https://hao-ai-lab.github.io/dsc204a-f25/

DSC 204A: Scalable Data Systems
Fall 2025

Staff
Instructor: Hao Zhang
TAs: Mingjia Huo, Yuxuan Zhang

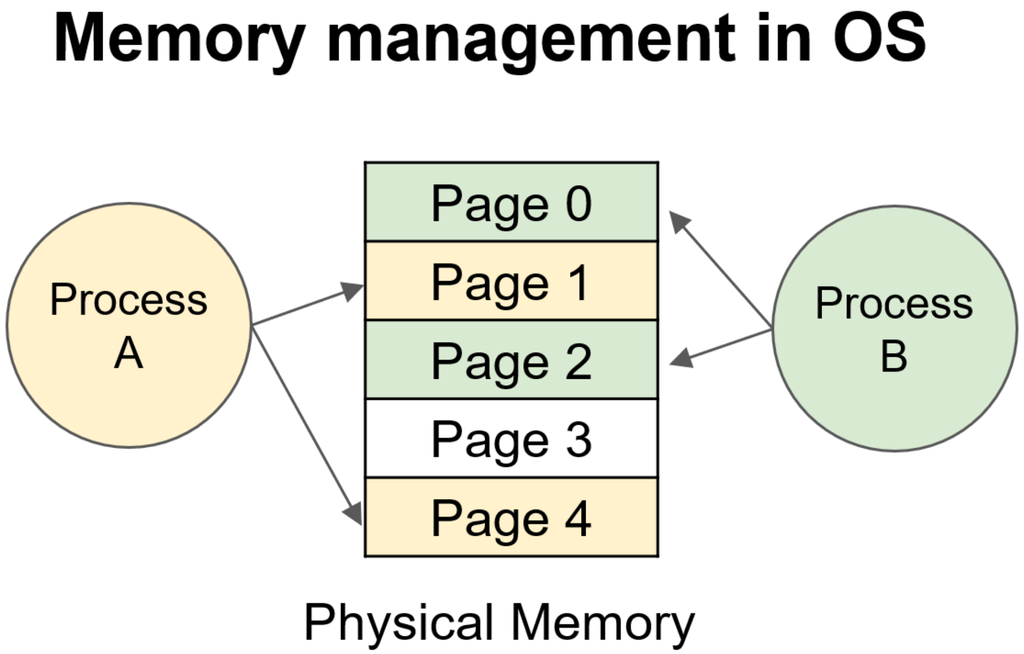
@haozhangml @haoailab
haozhang@ucsd.edu

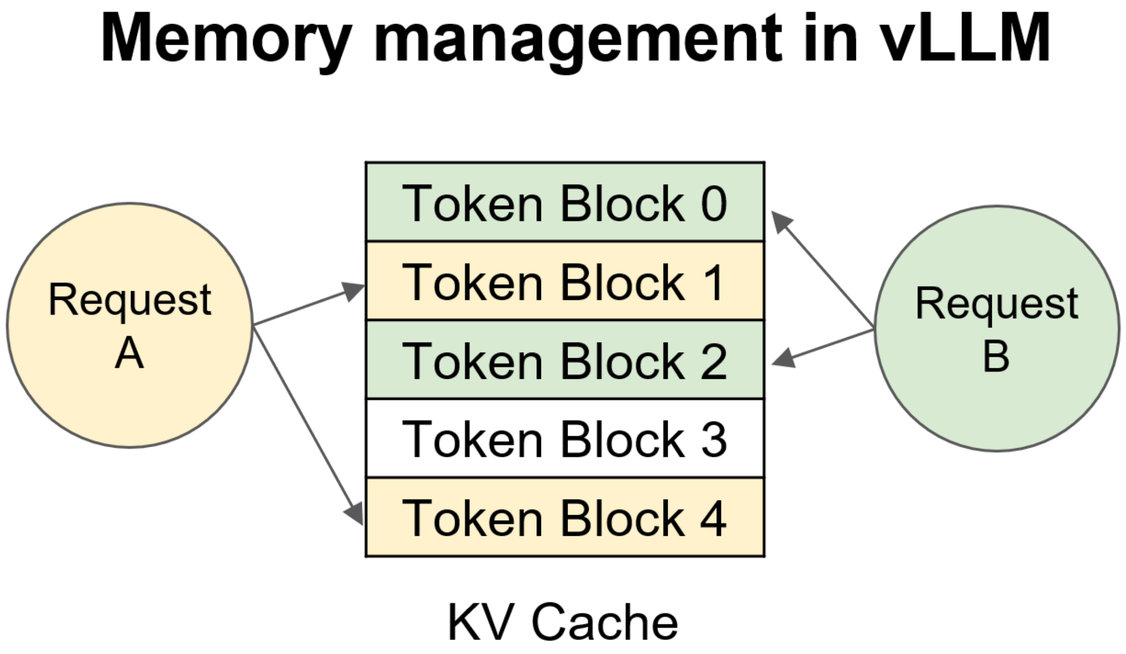
Large Language Models

- Transformers, Attentions
- **Serving and inference**
- Parallelization
- Attention optimization

vLLM: Efficient memory management for LLM inference

Inspired by **virtual memory** and **paging**



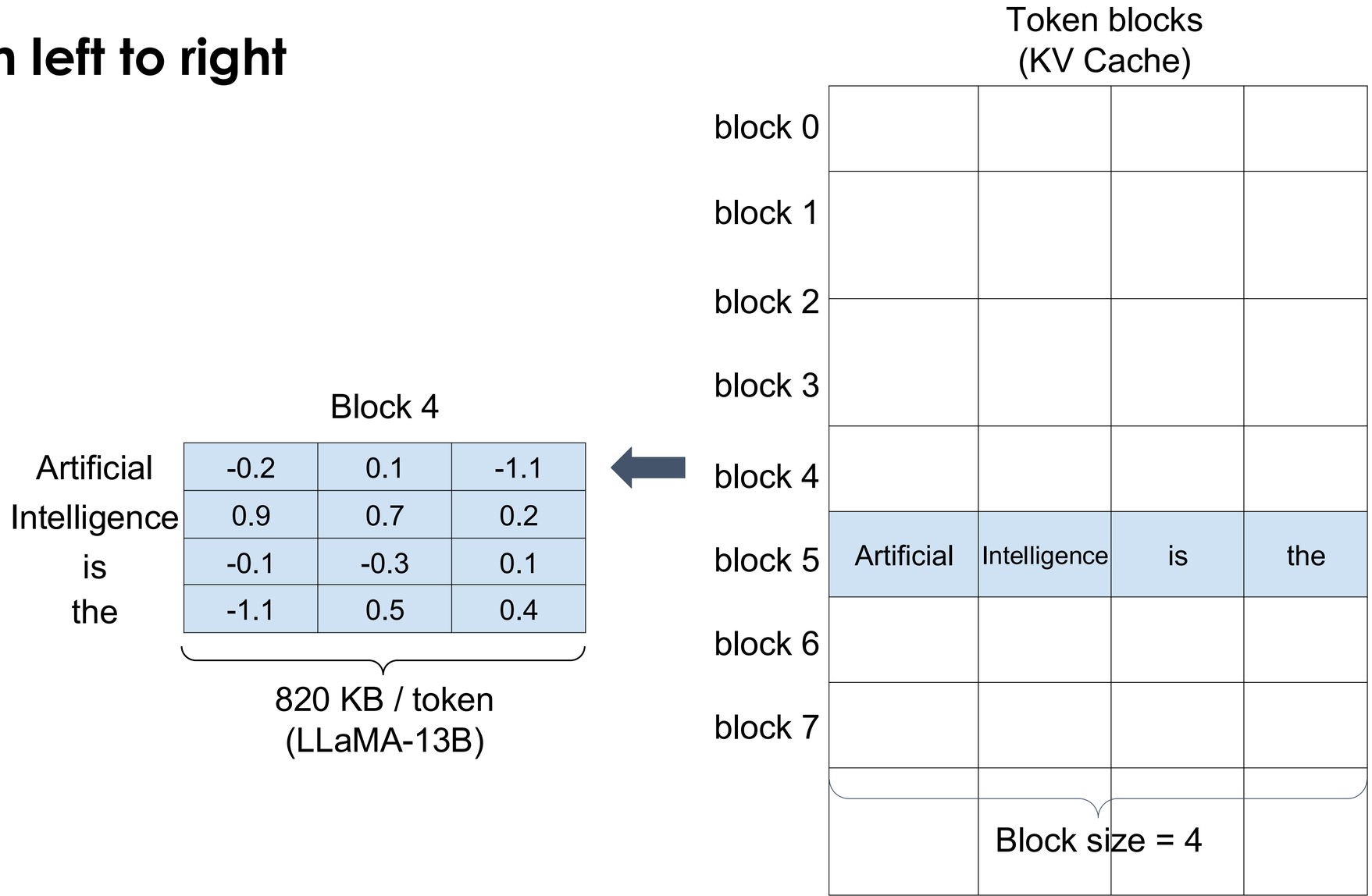


Token block

- A **fixed-size** contiguous chunk of memory that can store token states **from left to right**

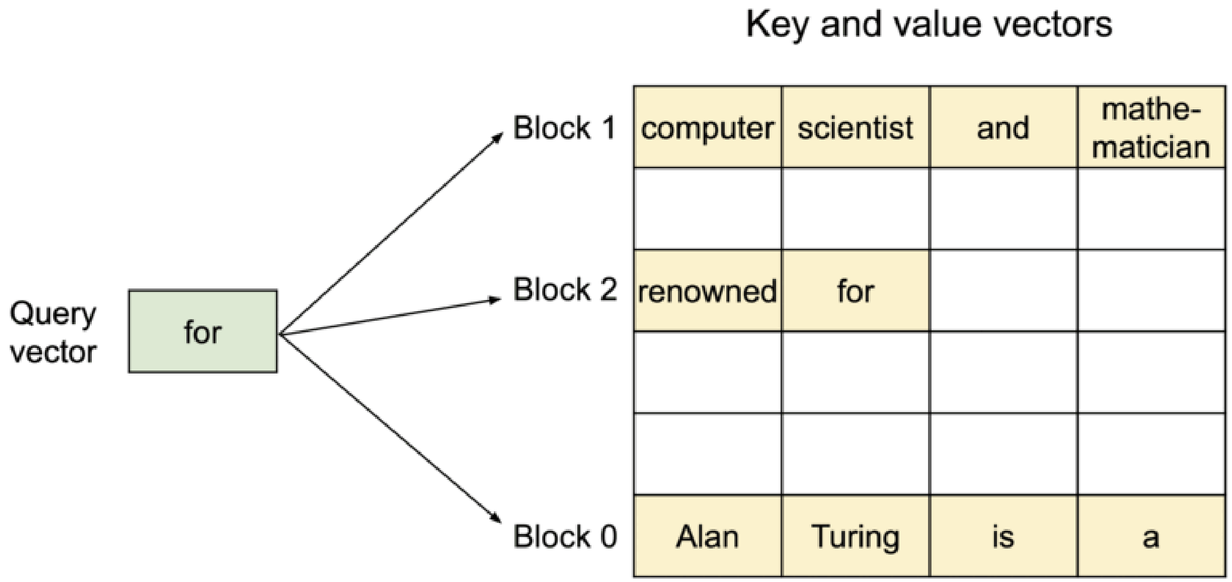
Token block

- A **fixed-size** contiguous chunk of memory that can store token states **from left to right**



Paged Attention

- An attention algorithm that allows for storing continuous keys and values in non-contiguous memory space



Logical & physical token blocks

Request A

Prompt: “Alan Turing is a computer scientist”

Logical token blocks

block 0	Alan	Turing	is	a
block 1	computer	scientist		
block 2				
block 3				

Physical token blocks (KV Cache)

block 0				
block 1				
block 2				
block 3				
block 4				
block 5				
block 6				
block 7				

Logical & physical token blocks

Request A

Prompt: “Alan Turing is a computer scientist”

Logical token blocks

block 0	Alan	Turing	is	a
block 1	computer	scientist		
block 2				
block 3				

Block table

Physical block number	# Filled
7	4
1	2
–	–
–	–

Physical token blocks (KV Cache)

block 0				
block 1	computer	scientist		
block 2				
block 3				
block 4				
block 5				
block 6				
block 7	Alan	Turing	is	a

Logical & physical token blocks

Request A

Prompt: “Alan Turing is a computer scientist”
Completion: “and”

Logical token blocks

block 0	Alan	Turing	is	a
block 1	computer	scientist		
block 2				
block 3				

Block table

Physical block number	# Filled
7	4
1	2
–	–
–	–

Physical token blocks (KV Cache)

block 0				
block 1	computer	scientist		
block 2				
block 3				
block 4				
block 5				
block 6				
block 7	Alan	Turing	is	a

Logical & physical token blocks

Request A

Prompt: “Alan Turing is a computer scientist”
Completion: “and”

Logical token blocks

block 0	Alan	Turing	is	a
block 1	computer	scientist	and	
block 2				
block 3				

Block table

Physical block number	# Filled
7	4
1	2
–	–
–	–

Physical token blocks (KV Cache)

block 0				
block 1	computer	scientist		
block 2				
block 3				
block 4				
block 5				
block 6				
block 7	Alan	Turing	is	a

Logical & physical token blocks

Request A

Prompt: “Alan Turing is a computer scientist”
Completion: “and”

Logical token blocks

block 0	Alan	Turing	is	a
block 1	computer	scientist	and	
block 2				
block 3				

Block table

Physical block number	# Filled
7	4
1	3
–	–
–	–

Physical token blocks (KV Cache)

block 0				
block 1	computer	scientist	and	
block 2				
block 3				
block 4				
block 5				
block 6				
block 7	Alan	Turing	is	a

Logical & physical token blocks

Request A

Prompt: “Alan Turing is a computer scientist”
Completion: “and mathematician”

Logical token blocks

block 0	Alan	Turing	is	a
block 1	computer	scientist	and	mathema tician
block 2				
block 3				

Block table

Physical block number	# Filled
7	4
1	4
–	–
–	–

Physical token blocks (KV Cache)

block 0				
block 1	computer	scientist	and	mathem atician
block 2				
block 3				
block 4				
block 5				
block 6				
block 7	Alan	Turing	is	a

Logical & physical token blocks

Request A

Prompt: “Alan Turing is a computer scientist”
Completion: “and mathematician renowned”

Logical token blocks

block 0	Alan	Turing	is	a
block 1	computer	scientist	and	mathema tician
block 2	renowned			
block 3				

Block table

Physical block number	# Filled
7	4
1	4
5	1
–	–

Physical token blocks (KV Cache)

block 0				
block 1	computer	scientist	and	mathem atician
block 2				
block 3				
block 4				
block 5	renowned			
block 6				
block 7	Alan	Turing	is	a

Allocated on demand

Serving multiple requests

Request A

Block Table

Logical token blocks

Alan	Turing	is	a
computer	scientist	and	mathematician
renowned			

Physical token blocks
(KV Cache)

computer	scientist	and	mathematician
Artificial	Intelligence	is	the
renowned			
future	of	technology	
Alan	Turing	is	a

Block Table

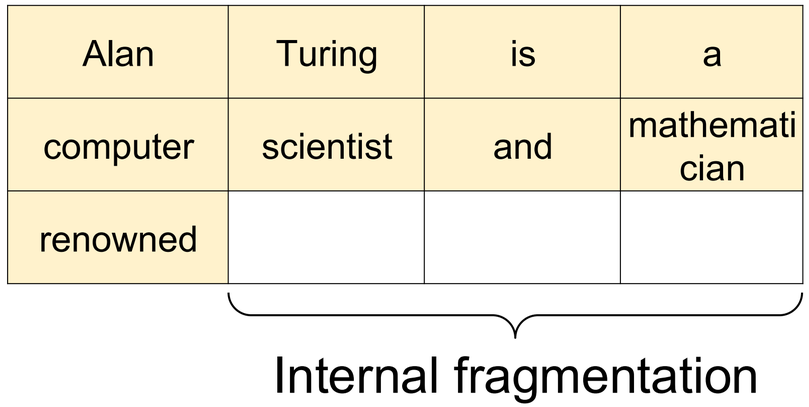
Request B

Logical token blocks

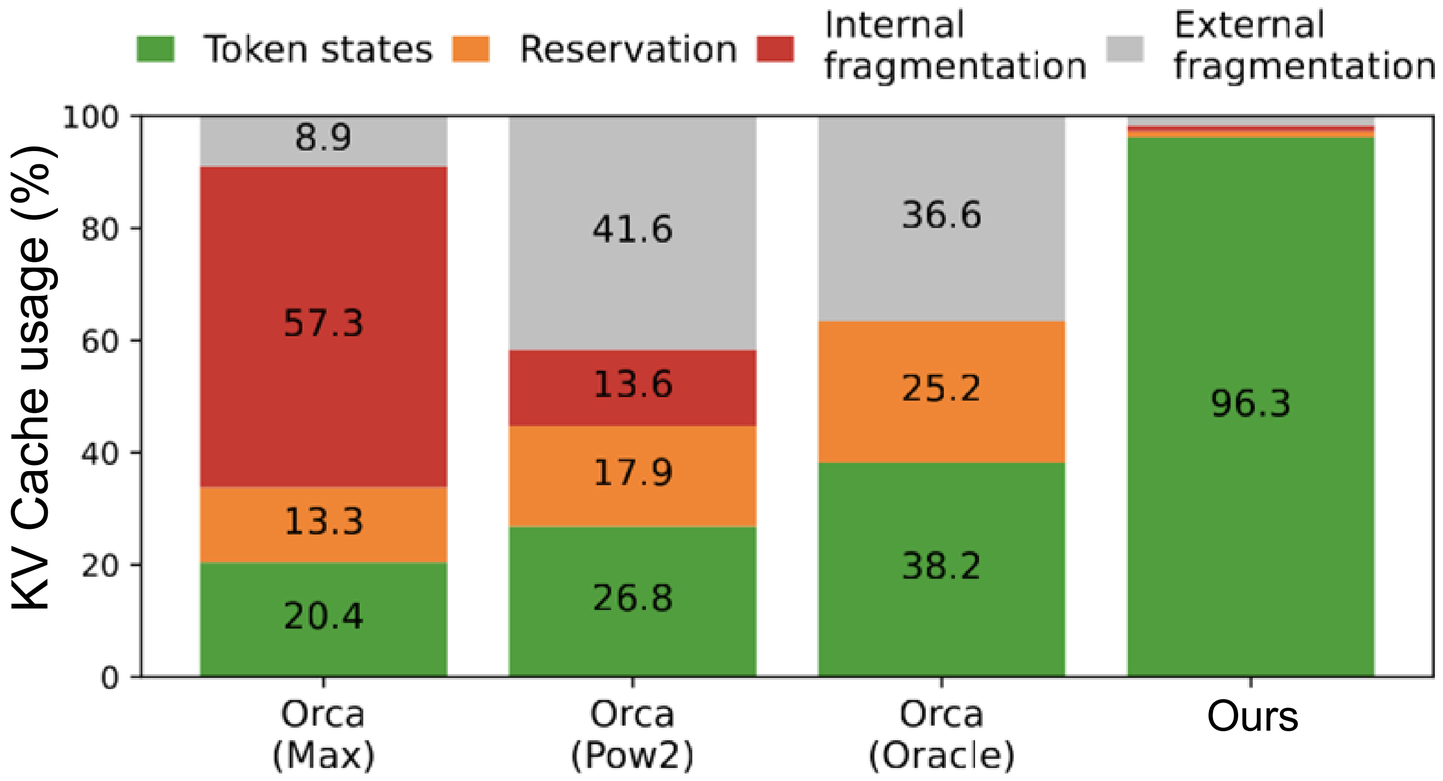
Artificial	Intelligence	is	the
future	of	technology	

Memory efficiency of vLLM

- Minimal internal fragmentation
  - Only happens at the last block of a sequence
  - **# wasted tokens / seq < block size**
    - Sequence: O(100) – O(1000) tokens
    - Block size: 16 or 32 tokens
- No external fragmentation



Effectiveness of PagedAttention



96.3% KV cache utilization

Other Inference Techniques

- Speculative Decoding
  - optimization for small bs, decoding multiple tokens per fwd pass
- Chunked prefill
  - Continuous batching enhancement
- Disaggregated Serving
- Prefix caching
- More…

Focus of the rest of lectures

Data

Model

Compute

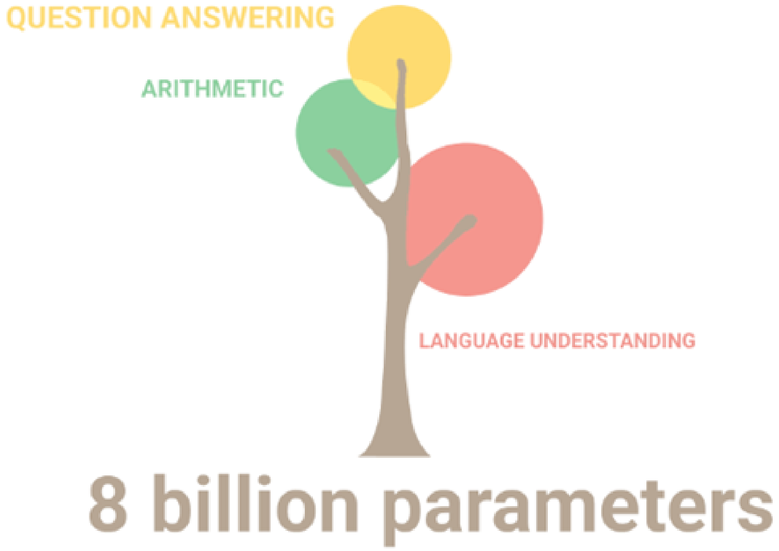
✅ $\{x_i\}^n{}_{i=1}$

✅ Math primitives (mostly matmul)

✅ A repr that expresses the computation using primitives

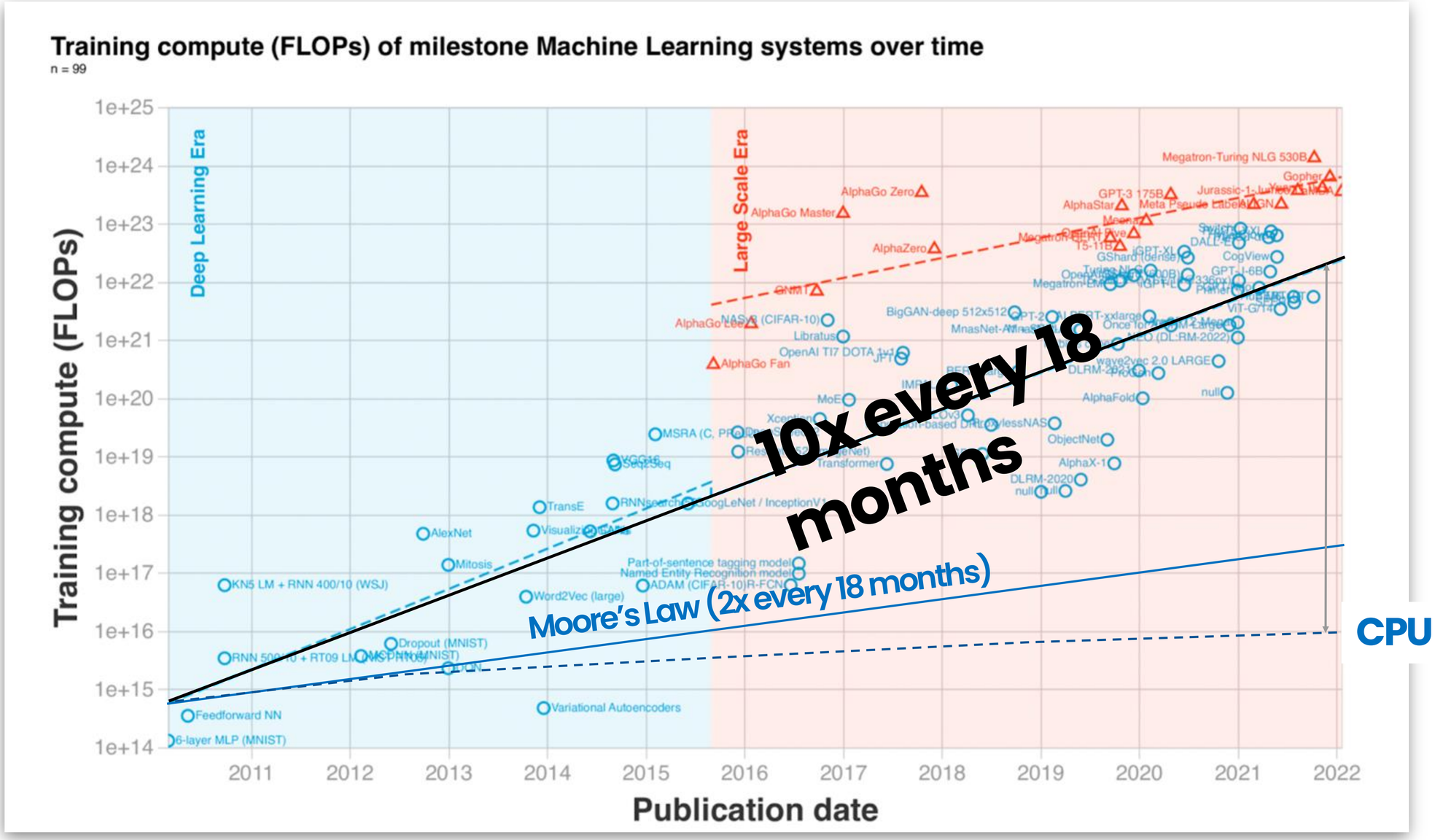
❓Make **LLMs** run on (**large clusters** of) **GPUs**

Big Models have Emergent capabilities



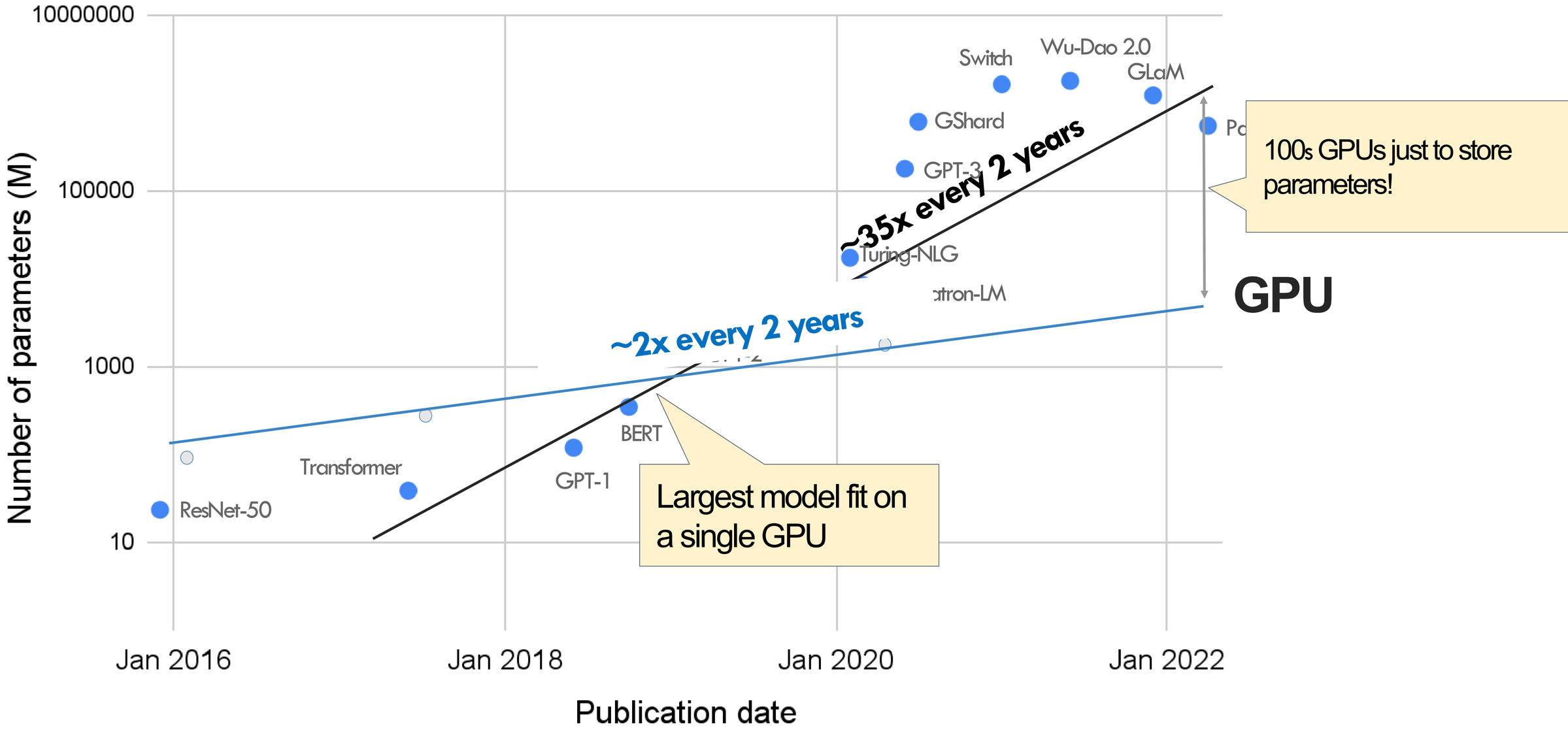
"Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance",
S Narang, A Chowdhery et al, https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html

Growing gap between demand and supply



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

Growing gap between memory demand and supply

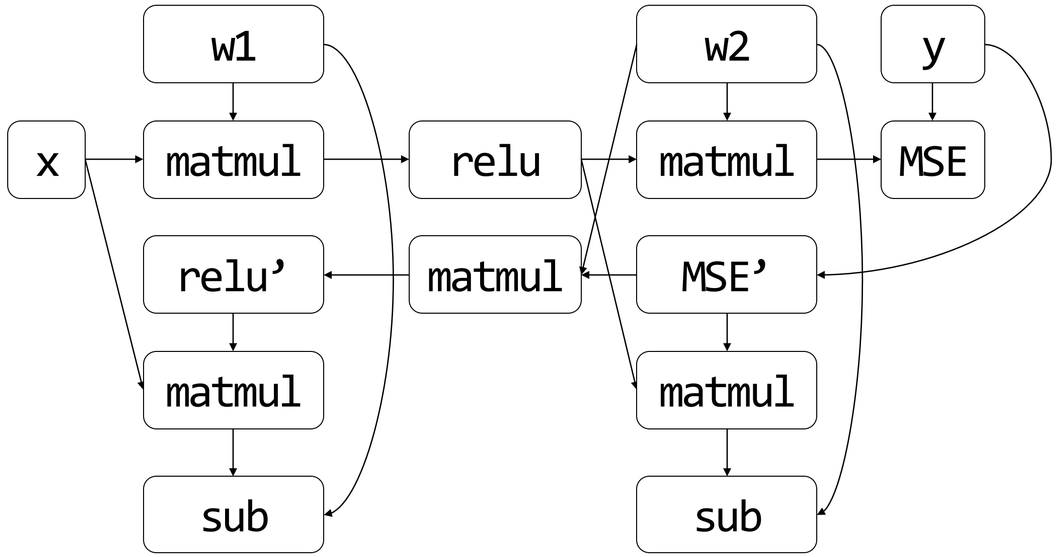


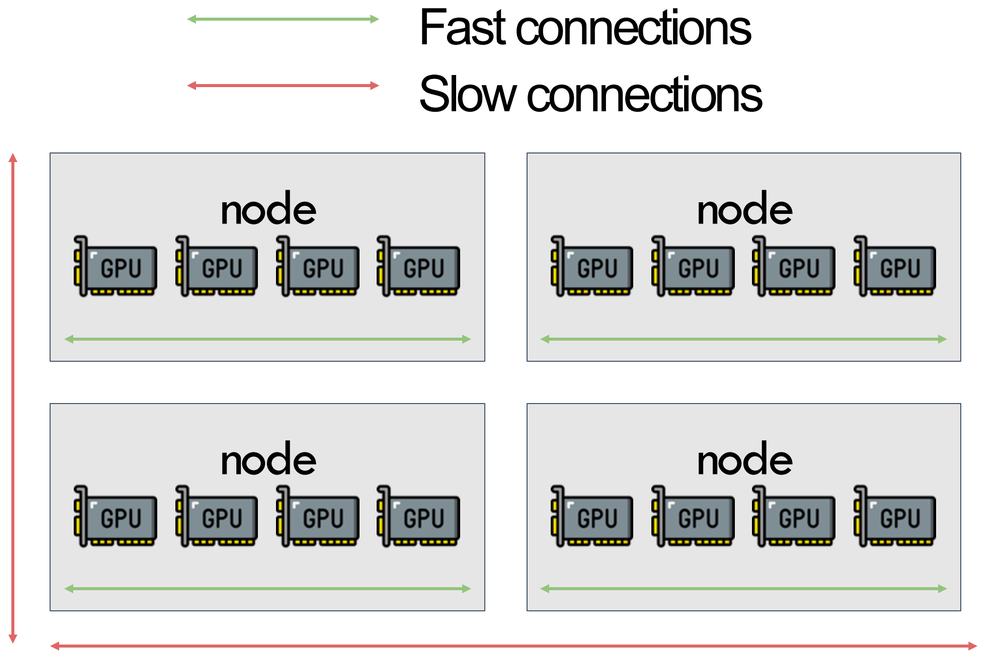
No way out but to parallelize these workloads !

Parallelization

- Goal: parallelize the graph compute over multiple devices

How to partition the computational graph on the device cluster?

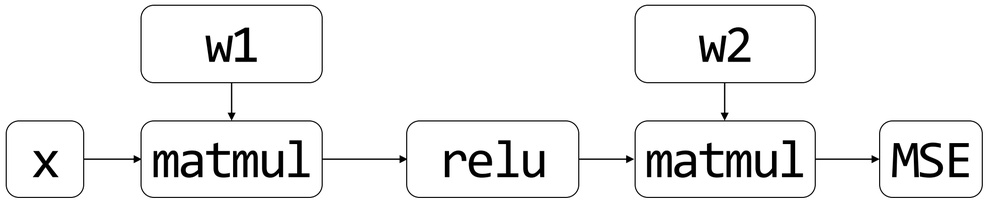


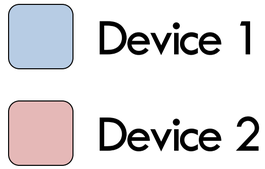


Parallelization Problems

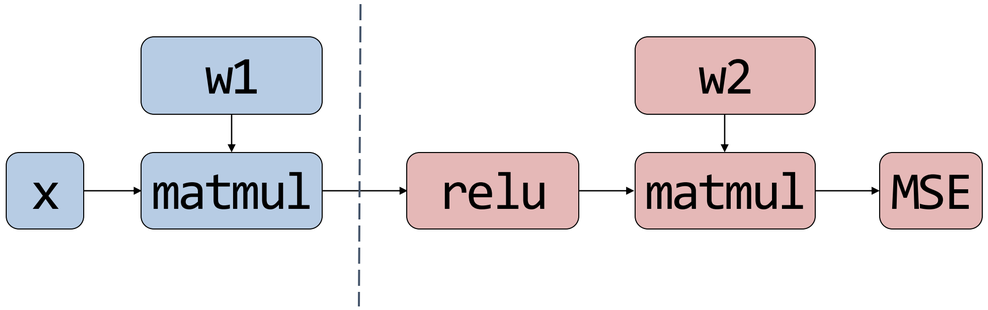
- **How to partition**
- **How to communicate**
- **How to schedule**
- Consistency
- How to auto-parallelize?

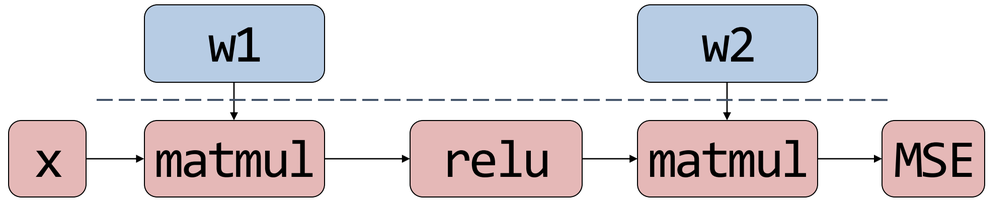
Partitioning Computation Graph





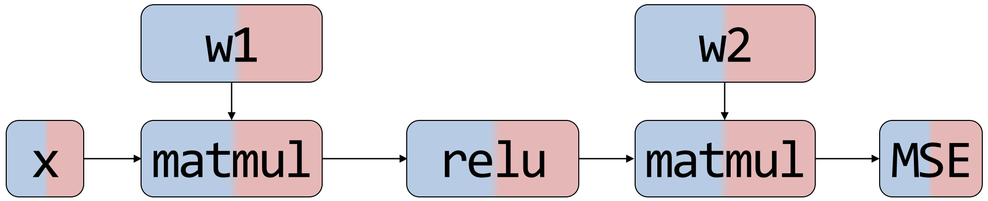
Strategy 1

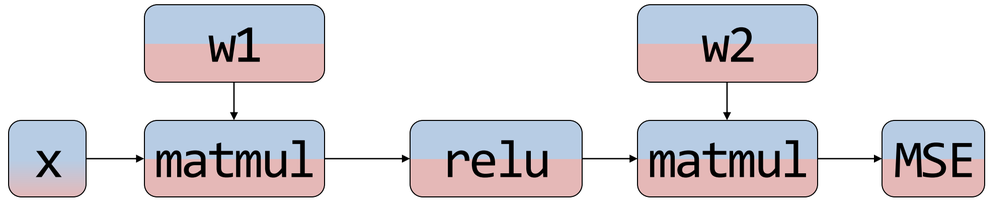




...

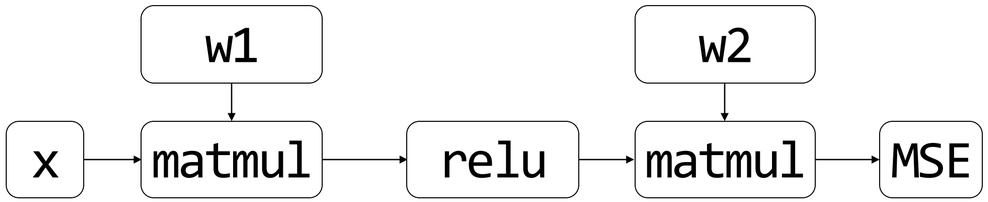
Strategy 2

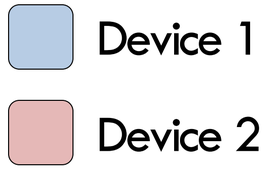




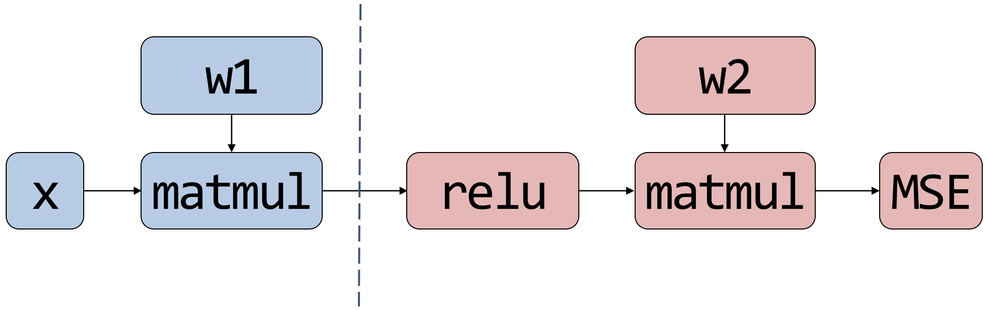
...

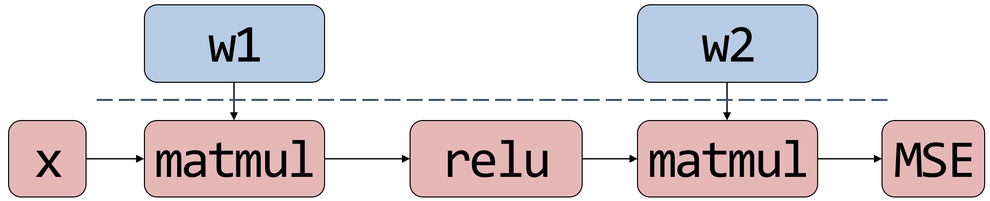
Partitioning Computation Graph





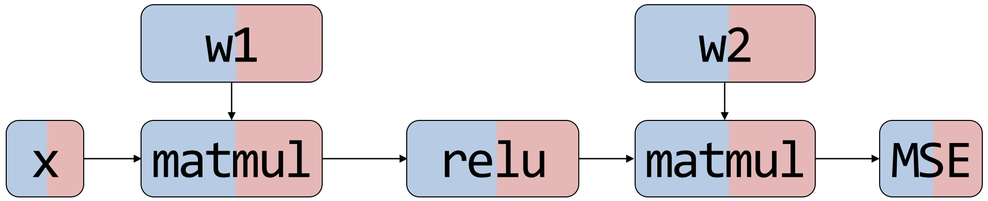
Strategy 1: Inter-operator Parallelism

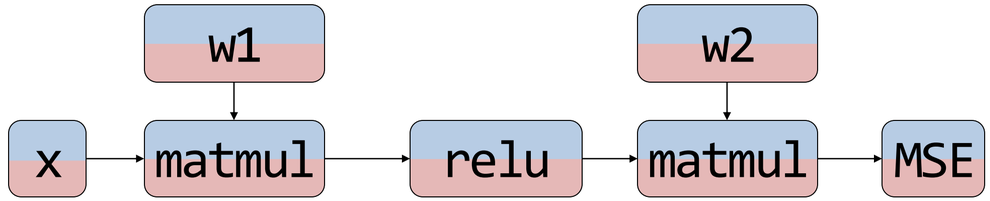




...

Strategy 2: Intra-operator Parallelism



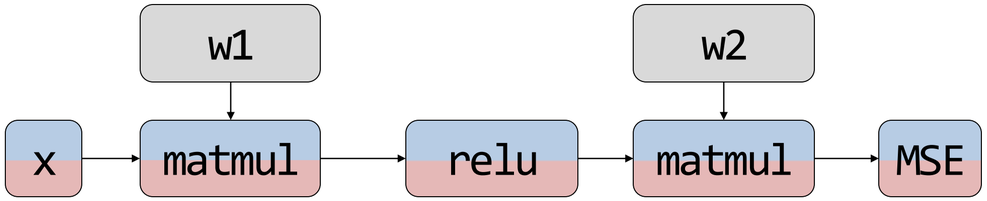


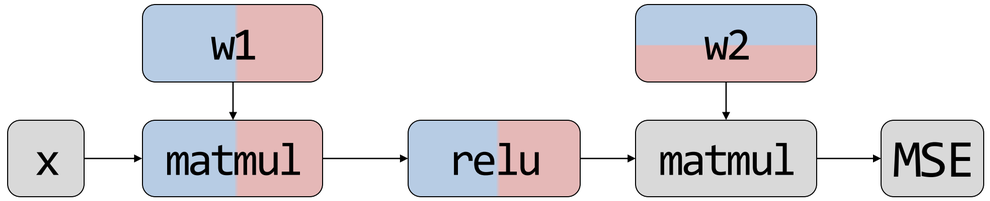
...

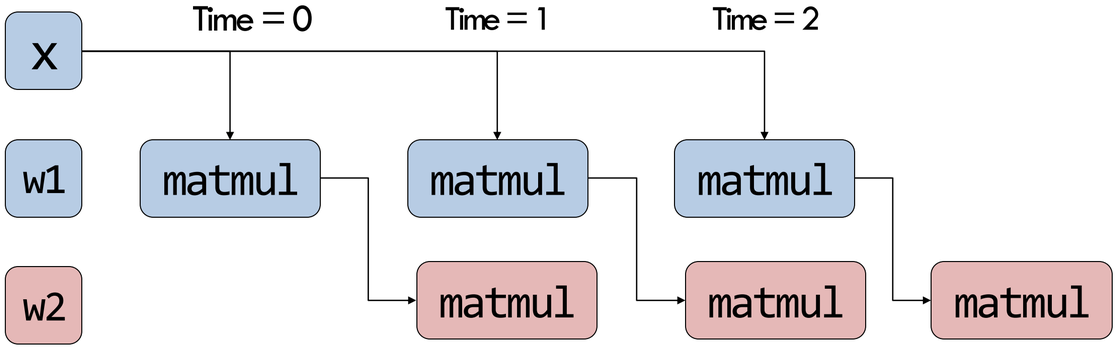
More Parallelisms…

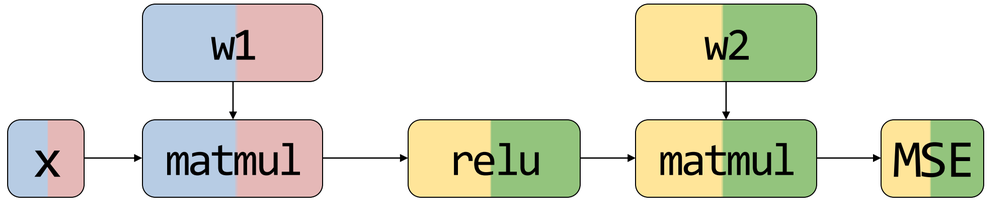
Multiple intra-op strategies for a single node

More strategies

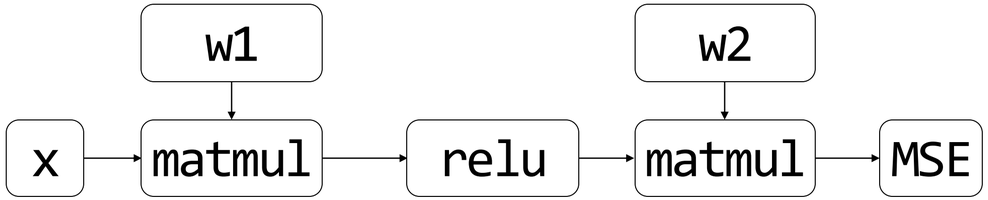


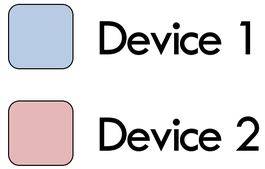




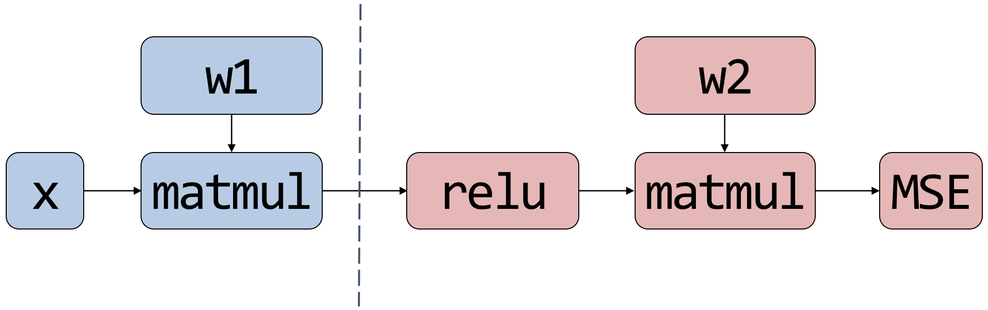


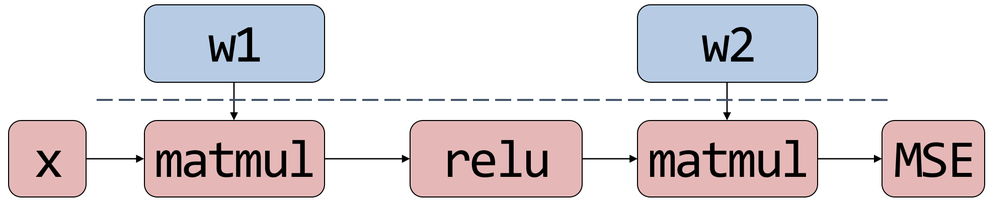
Summary: Inter-op and Intra-op Parallelisms





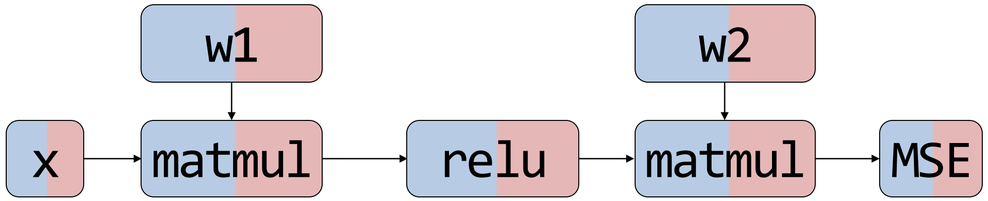
Inter-op parallelism: Assign different operators to different devices.

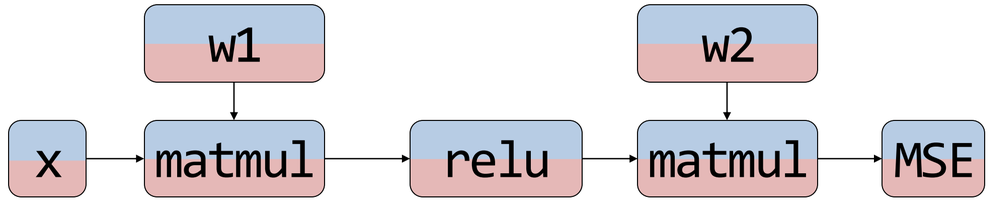




...

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

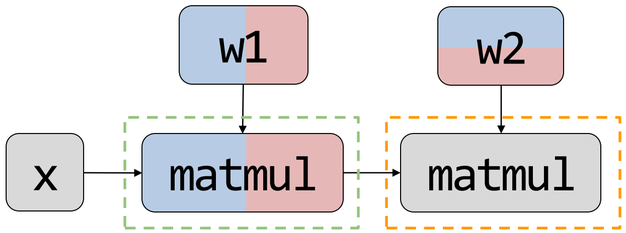


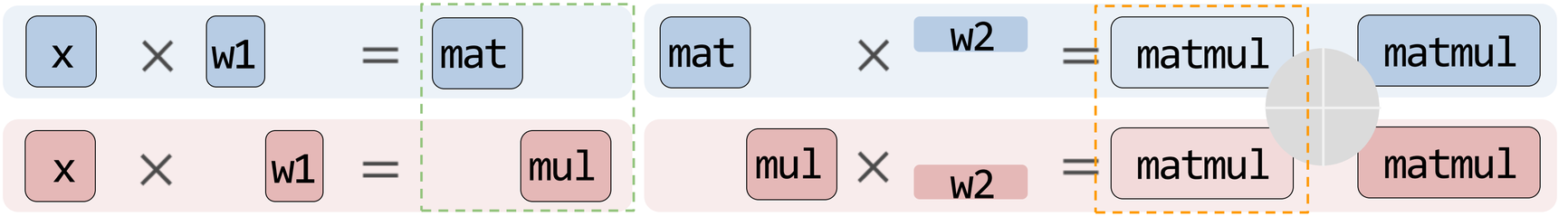


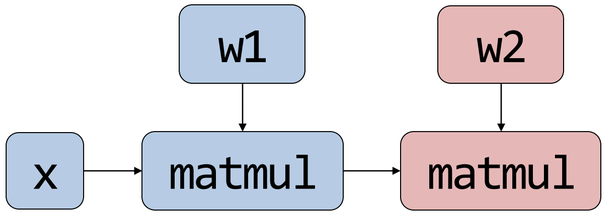
...

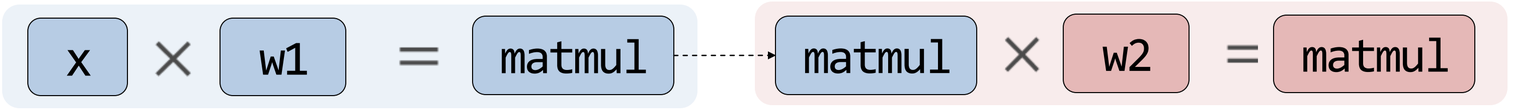
Inside Intra- and Inter-op Parallelism

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

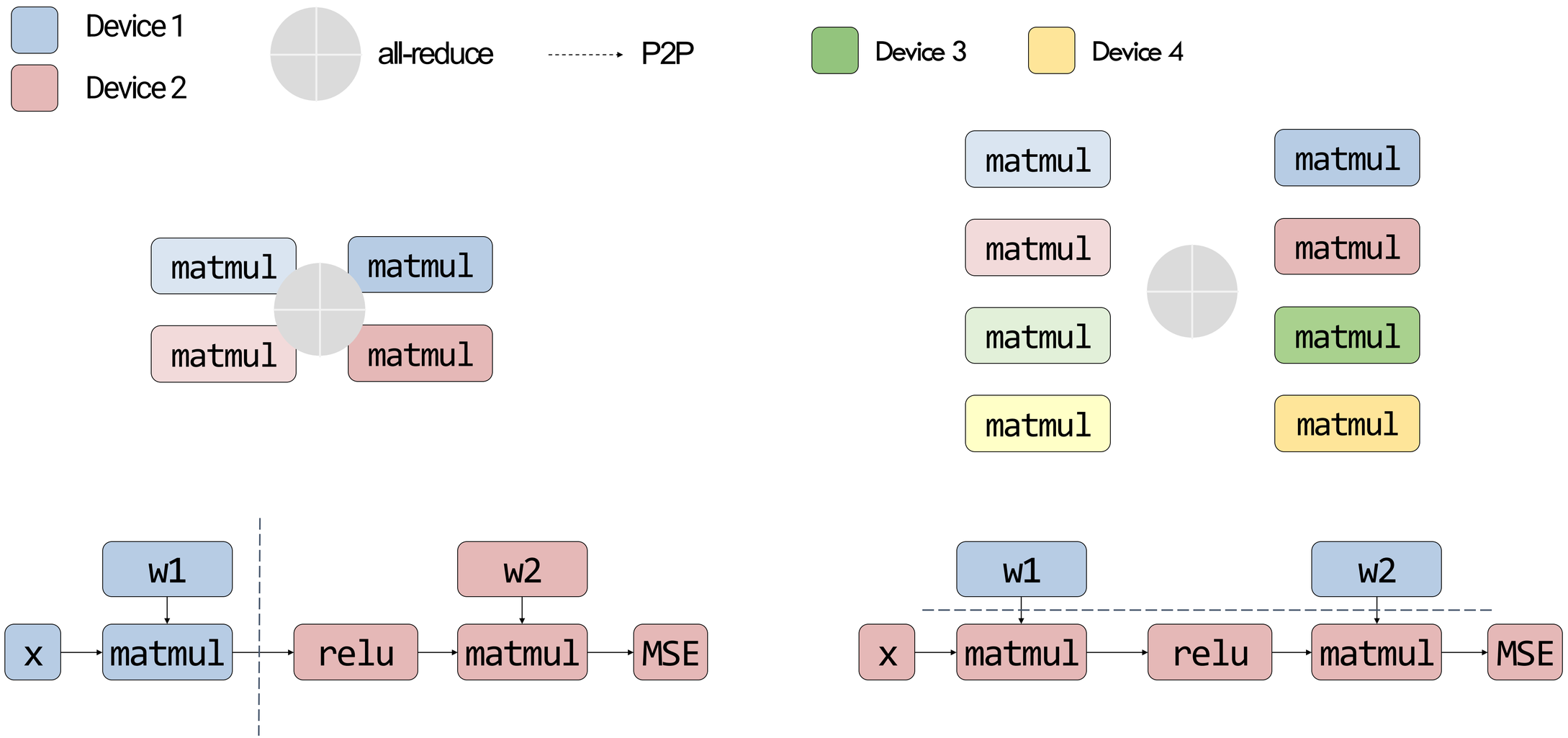




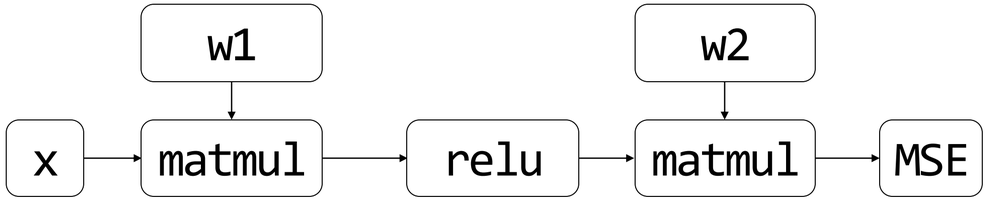


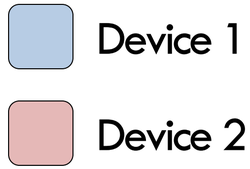


Looking Into the Communication

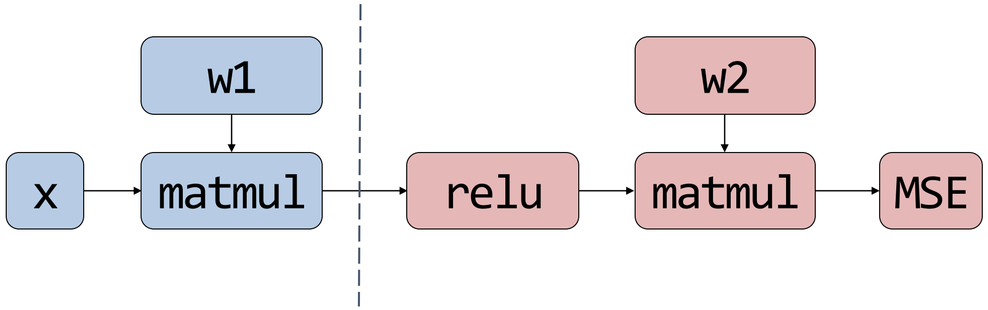


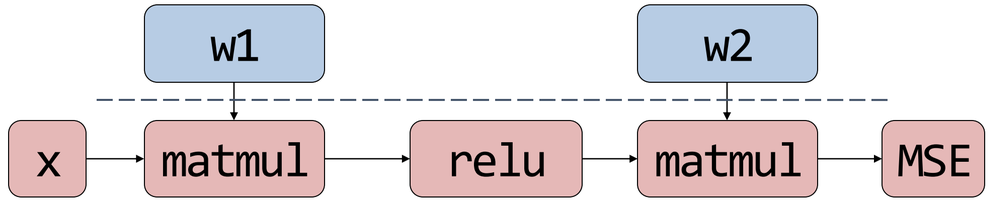
Parallelism: Key Characteristics





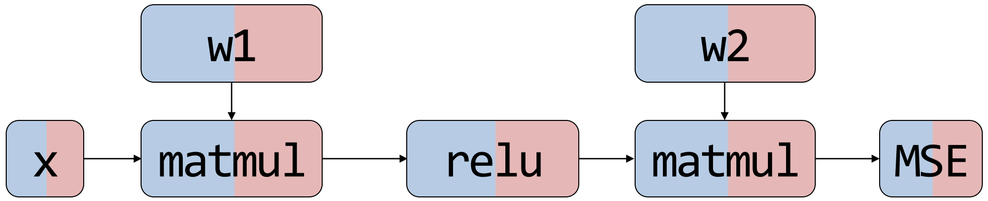
Inter-op parallelism: Requires point-to-point communication but results in device idle

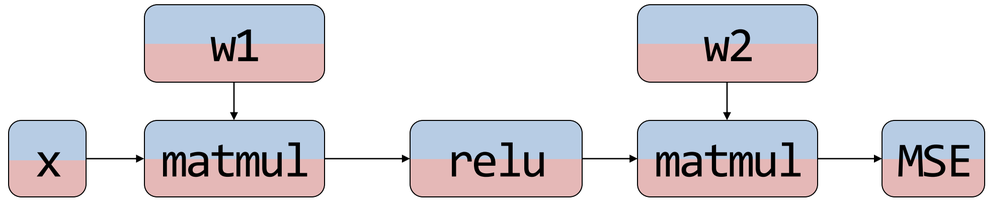




...

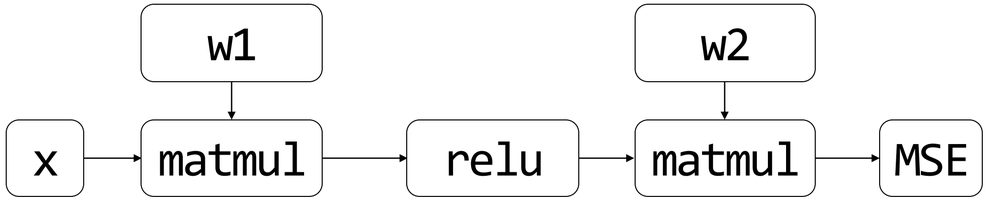
Intra-op parallelism: Devices are busy but requires collective communication

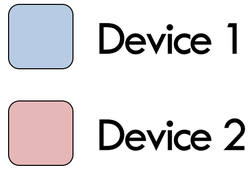




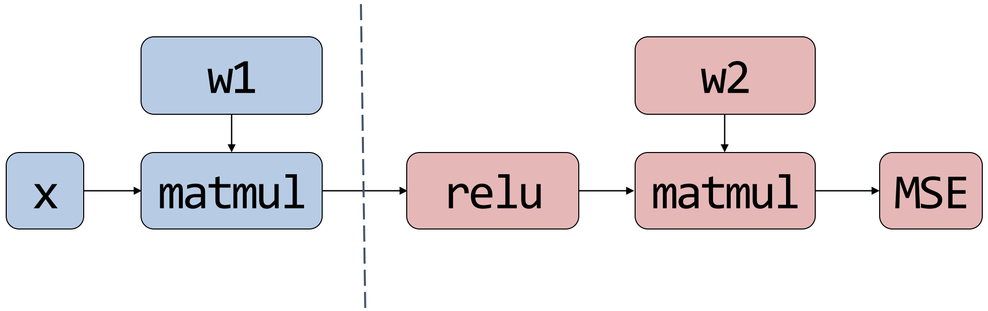
...

Inter-op and Intra-op Parallelism: Characteristics [Important]

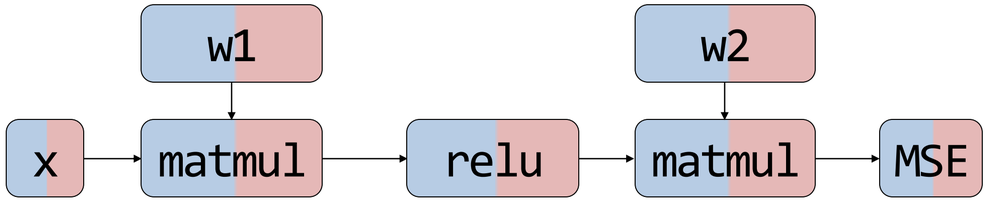




Inter-op parallelism



Intra-op parallelism



Trade-off

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

Computational View of ML Parallelisms

Classic view

Data parallelism

Model parallelism

New view (this class)

Inter-op parallelism

Intra-op parallelism

Two Views of ML Parallelisms

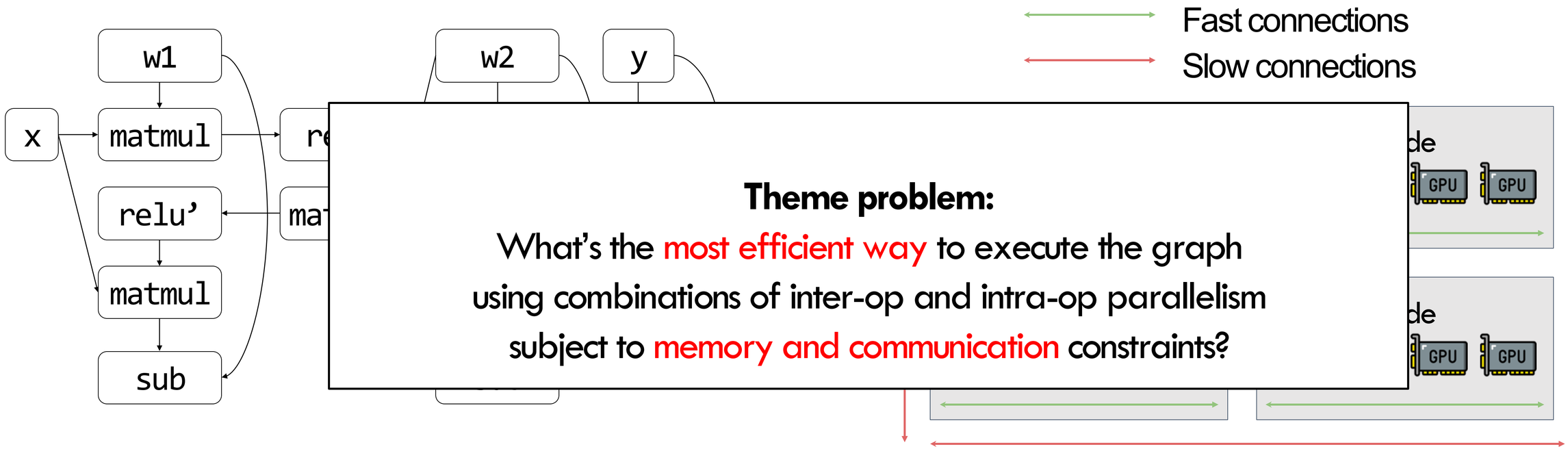
Data and model parallelism

- 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.

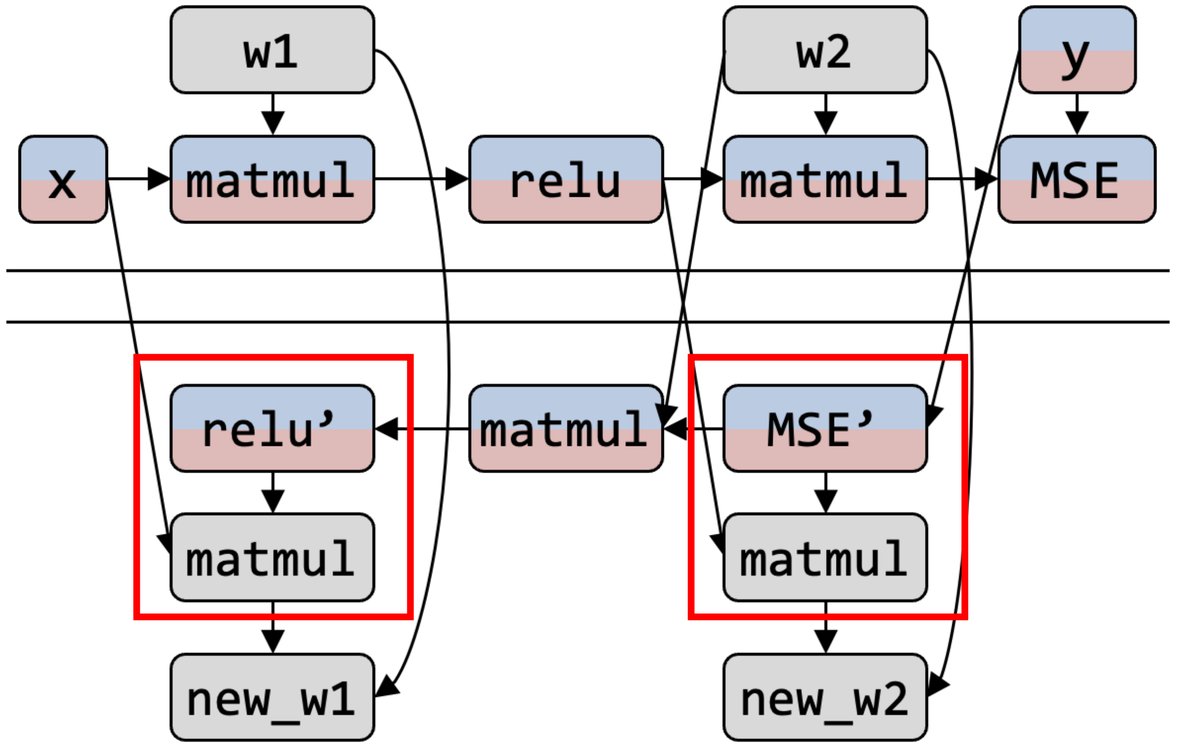
New: Inter-op and Intra-op parallelism.

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

ML Parallelization under New View



Data Parallelism



How to implement this communication?

Two Solutions

- Parameter Server
- AllReduce
- Key assumption:
  - The model can fit into an (GPU) worker memory hence we can create many replica