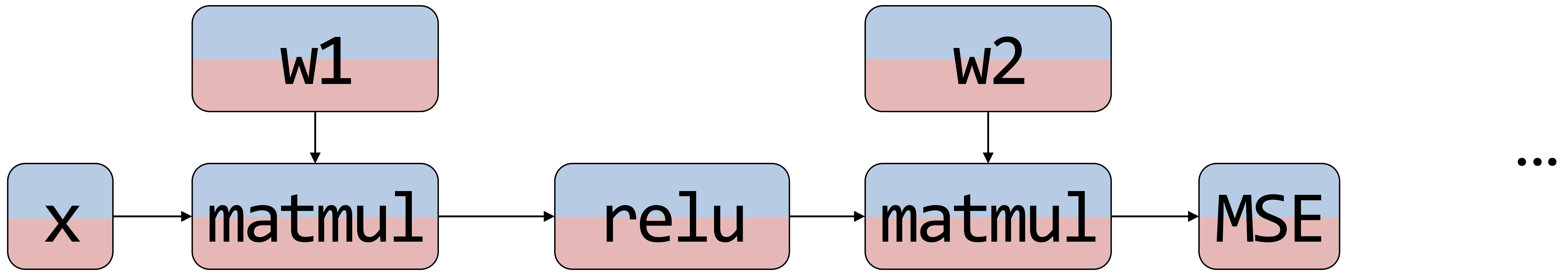
Inter-op parallelism: Assign different operators to different devices.





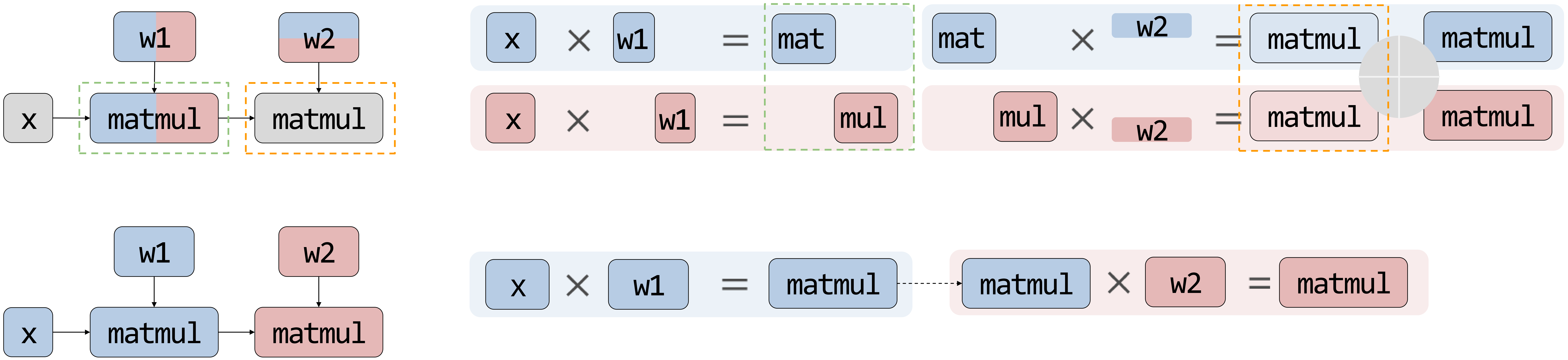
Intra-op parallelism: Assign different regions of a single operator to different devices.



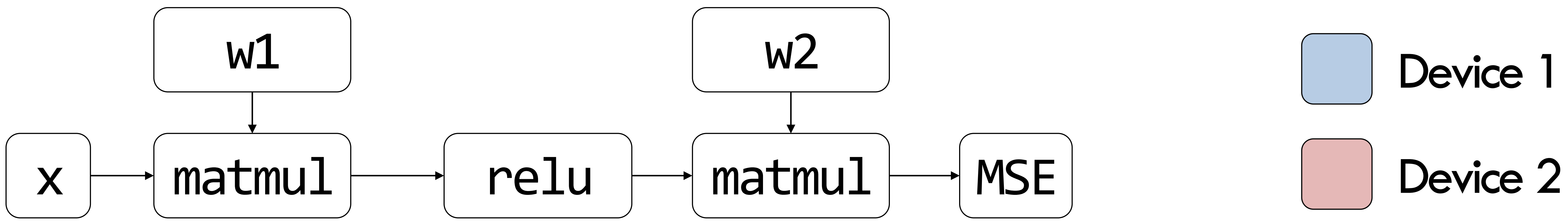


Inside Intra- and Inter-op Parallelism

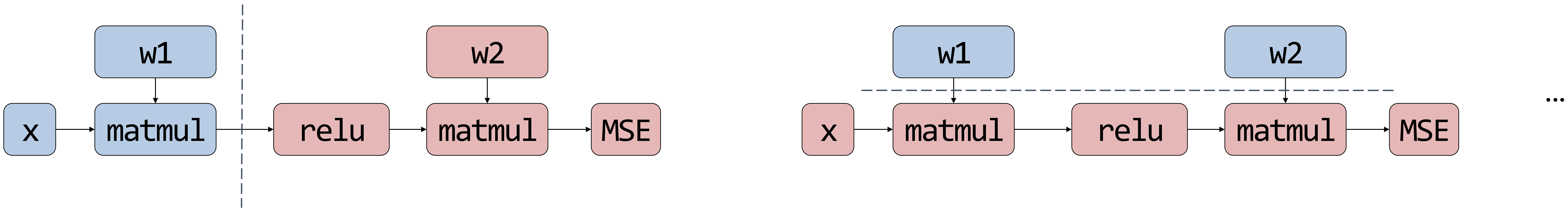
$$Y = X \cdot W_1 \cdot W_2 = X \cdot \begin{bmatrix} W_1^{d1} & W_1^{d2} \end{bmatrix} \cdot \begin{bmatrix} W_2^{d1} \\ W_2^{d2} \end{bmatrix}$$



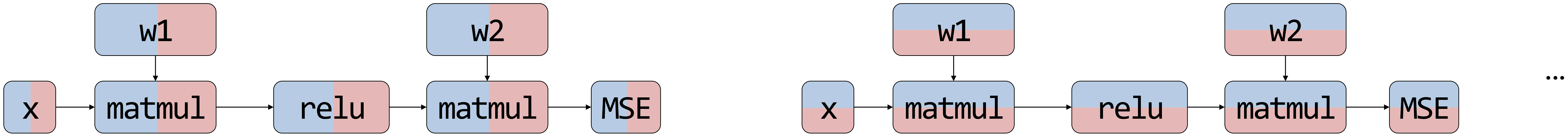
Inter-op and Intra-op Parallelism: Characteristics



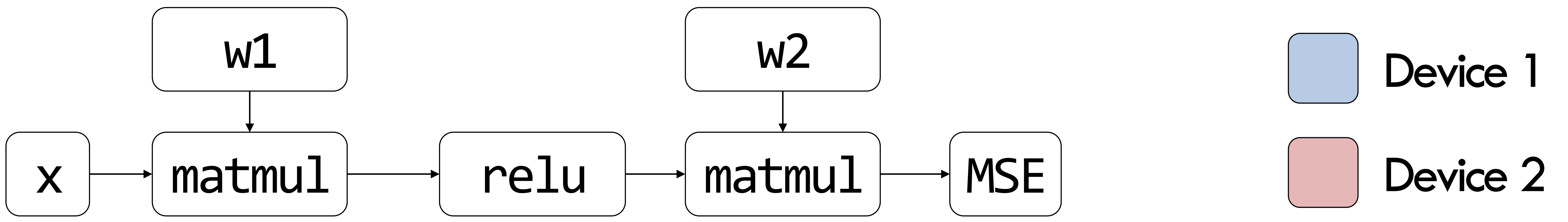
Inter-op parallelism: Requires point-to-point communication but results in device idle



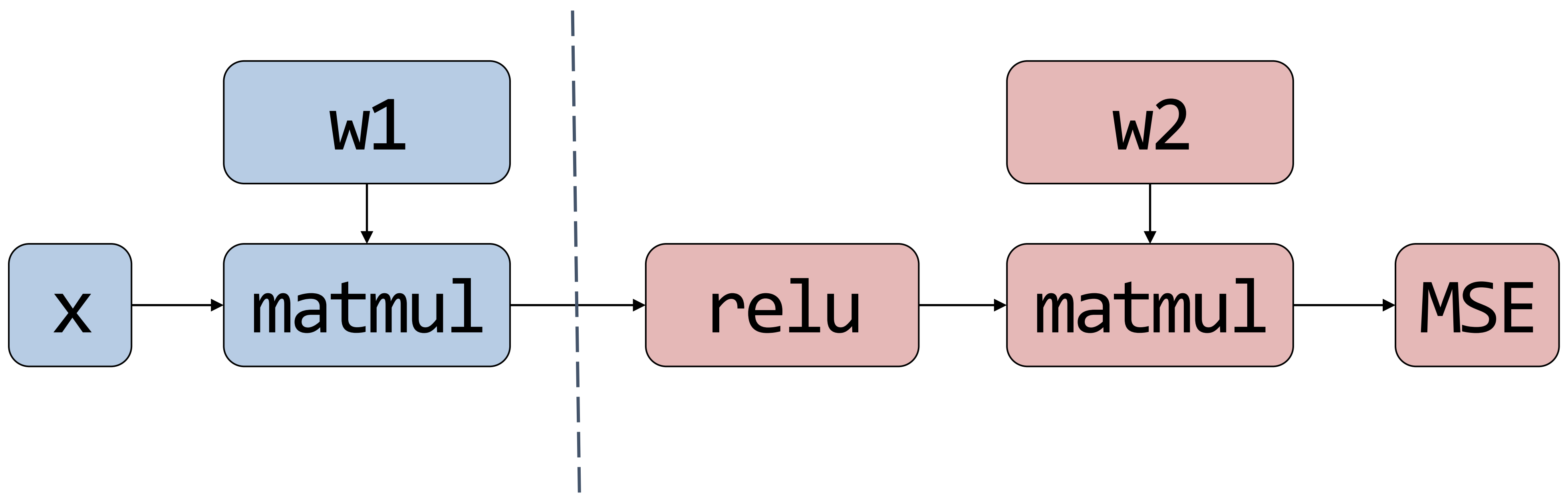
Intra-op parallelism: Devices are busy but requires collective communication



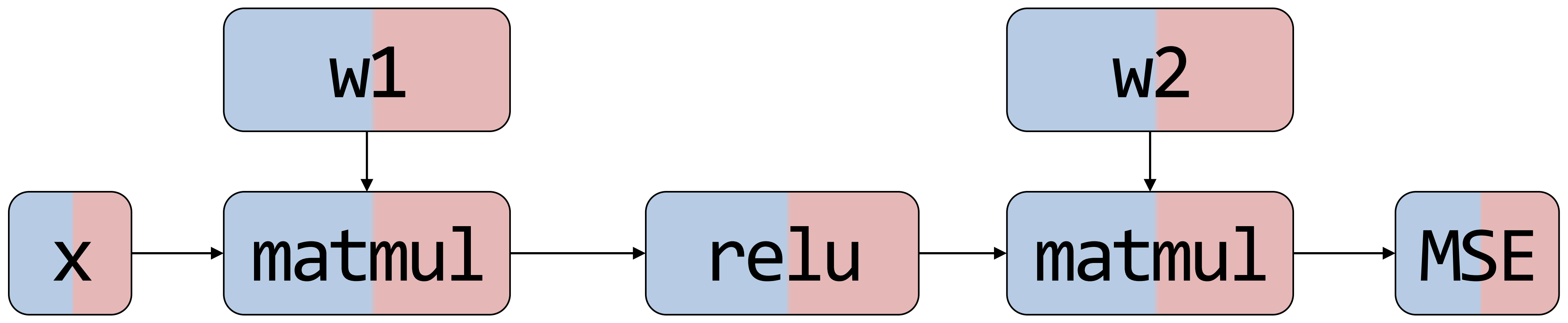
Inter-op and Intra-op Parallelism: Characteristics



Inter-op parallelism



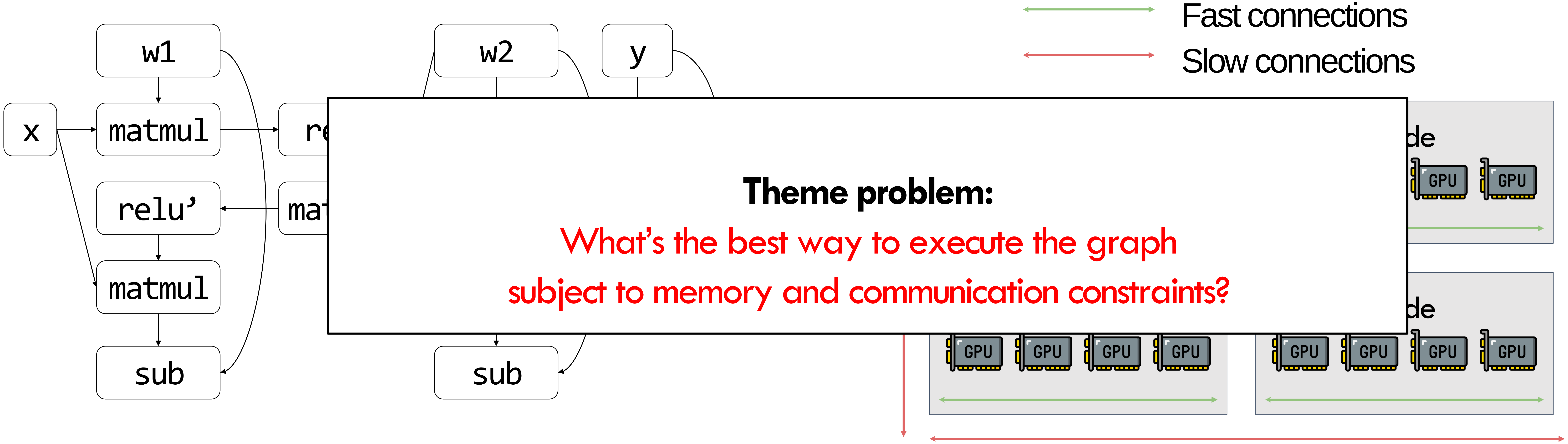
Intra-op parallelism



Trade-off

	Inter-operator Parallelism	**Intra-operator Parallelism**
Communication	Less	More
Device Idle Time	More	Less

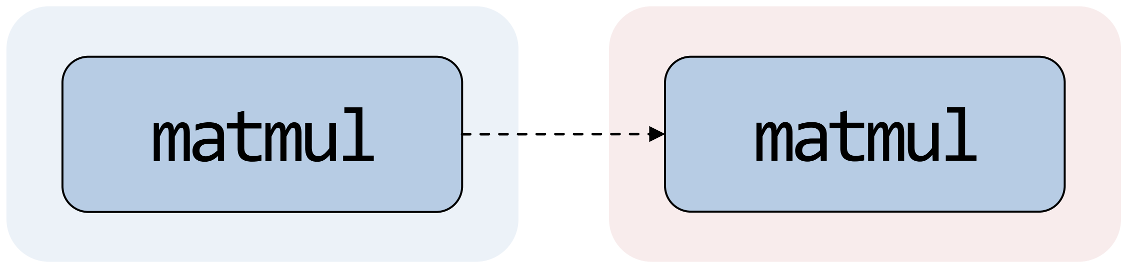
ML Parallelization under New View

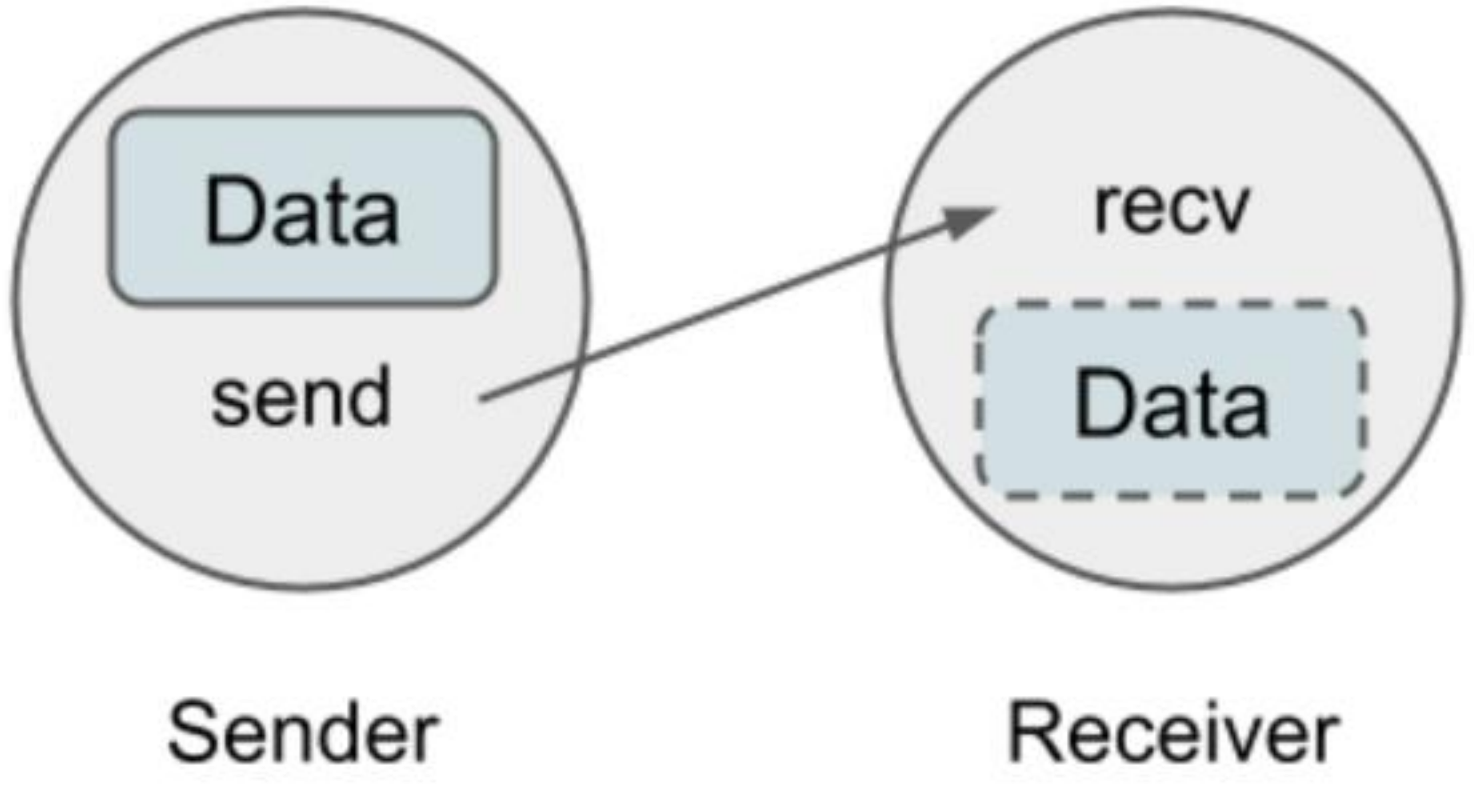


Theme problem:

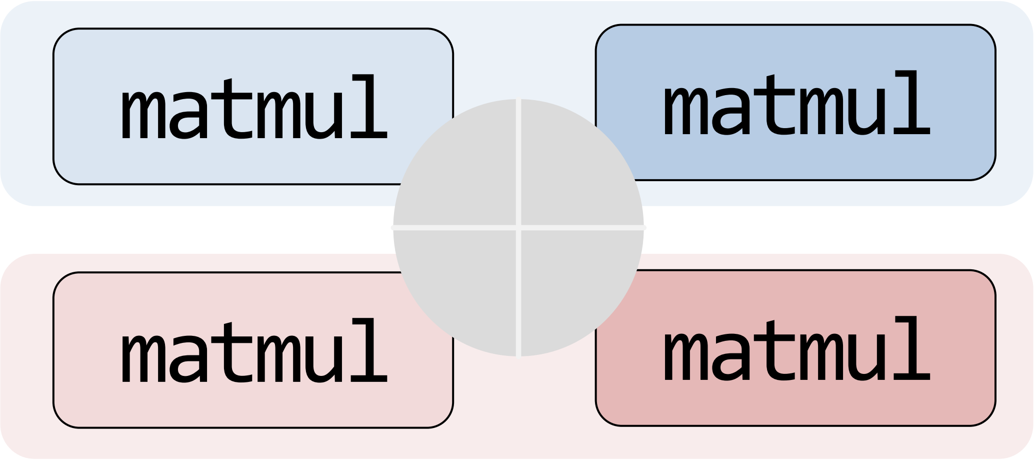
What's the best way to execute the graph subject to memory and communication constraints?

Terminologies: Point-to-point Communication





Terminologies: Collective Communication




```python
ddp_model = DDP(Model(), device_ids=[rank])
for batch in data_loader:
    loss = train_step(ddp_model, batch)
```

Implicit allreduce here

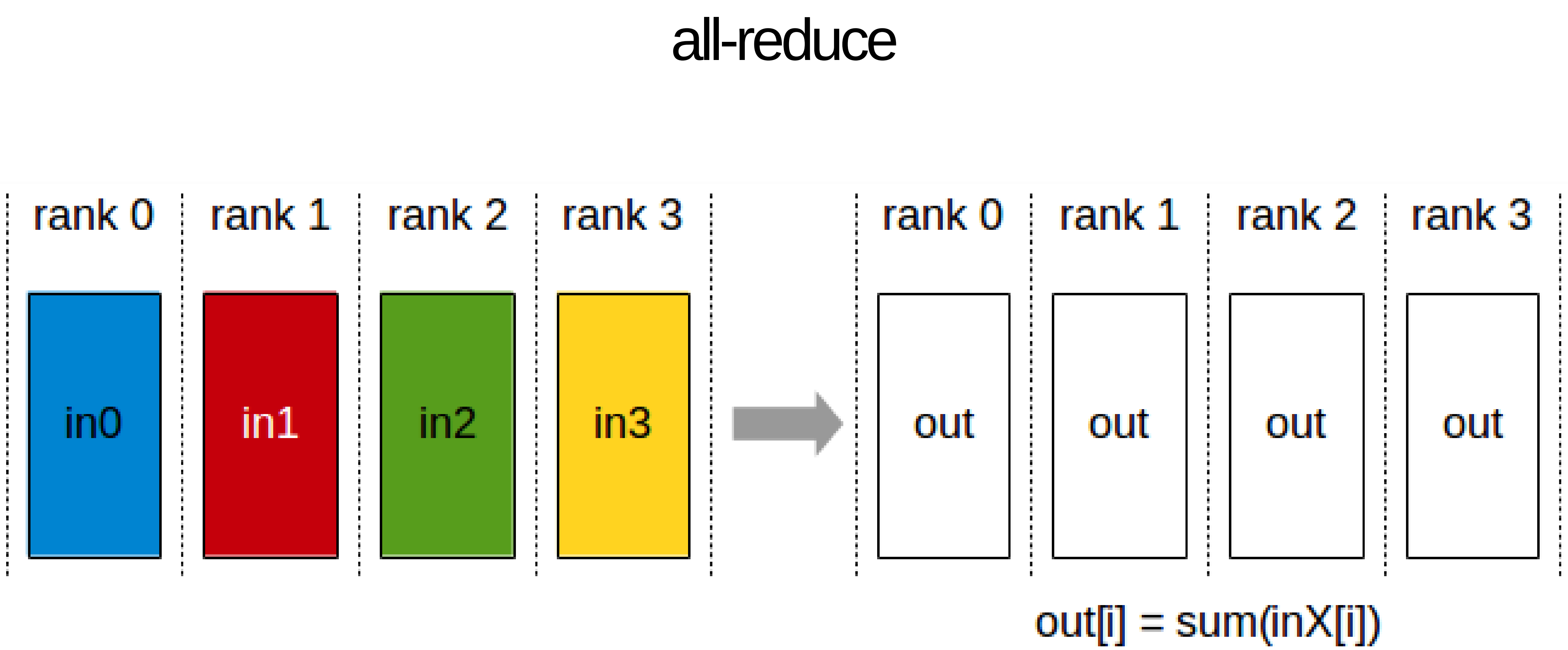
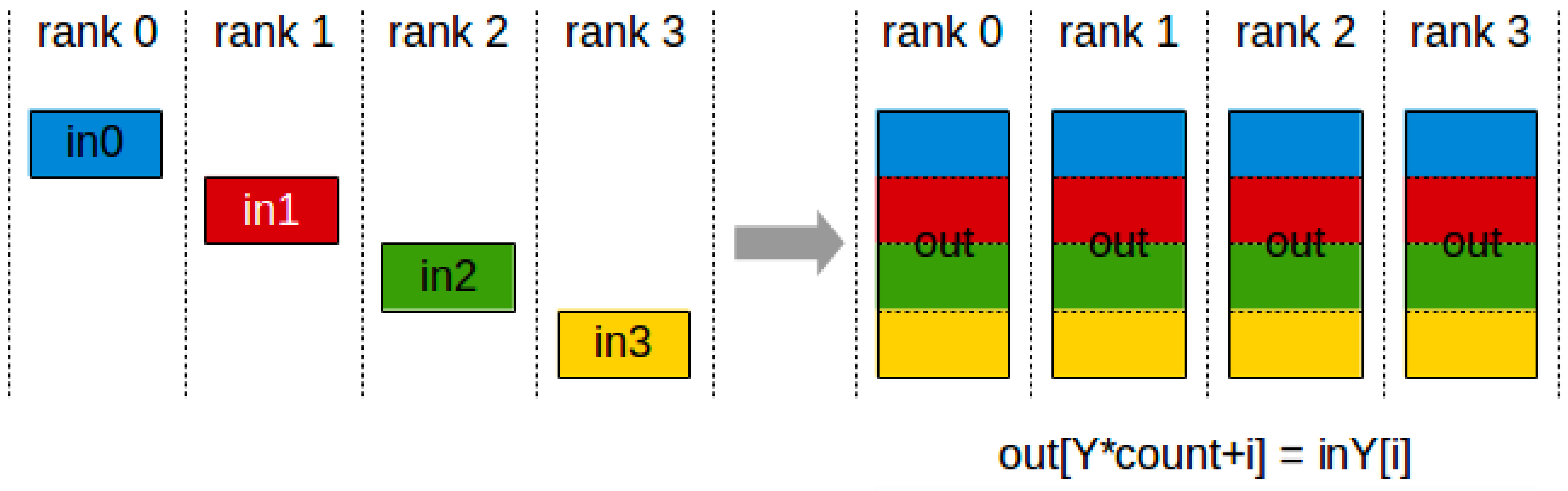


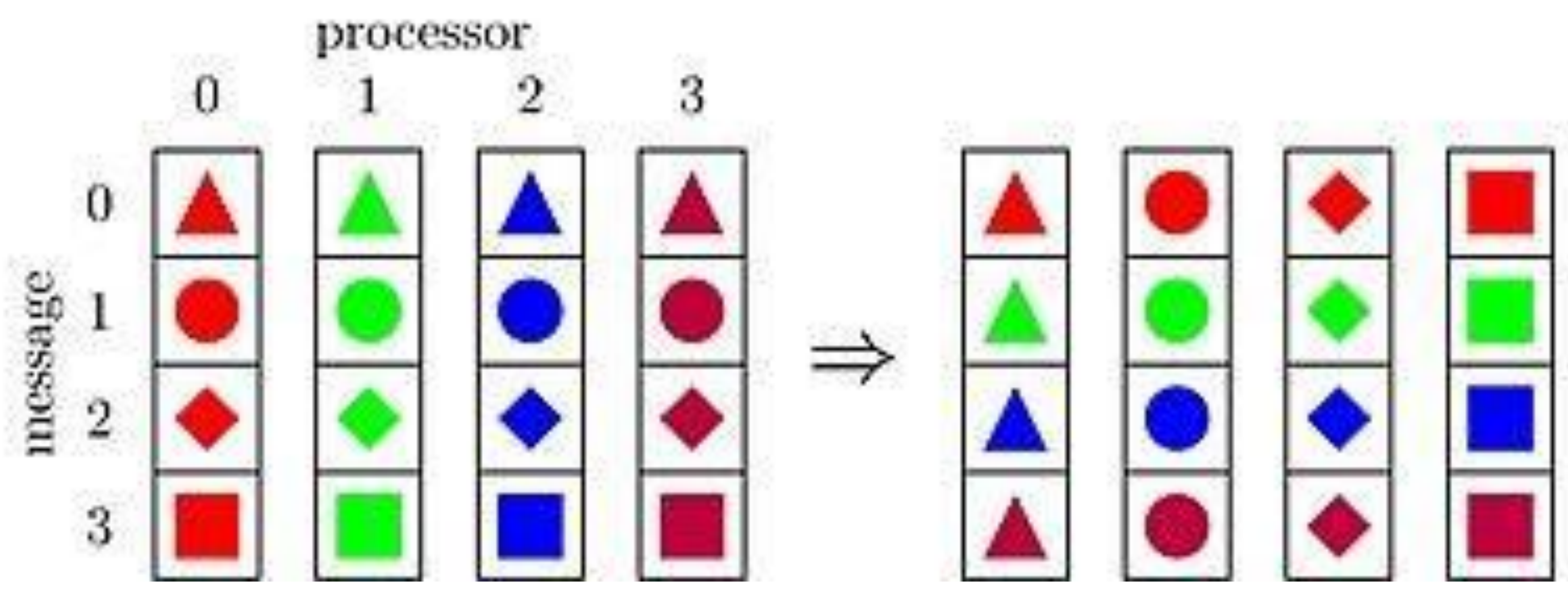
Figure from NCCL documentation

Terminologies: Collective Communication

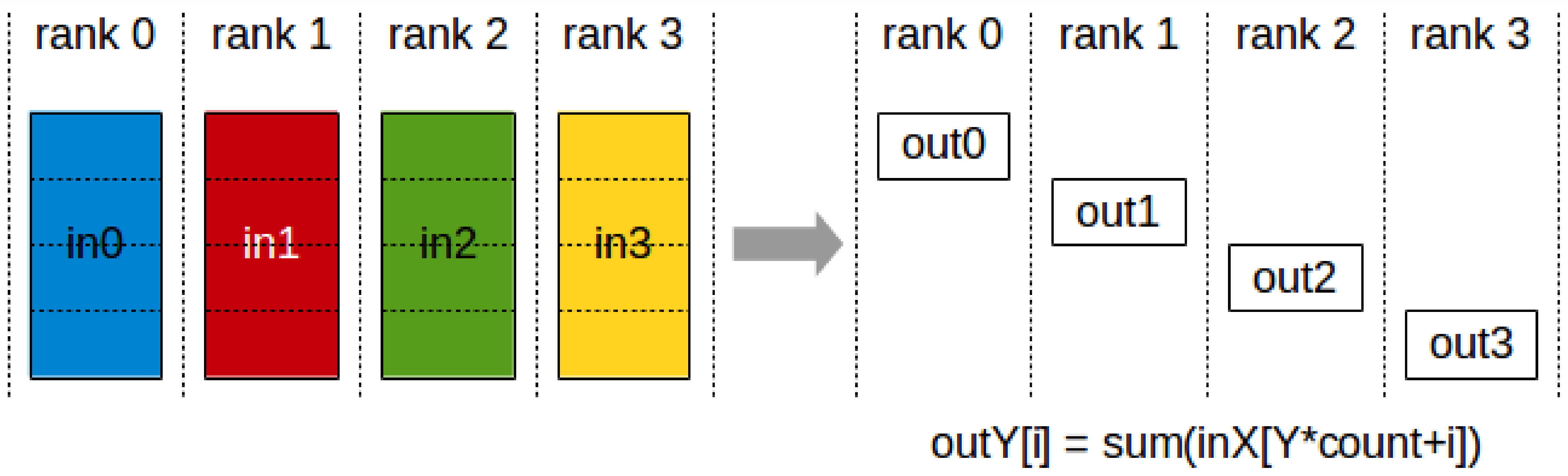
all-gather



all-to-all



Reduce-scatter



Figures from NCCL documentation

Next Week

- Motivation
- History
- Parallelism Overview
- **Data parallelism**
- Model parallelism
  - Inter and intra-op parallelism
- Auto-parallelization