DATA+AI SUMMIT 2022

Scaling Deep Learning

On Databricks

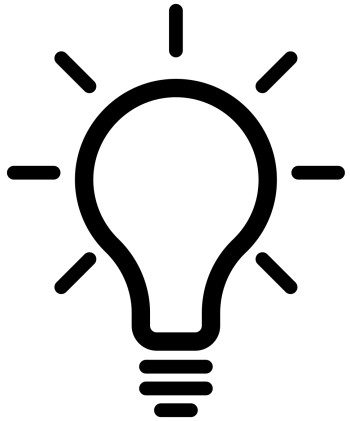
Brian Law

Specialist Solution Architect, Databricks

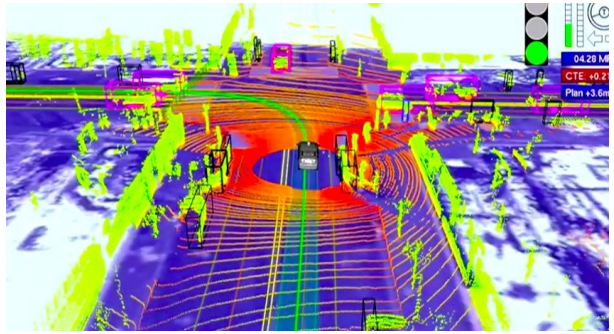
ORGANIZED BY databricks

This talk is not about:

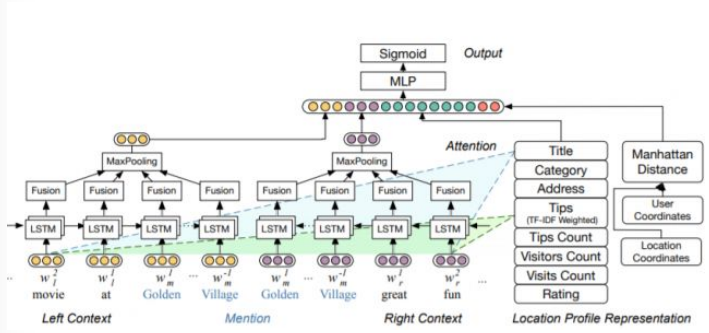
Ideation



Deep Dives into Major Areas

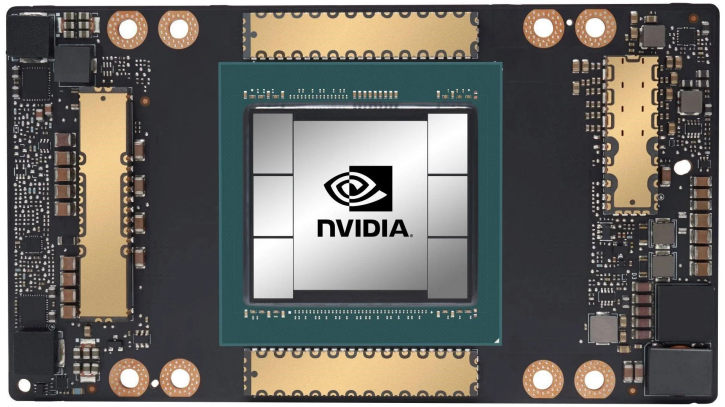


SoTA models and trends

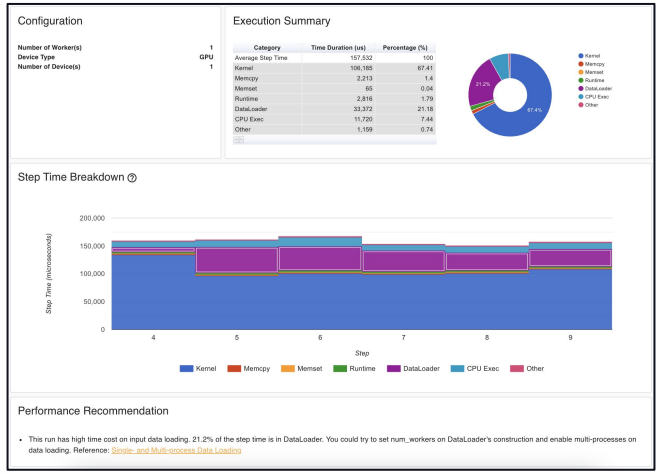


Today we will talk about:

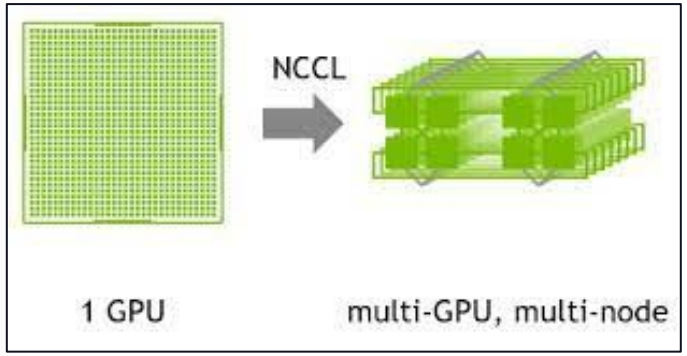
Hardware



Code Profiling



Distributed Compute



Agenda

Topics

When to use Deep Learning?

Why Scale Deep Learning?

Life of a Tuple in Deep Learning

- Hardware
- Training Loop

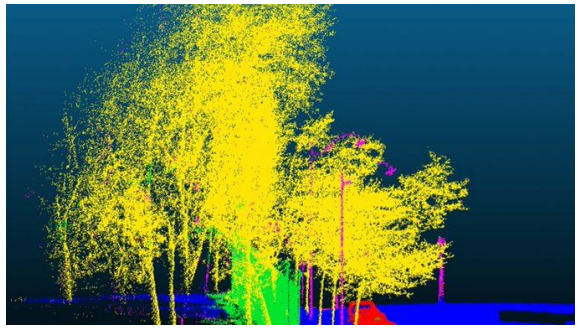
How to structure your code?

Understanding Performance

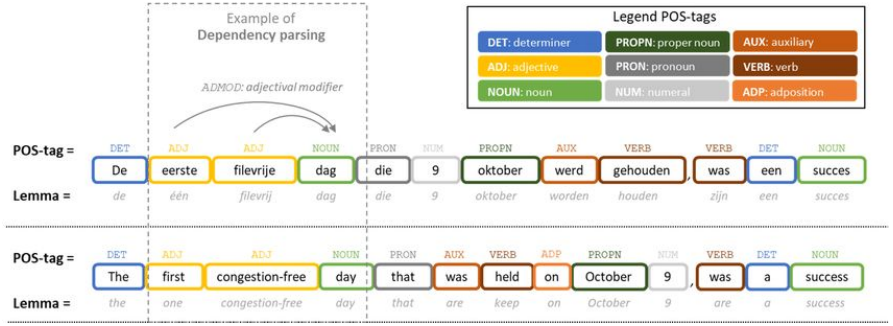
When to use Deep Learning

Slam Dunk use cases

Computer Vision



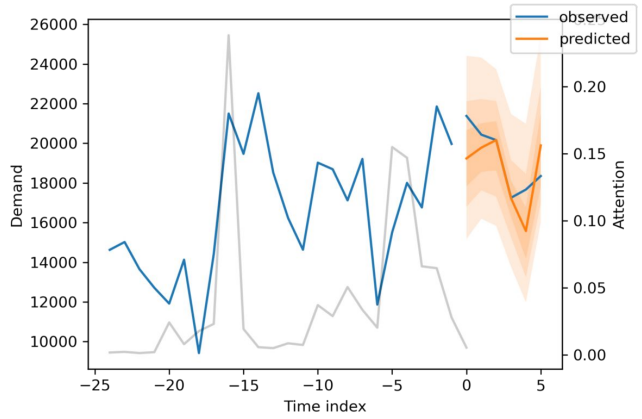
Natural Language Processing



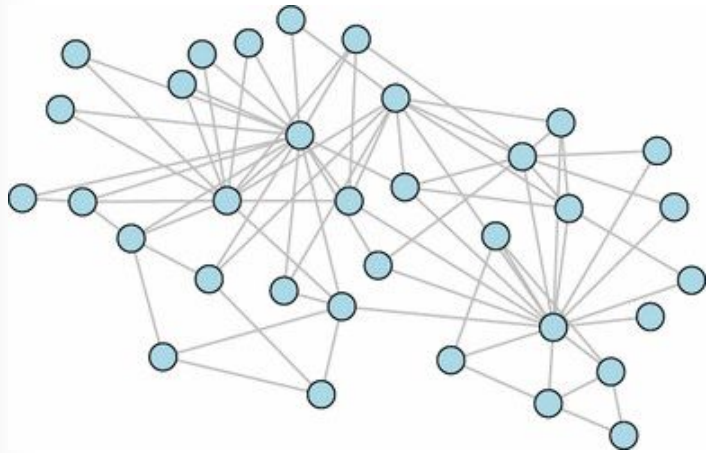
When to use Deep Learning

Exploratory

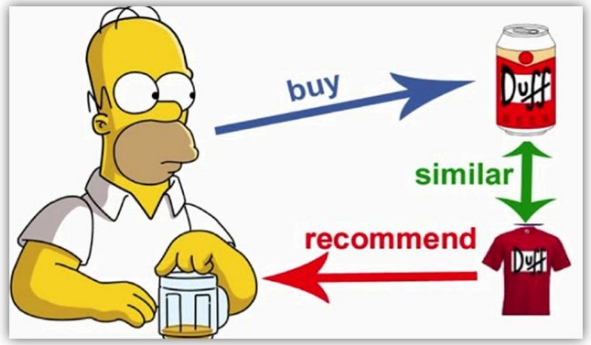
Timeseries



Graphs



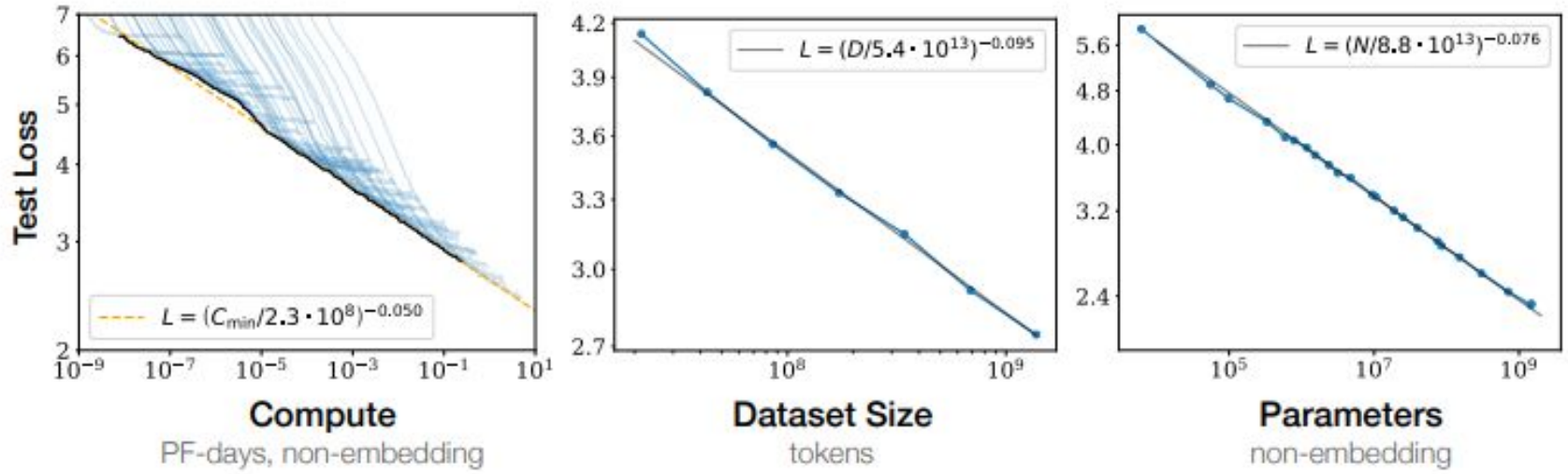
Recommenders



Why Scale Deep Learning?

https://arxiv.org/pdf/2001.08361.pdf

Performance Improves with Scale



Why Scale Deep Learning?

Training Takes Time

BLOOM: BigScience 176B Model

BLOOM (*BigScience Language Open-science Open-access Multilingual*): the BigScience 176 billion parameters model is currently training.

The training started on **March 11, 2022 11:42am PST** and will last 3-4 months on the 416 A100 GPUs of the Jean Zay public supercomputer

Follow the training at https://twitter.com/BigScienceLLM
Send questions about the training to bigscience-large-model-training [AT] googlegroups [.] com

Why Scale Deep Learning?

Training Takes Time

BLOOM: BigScience 176B Model

BLOOM (*BigScience Language Open-science Open-access Multilingual*): the BigScience 176 billion parameters model is currently training.

The training started on **March 11, 2022 11:42am PST** and will last 3-4 months on the 416 A100 GPUs of the Jean Zay public supercomputer

Follow the training at https://twitter.com/BigScienceLLM
Send questions about the training to bigscience-large-model[illegible]aining [AT] googlegroups [.] com

3-4 Months on 416 GPUS!

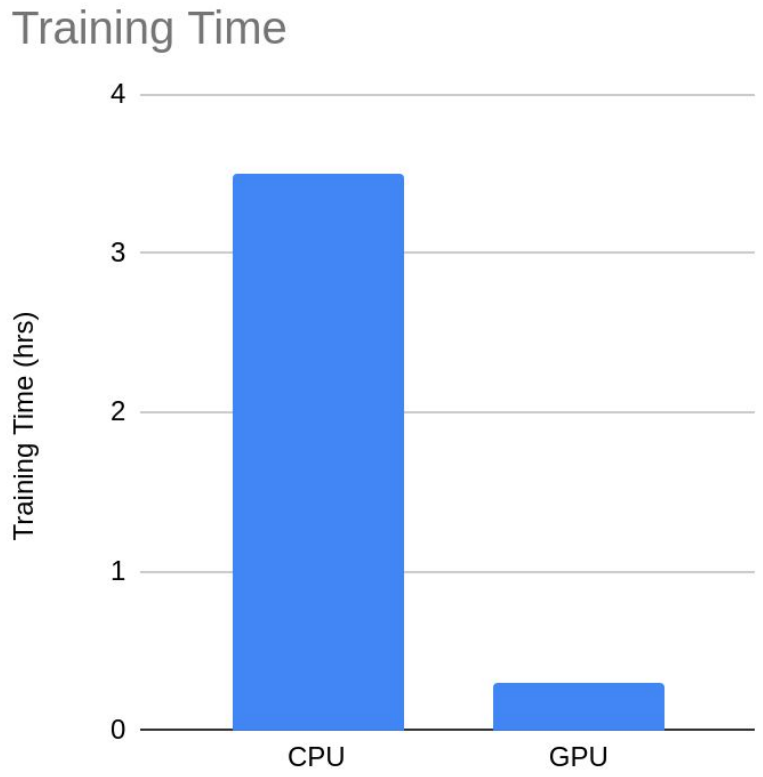
1% per day!

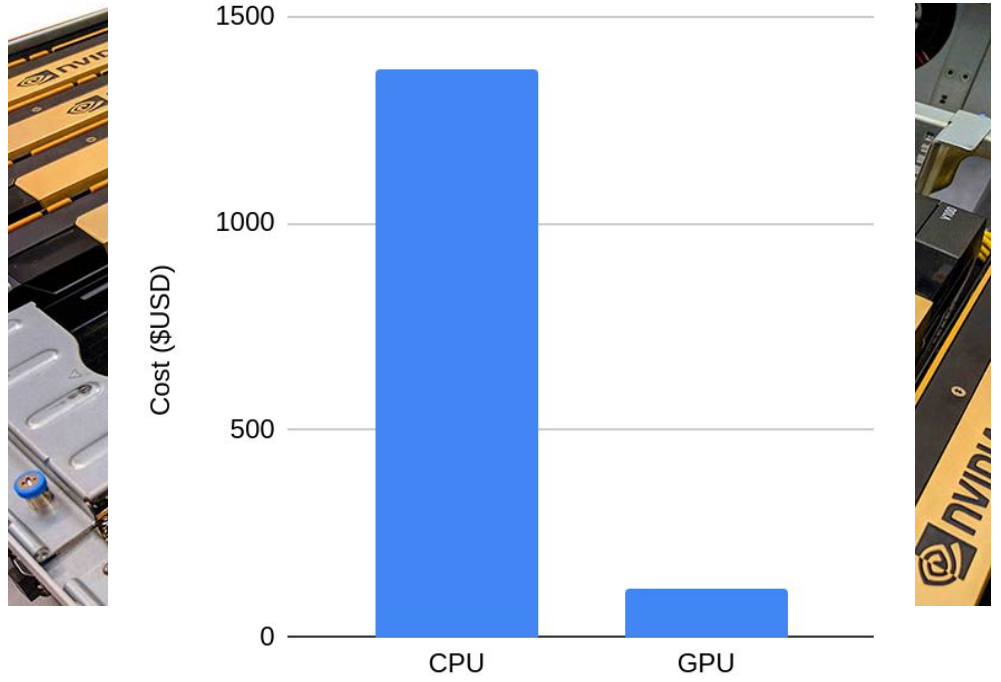
GPU vs CPU

Gpus are complex and expensive

GPU vs CPU

But make a big difference





* From Stanford Dawnbench

Why Scale Deep Learning?

"Normal" models take time too

Computer Vision

Imagenet Dataset:

Resnet50

- 20 mins to train
- 128 GPUs
- 23 million params

From Dawnbench (FastAI training run)

https://dawn.cs.stanford.edu/benchmark/v1/index.html

Natural Language Processing

Various Datasets:

BERT Large

- 4 days to train
- 64 TPU chips
- 340 million params

From: https://arxiv.org/pdf/1810.04805.pdf

Factors in Scaling

- Hardware Config and Node Selection
- Scaling Methodology
- Algorithm

Factors in Scaling

- **Hardware Config and Node Selection**
- **Scaling Methodology**
- Algorithm – will not focus on today

Life of a Tuple in Deep Learning

Main Steps and Hardware transitions

Code Flow

Raw Data → Data Loader → Tensor Batch → Forward / Backward Pass → Weights Update

Hardware Flow

Object Store → Compute Instance → CPU Compute → GPU Compute

Life of a Tuple in Deep Learning

Main Steps and Hardware transitions

Code Flow

Raw Data → Data Loader → Tensor Batch → Forward / Backward Pass → Weights Update

Hardware Flow

Object Store → Compute Instance → CPU Compute → GPU Compute

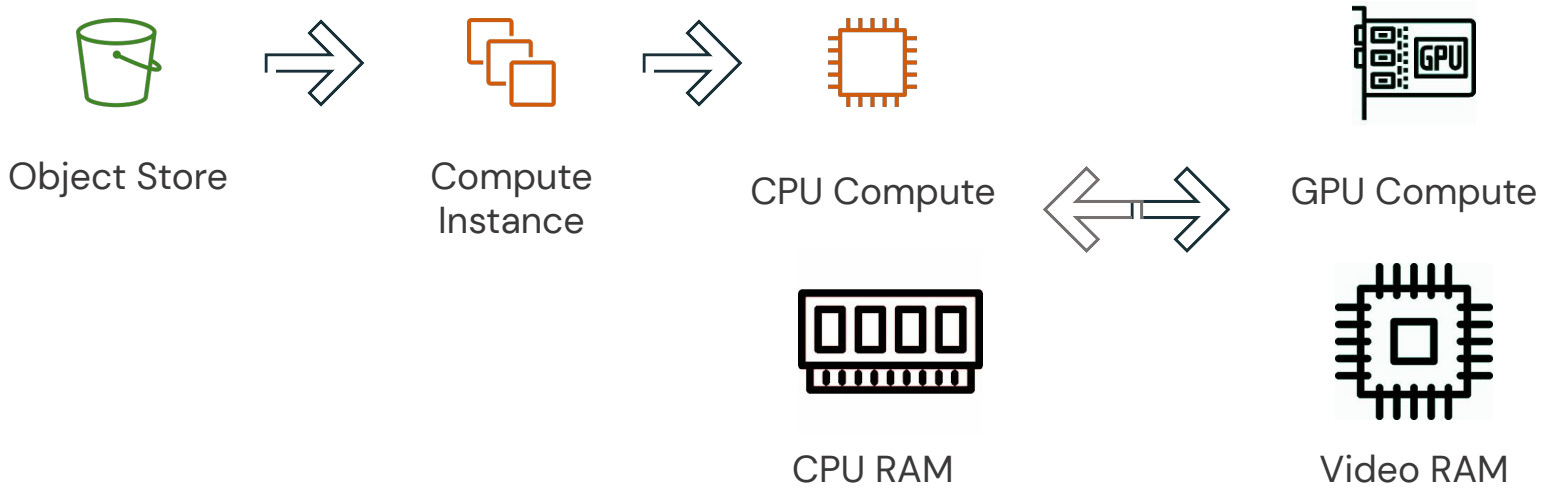
Goals in Scaling

To train models fast (and cheap) we need to:

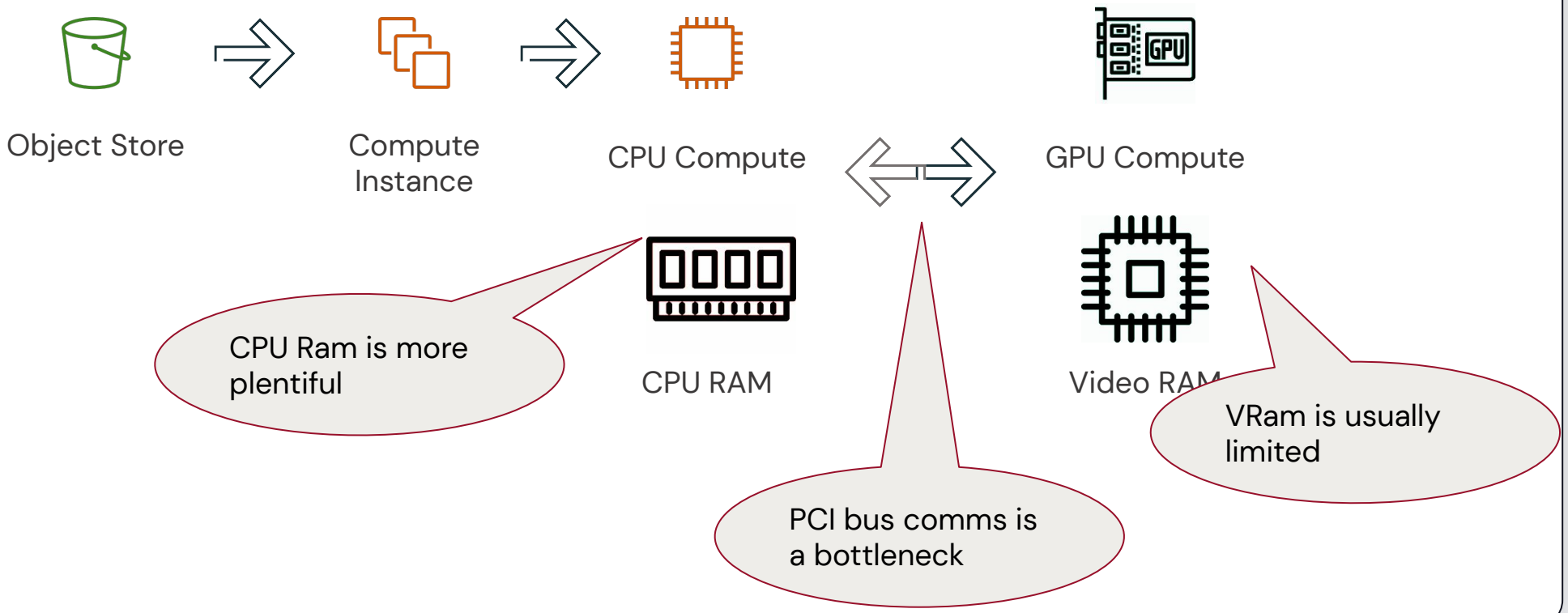
- Max out GPU Consumption (Most expensive / hr)
- Get through a Training Epoch as fast as possible

Understanding Hardware

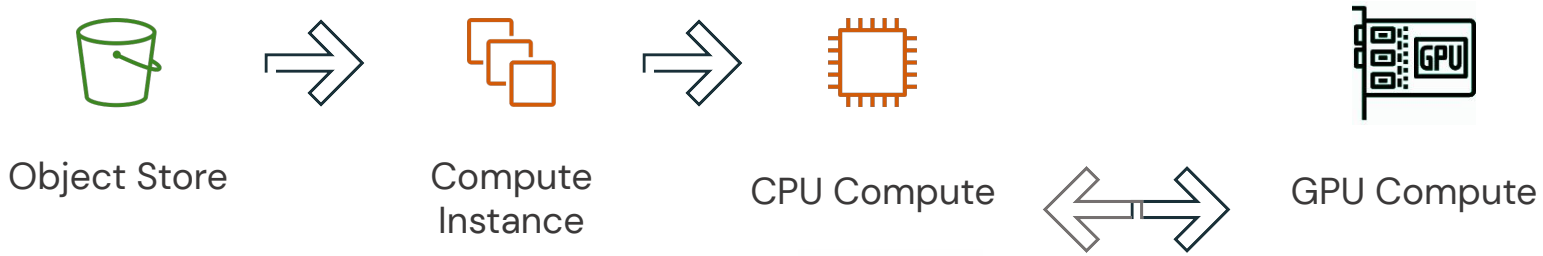
Exploring the Hardware Flow



Exploring the Hardware Flow



Exploring the Hardware Flow



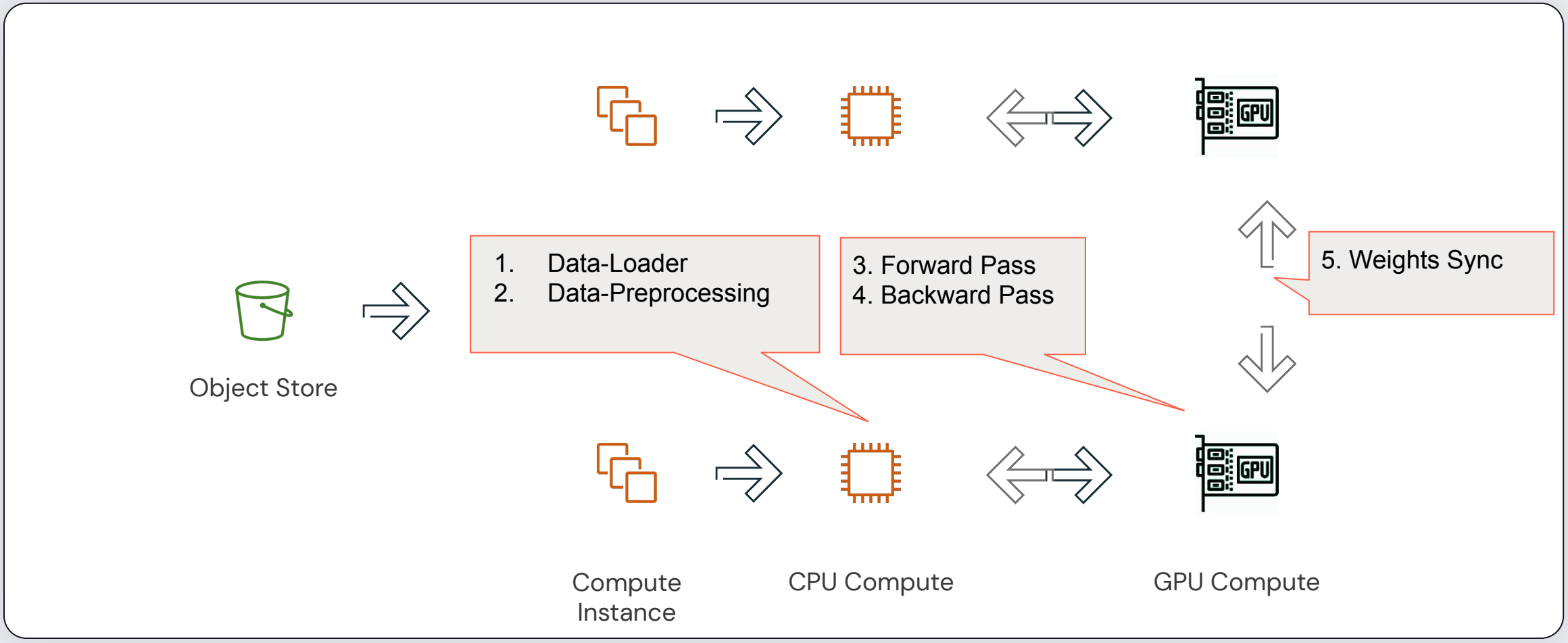
DataLoader

Data-Preprocessing

Model Calculations
Forward Backward Pass

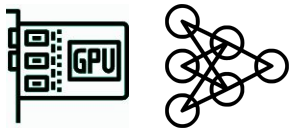
Exploring the Hardware Flow

From Single Node to multi-node



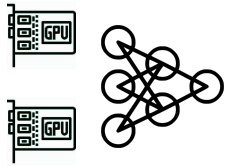
Scaling models

GPU Scaling Paradigms



Data Parallel

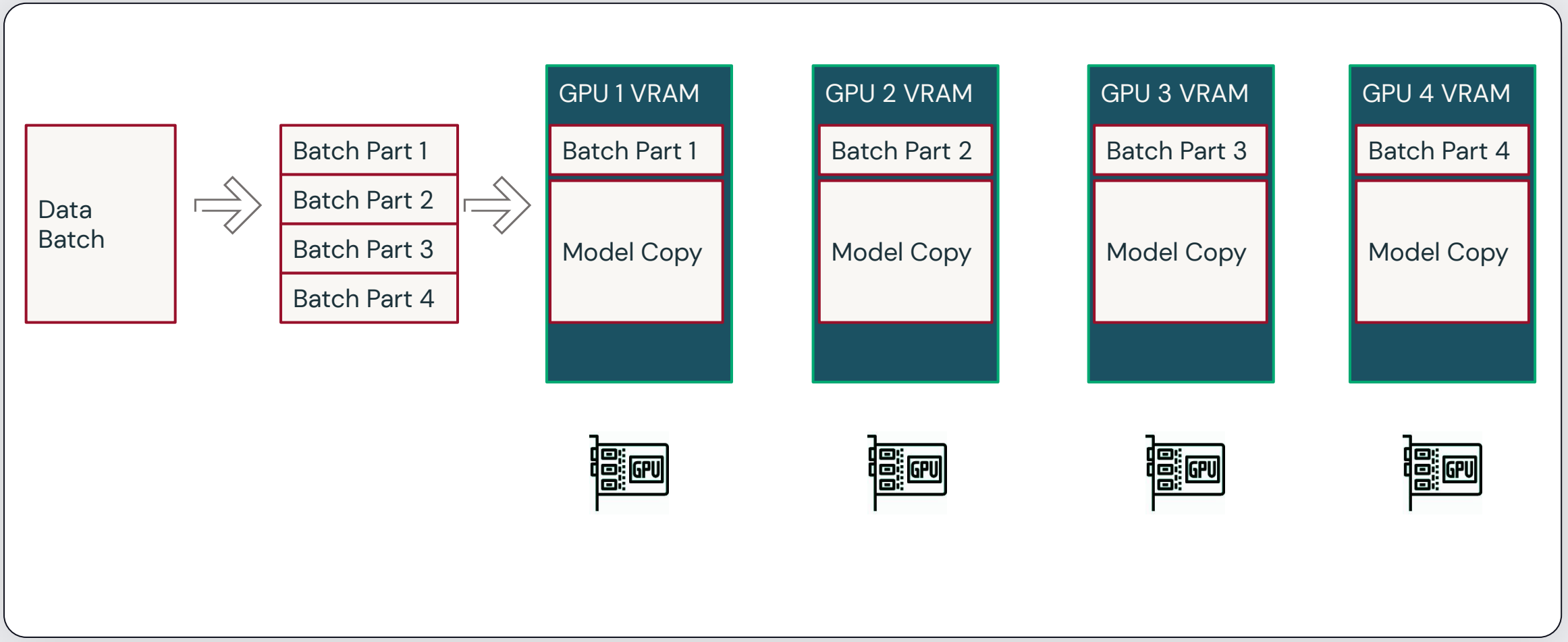
One full copy of the model per GPU



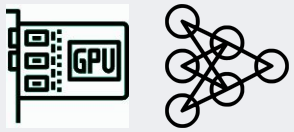
Model Parallel

Model is split across GPUs due to VRAM constraints.

Data Parallel

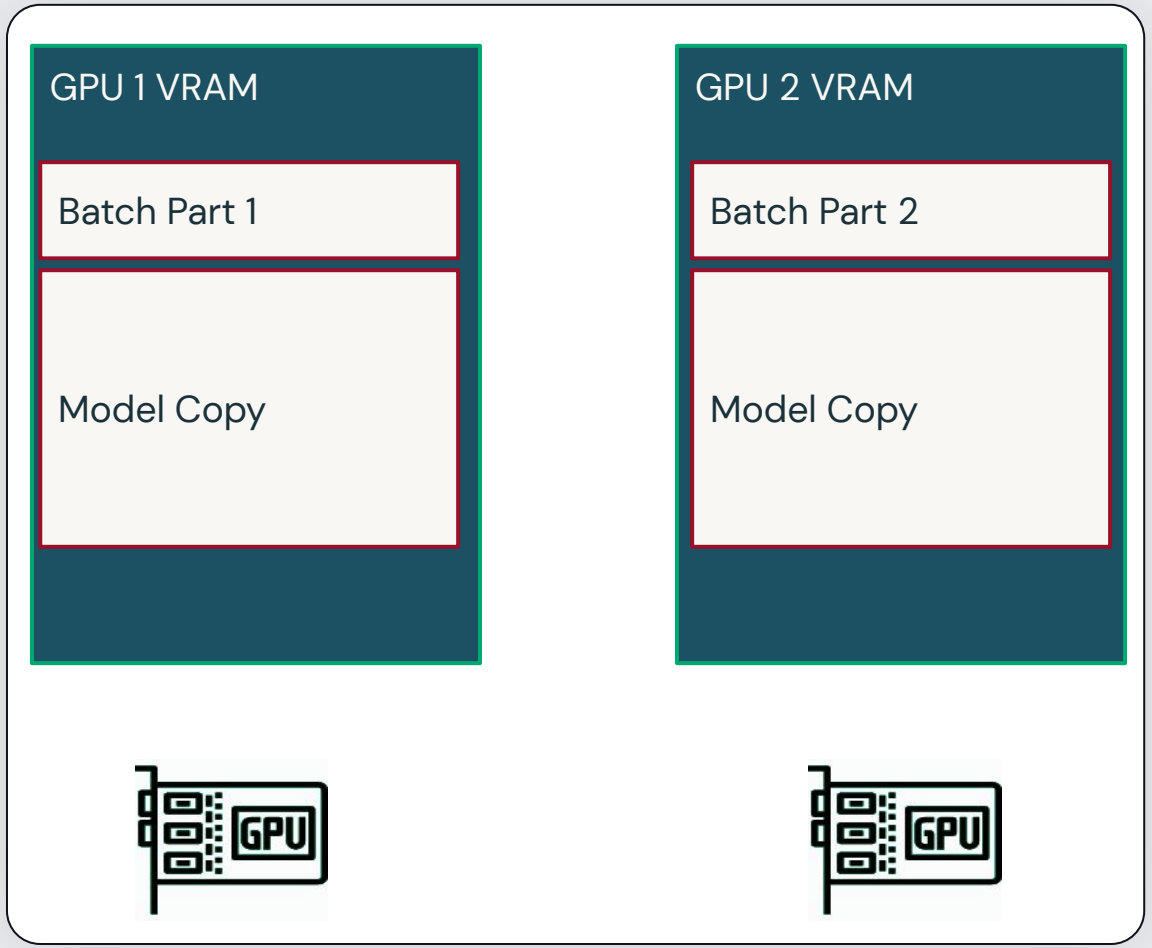


Data Parallel

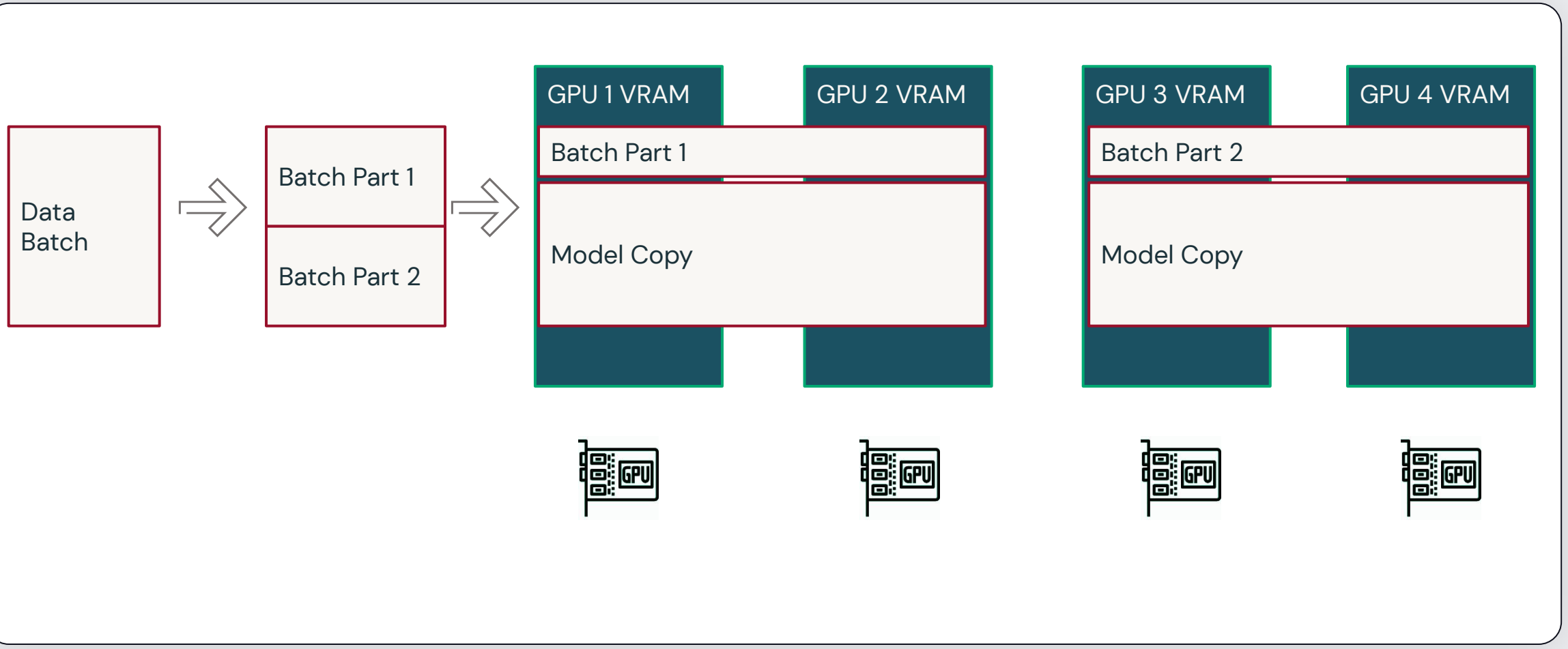


Most Common Pattern

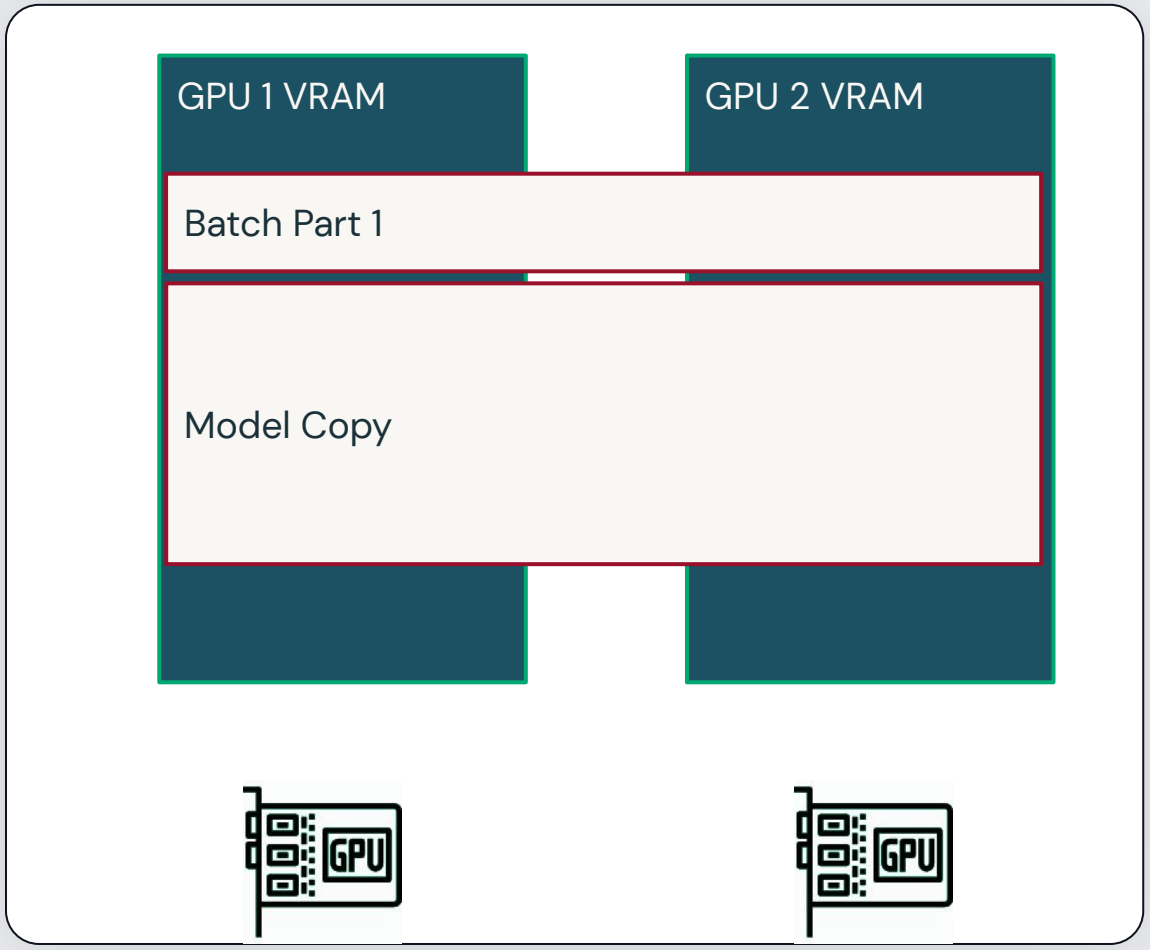
- Scales up the batch size so we can run more epochs
- Model weights sync after each batch can be bottleneck

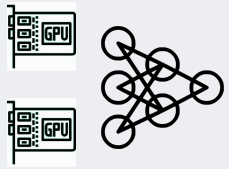


Model Parallel



Model Parallel





Model Split across GPUs

- Harder to tune
- Consideration needs to be given to cross GPU comms

Demo

Demo

How to code and scale Deep Learning

- Working with datasets
- Scaling up on single node
- Scaling up to multi node
- Understanding Performance

How to Scale - A quick cheat sheet

How to scale

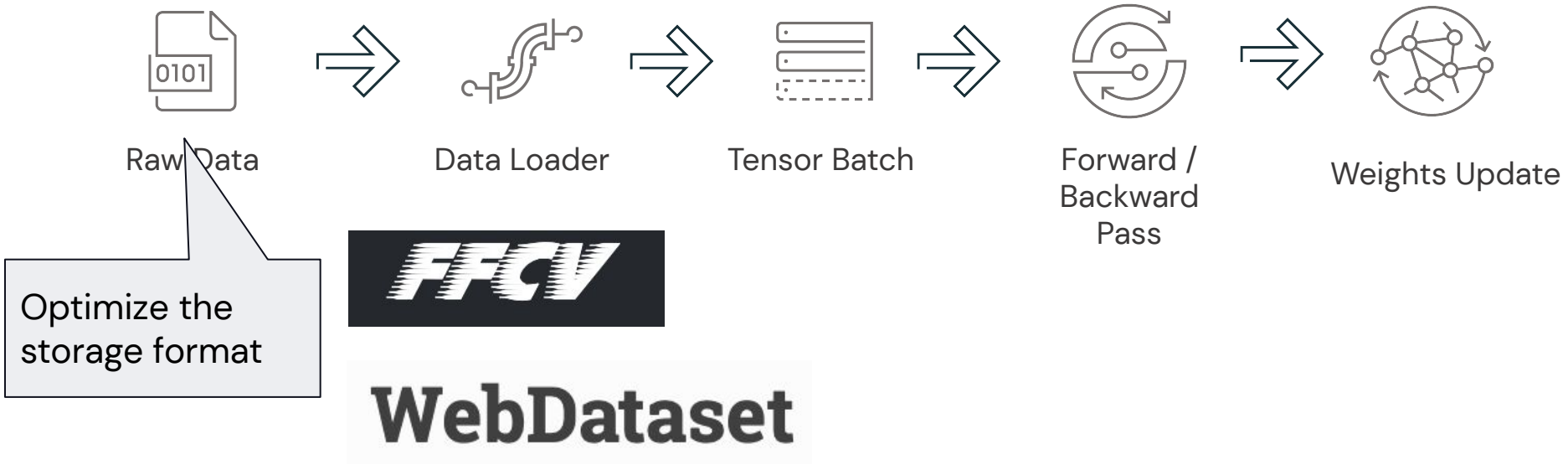
How to code and scale Deep Learning

- Focus on Single Node Single GPU to start
- Scale works by increasing batch size
- Make sure to optimize your RAM usage and check your dataloaders
  - pin_memory
  - Num_workers
  - fp16
- https://github.com/Data-drone/DAIS2022-Scaling-Deep-Learning-Talk

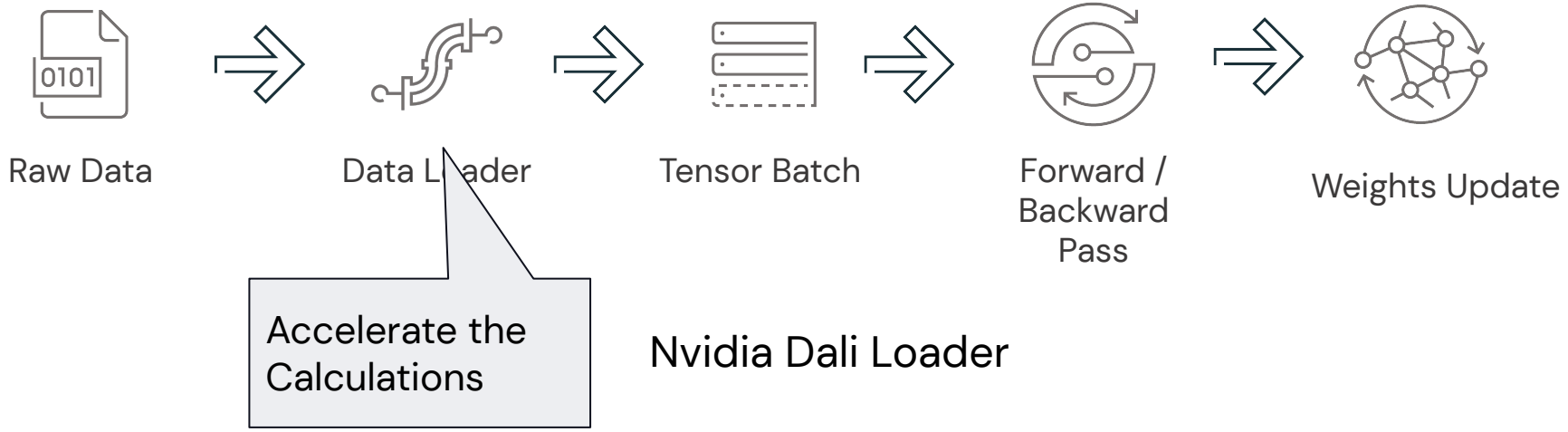
Where are things heading?

Raw Data → Data Loader → Tensor Batch → Forward / Backward Pass → Weights Update

What other options are there?



What other options are there?



What other options are there?

mosaicML

DeepSparse Engine

Enhance the model architectures

DATA+AI
SUMMIT 2022

Thank you

Brian Law

Specialist Solution Architect