ADC-B4

# How to scale and optimise training of Large Language Models (LLMs) on Amazon SageMaker

Daniel Zagyva
Data Scientist
AWS Professional Services

Laurens van der Maas
Machine Learning Engineer
AWS Professional Services

Somita Yogi
Delivery Practice Manager
AWS Professional Services

# Agenda

Training on SageMaker

Optimisation (profiling)

Distributed training, data parallel

Distributed training, model parallel

ProServe

# Amazon SageMaker is a managed service that accelerates every stage of the ML lifecycle

**Build**

**Train**

**Deploy**

**Monitor**

# Large-scale training on SageMaker

## OPTIMISED DISTRIBUTED TRAINING LIBRARIES & FRAMEWORKS

- TensorFlow
- PyTorch
- Hugging Face
- SageMaker Distributed Training Libraries
- Bring your own library (e.g. DeepSpeed, Megatron)

## AMAZON SAGEMAKER TRAINING

- Large Scale Cluster Orchestration
- NCCL Health Checks
- SageMaker Jumpstart for foundation models
- SageMaker Compiler
- Warm pools
- SSH to container
- Data loading
- Debugger
- Profiling
- Experiment tracking
- Hyperparameter optimisation
- Pay for what you use

## ML COMPUTE INSTANCES & ACCELERATORS

- NVIDIA GPUS H100, A100, V100, K80, T4, A10
- AWS Nitro
- 400/800 Gbps EFA Networking
- CPU instances
- AWS Trainium

# Optimisation

# Profiling your training jobs

Inefficient utilisation leads to

- Longer training times
- Incomplete training runs
- Increased overall costs and project timelines

Efficient resource usage is key

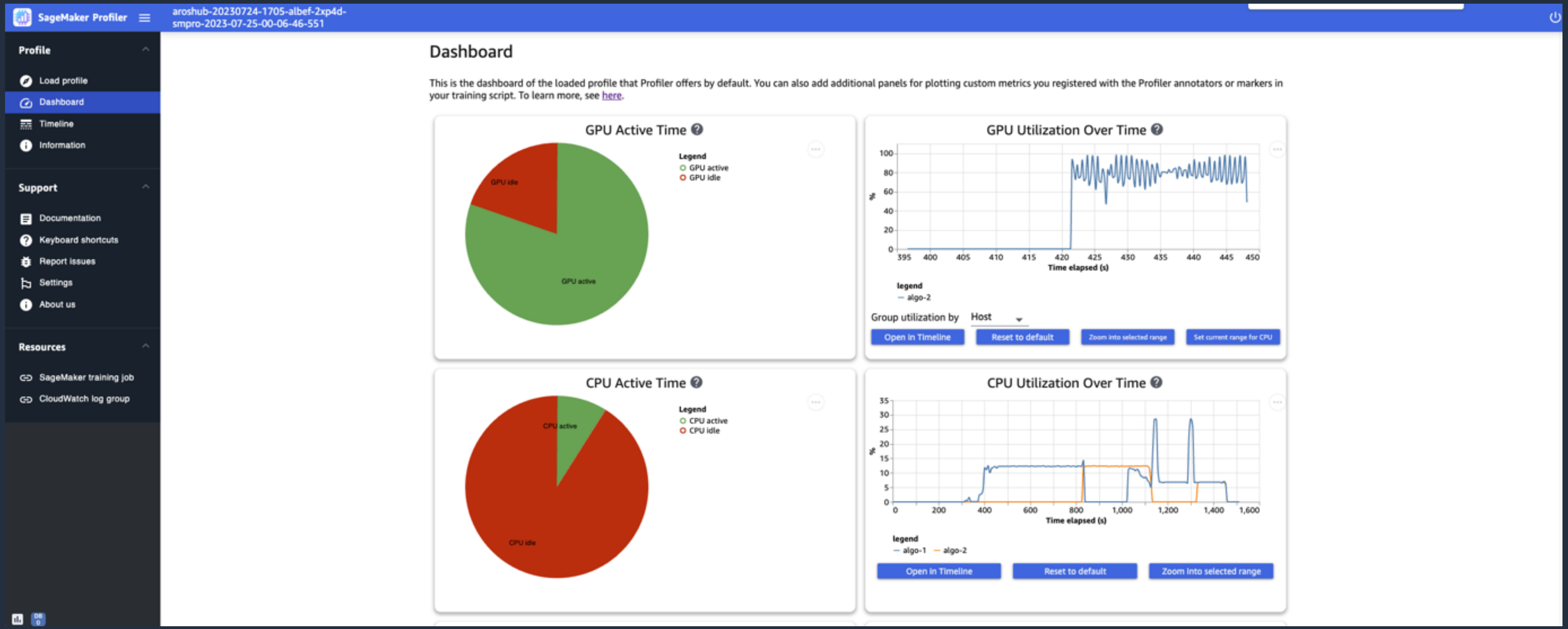
With profiling, you can solve problems such as

- I/O bottlenecks
- Kernel launch latencies
- Memory limits
- Low resource utilisation

# SageMaker Profiler – launched last month

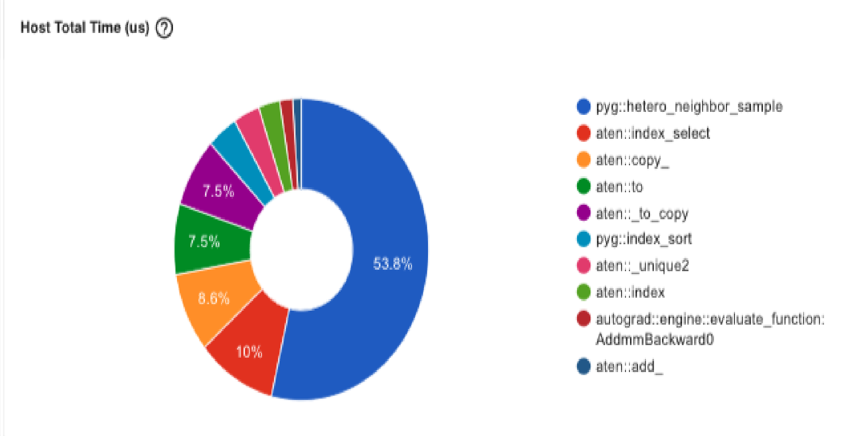
## RESOLVE YOUR TRAINING INEFFICIENCIES

```python
import smppy

sm_prof = smppy.SMProfiler.instance()
config = smppy.Config()
config.profiler = {
    "EnableCuda": "1",
}
sm_prof.configure(config)
sm_prof.start_profiling()

for epoch in range(args.epochs):
    if world_size > 1:
        sampler.set_epoch(epoch)
    tstart = time.perf_counter()
    for i, data in enumerate(trainloader, 0):
        with smppy.annotate("step_"+str(i)):
            inputs, labels = data
            inputs = inputs.to("cuda", non_blocking=True)
            labels = labels.to("cuda", non_blocking=True)
```



# SageMaker Profiler

## RESOLVE YOUR TRAINING INEFFICIENCIES

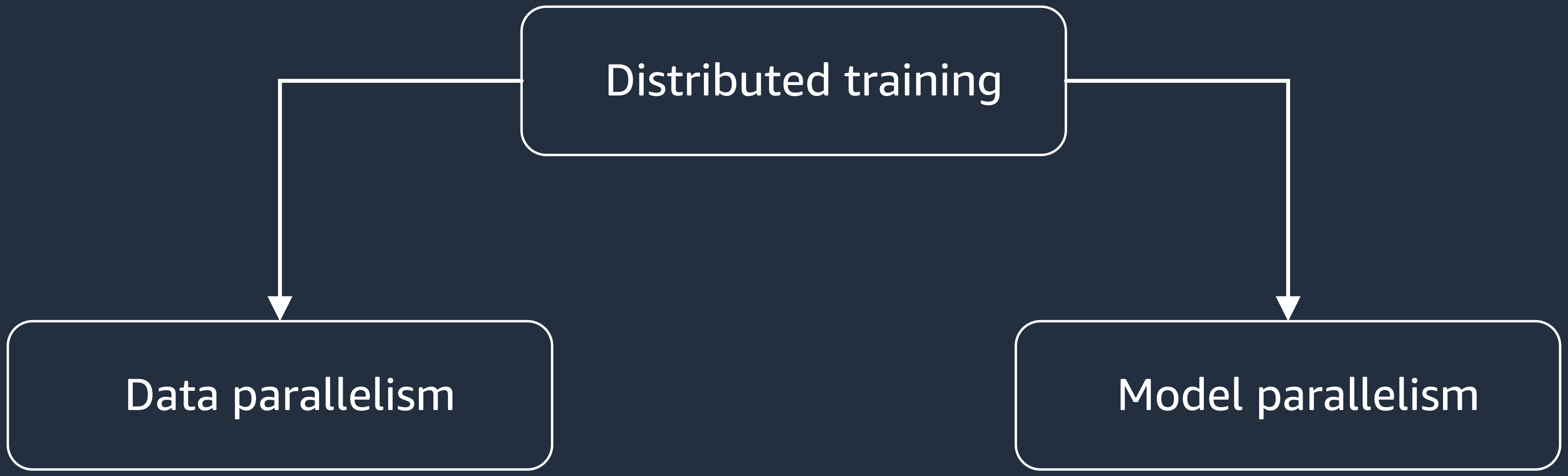




# Distributed training

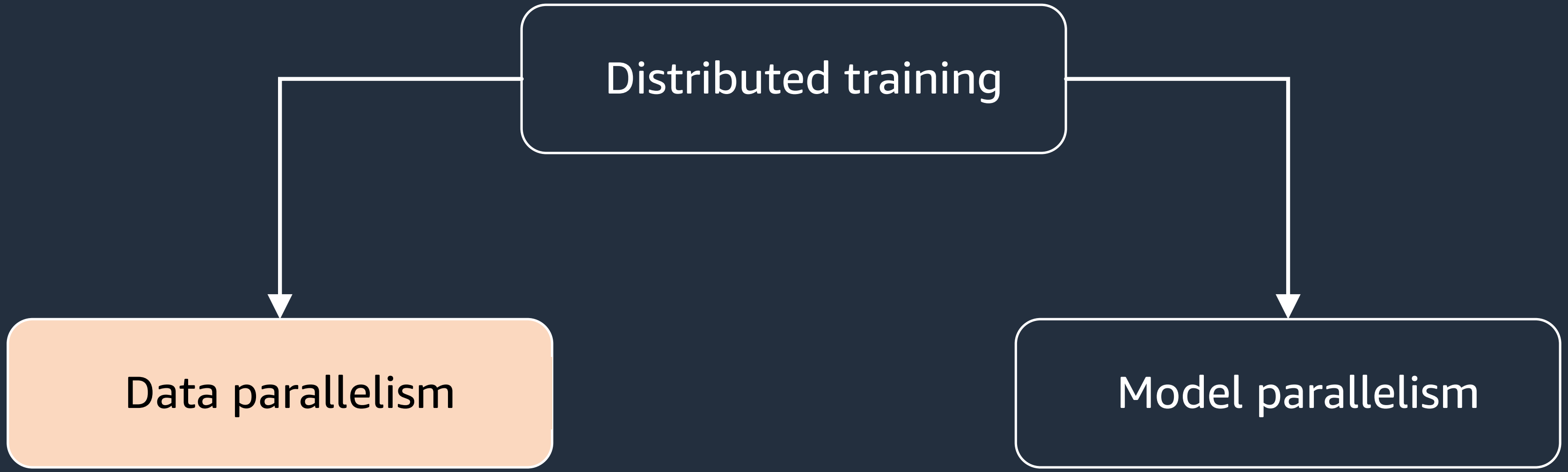
Lyft, one of the largest transportation networks in the United States and Canada, launched its Level 5 autonomous vehicle division in 2017 to develop a self-driving system to help millions of riders. Lyft Level 5 aggregates over 10 terabytes of data each day to train ML models for its fleet of autonomous vehicles. Managing ML workloads on its own was becoming time-consuming and expensive.

> "Using Amazon SageMaker distributed training, we reduced our model training time from days to a couple of hours. By running our ML workloads on AWS, we streamlined our development cycles and reduced costs, ultimately accelerating our mission to deliver self-driving capabilities to our customers."
>
> **Alex Bain, Lead for ML Systems, Lyft Level 5**

# Distributed training

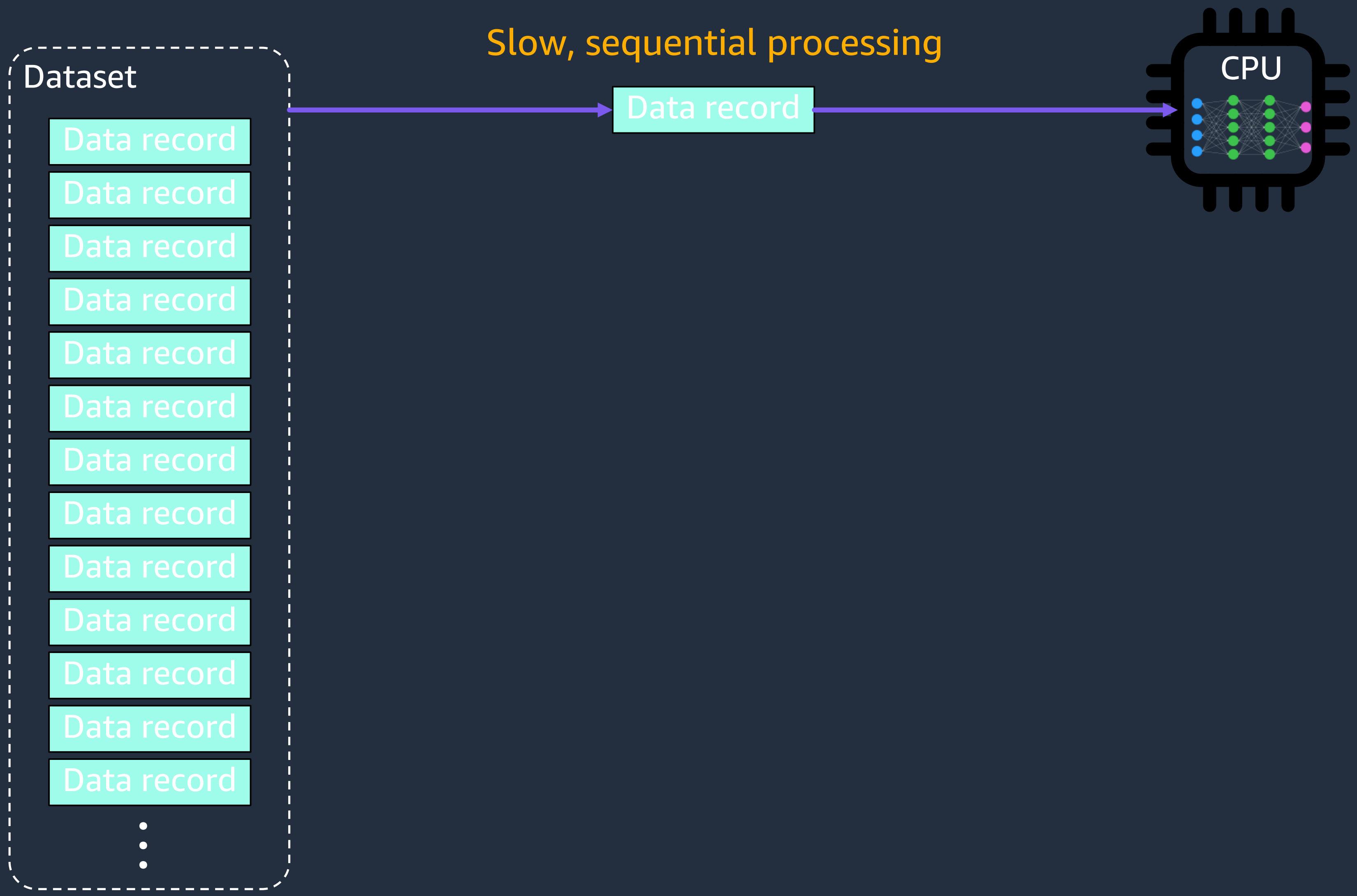


# Distributed training

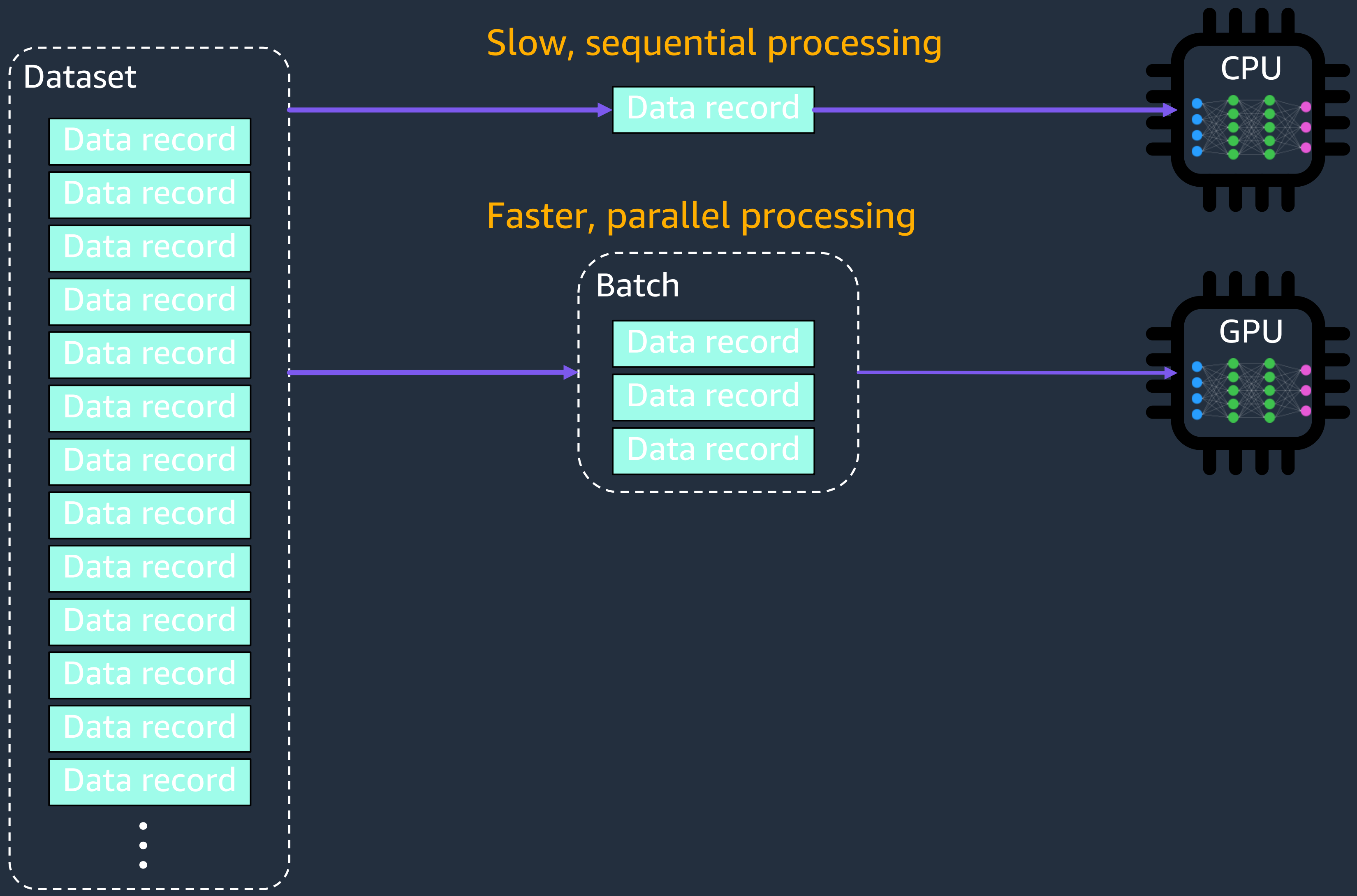


# The use case for data parallelism

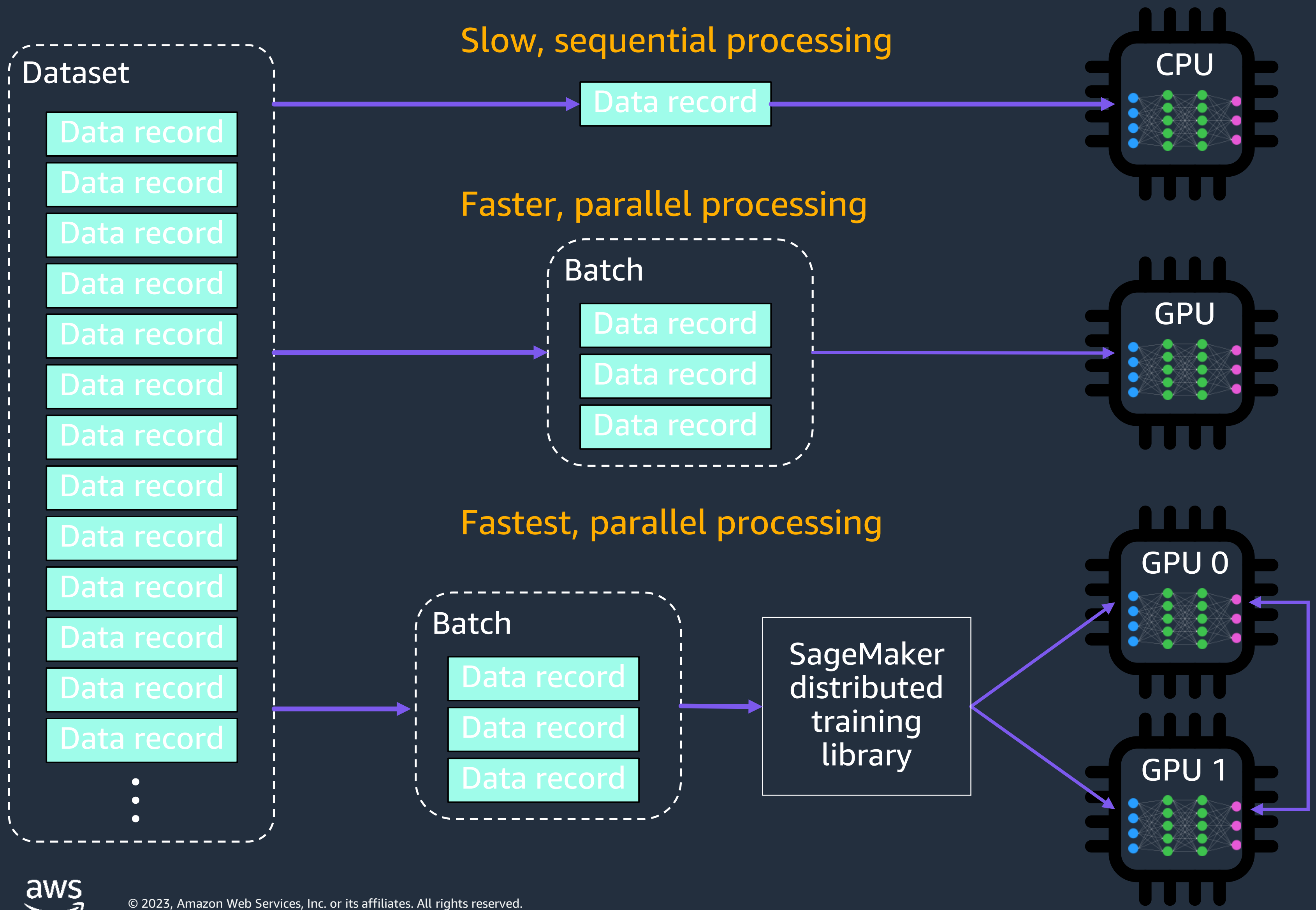
# The use case for data parallelism



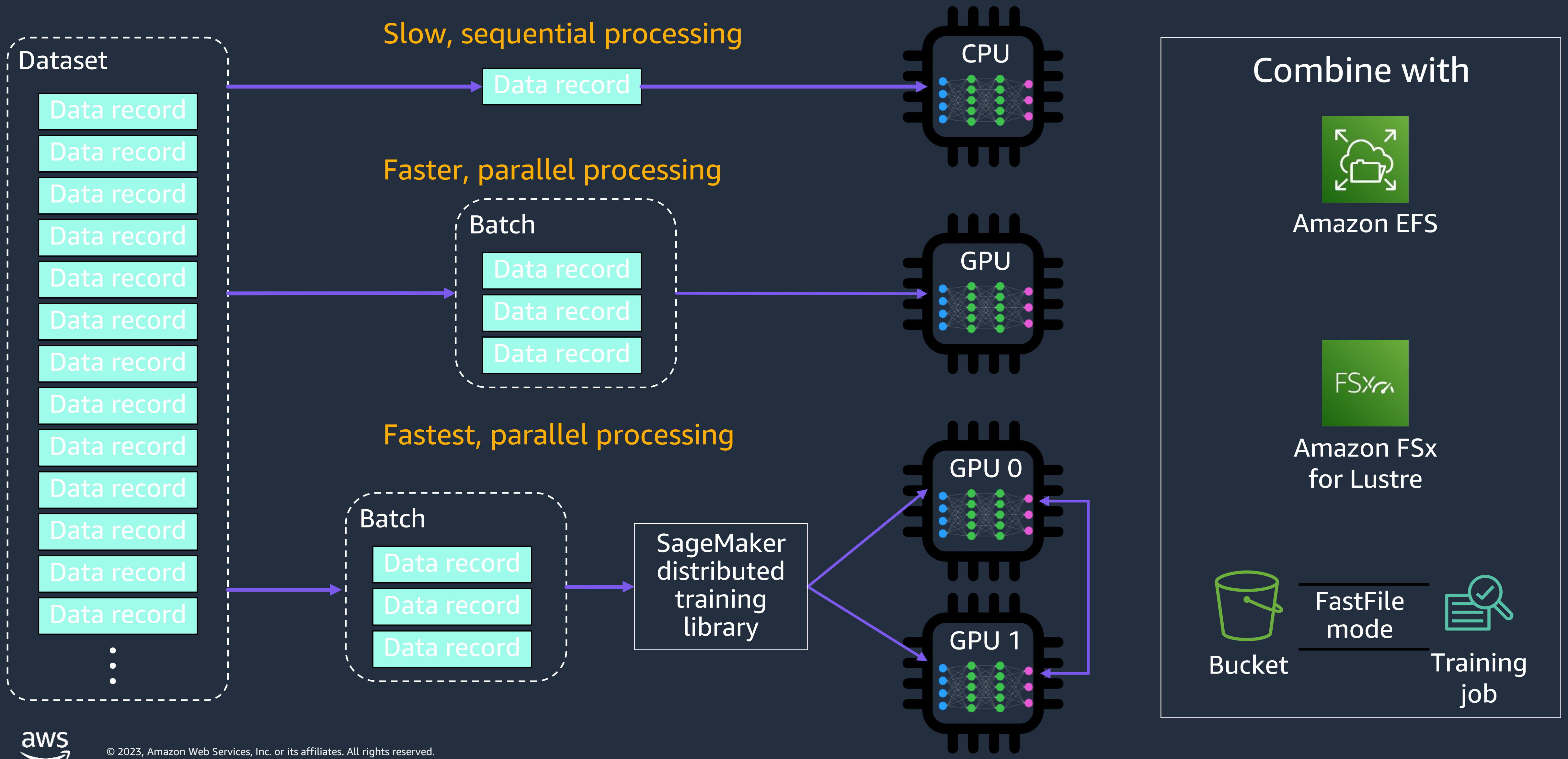
# The use case for data parallelism



# The use case for data parallelism



# The use case for data parallelism



# Frameworks on Amazon SageMaker (PyTorch)

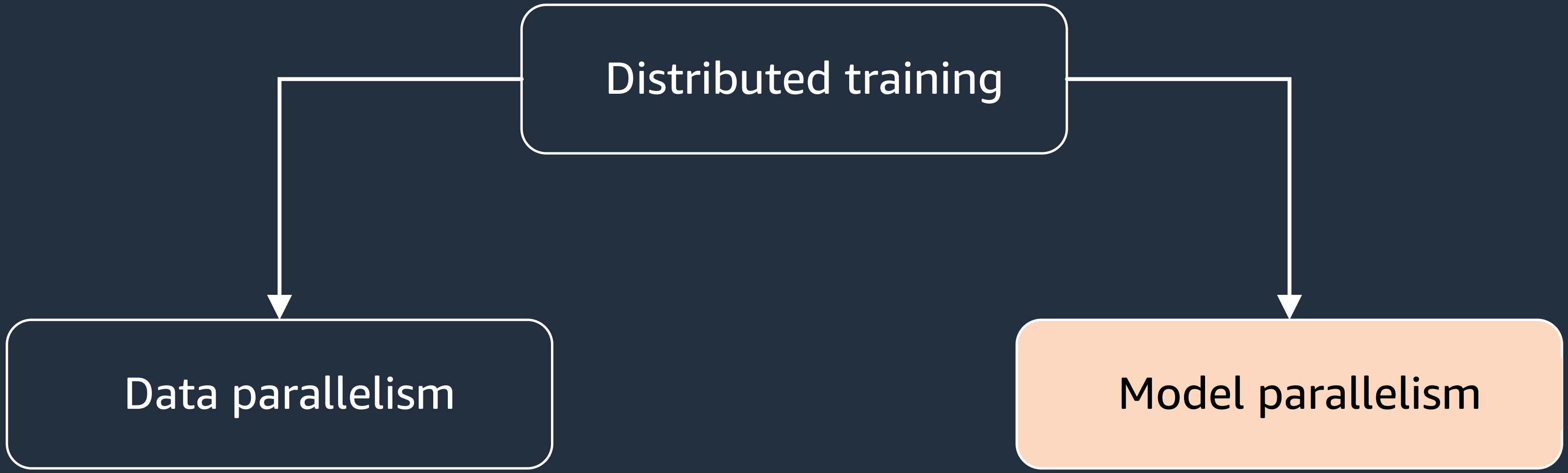
- Horovod
- PyTorch Distributed Data Parallel
- SageMaker Distributed Data Parallel

# SageMaker Distributed Data Parallel library

- Optimised backend for distributed training of deep learning models in TensorFlow, PyTorch
- Accelerates training for network-bound workloads
- Built and optimised for AWS network topology and hardware
- 20–40% faster and cheaper than NCCL and MPI-based solutions – **best performance on AWS for large clusters**

| Number of Instances | Training Time (minutes) | Improvement |
|---|---|---|
| 1 | 99 | Baseline |
| 2 | 55 | 1.8x |
| 4 | 27 | 3.7x |
| 8 | 13.5 | 7.3x |

# Distributed Training



# Model parallelism

## “Large Models” – splits the model across multiple GPUs

AWS Machine Learning Blog

# Stability AI builds foundation models on Amazon SageMaker

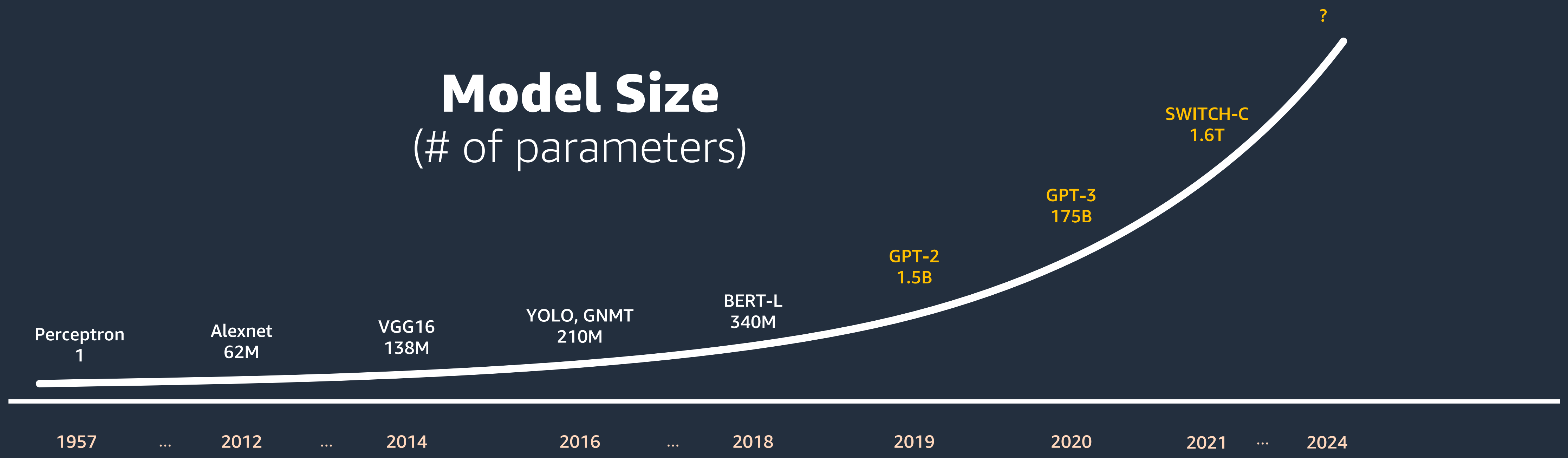
by Aditya Bindal | on 30 NOV 2022 | in Amazon SageMaker, Artificial Intelligence, Intermediate (200) | Permalink | Comments | Share

Prompt: *Four people riding a bicycle in the Swiss Alps, renaissance painting, epic breathtaking nature scene, diffused light*

Stability AI was able to reduce training time and cost by 58% using SageMaker and its model parallel library.

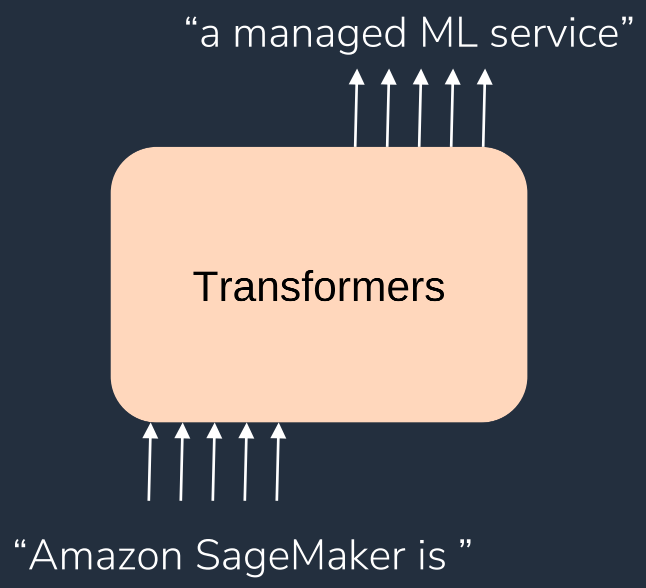
# AI models are getting bigger

… A LOT BIGGER



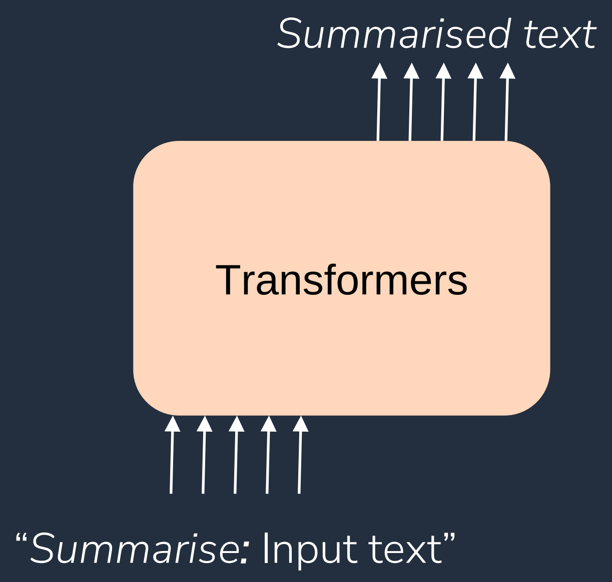
# LLM training techniques

## Pre-training



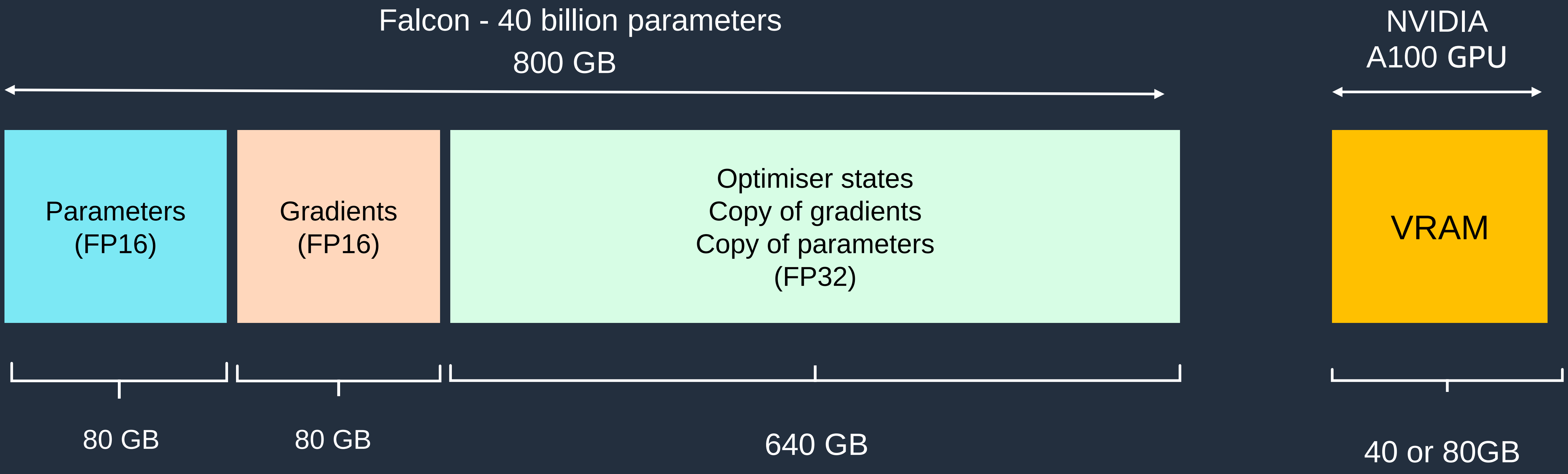

- Customisation of architecture, vocabulary size, context length
- Large-scale unlabelled data
- Days/weeks training time

## Full fine-tuning




- Update of all weights
- Task specific dataset
- Minutes/hours of training time

# Full fine tuning LLM-s requires multiple GPU-s (above ~1-2 billion parameters)

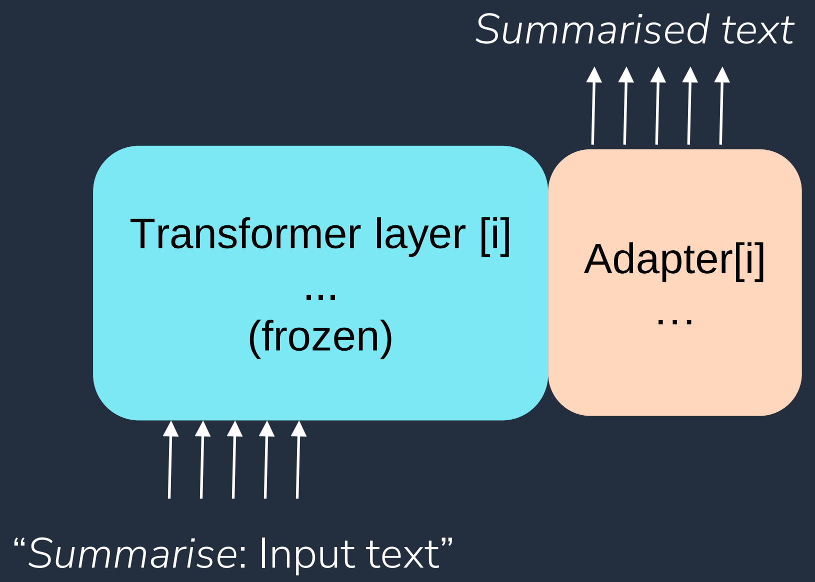





https://docs.aws.amazon.com/sagemaker/latest/dg/model-parallel-intro.html

# Efficient fine-tuning of LLM-s can still require multiple GPUs (above ~30-40 billion parameters)

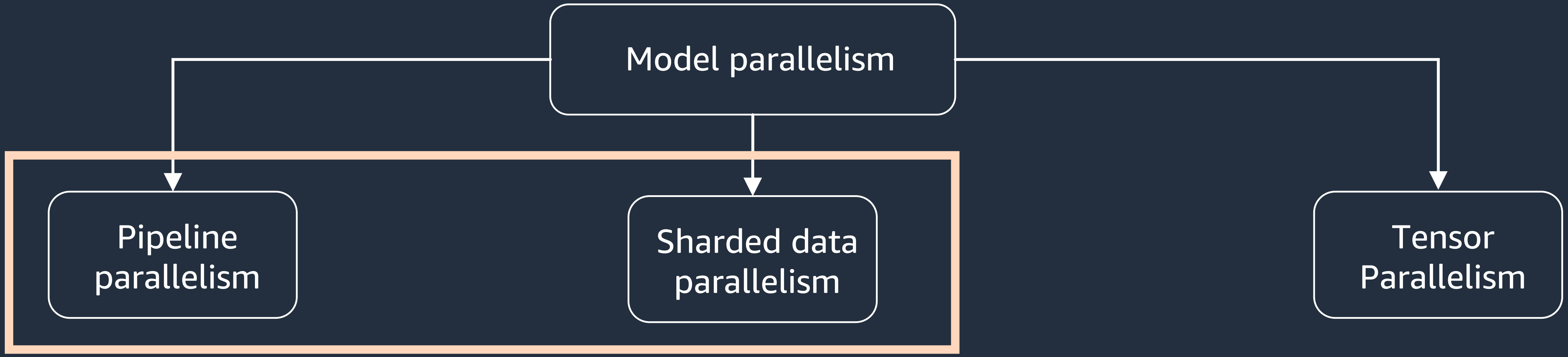
## Efficient fine-tuning – LoRA/QLoRA



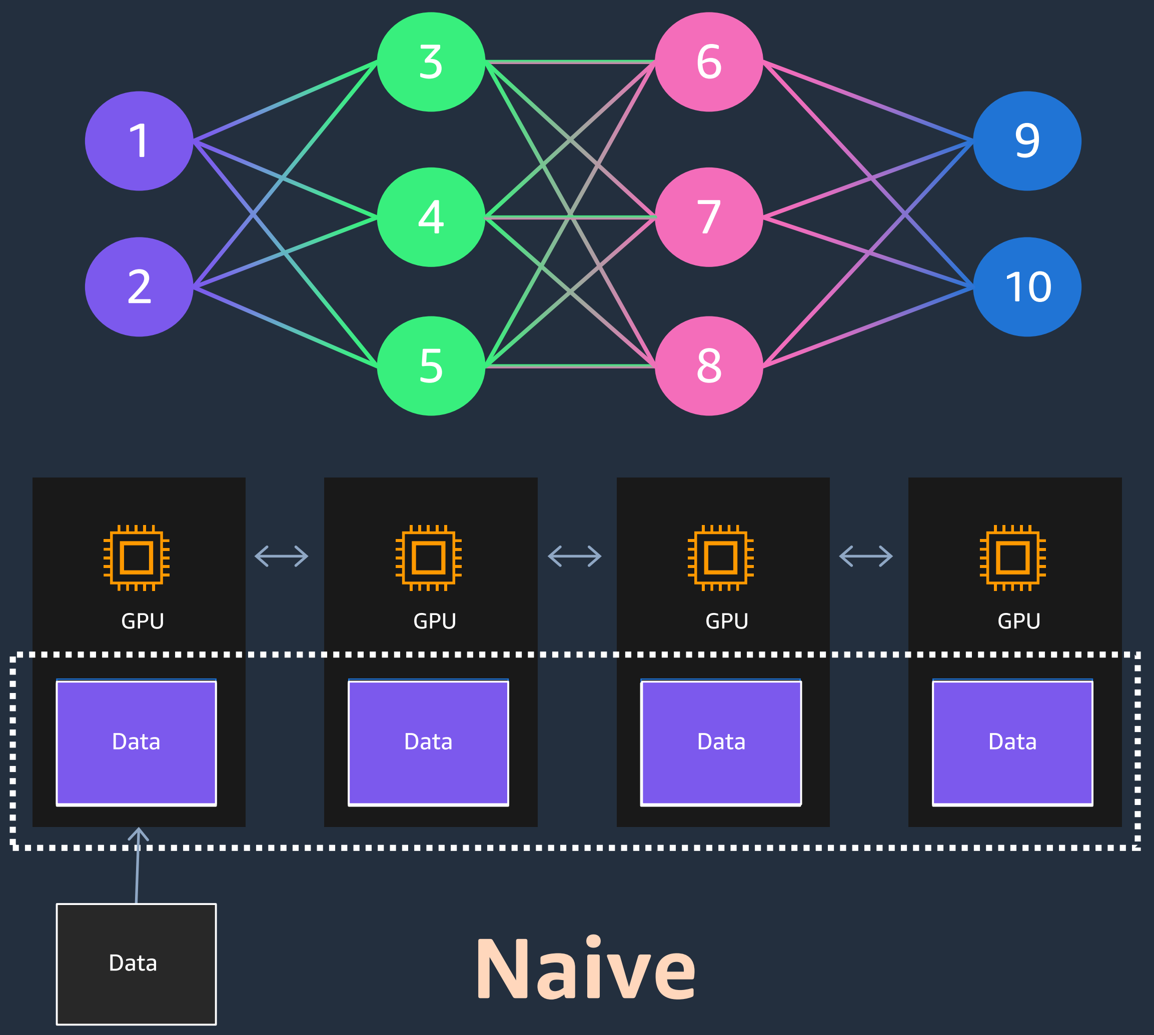

- Hypothesis: updates can be learned with 2 small matrices
- Reduces # trainable parameters by >1,000x; with comparable performance
- QLoRA: Quantise pre-trained model to 4-bit (FP)
- Often single GPU is enough
- Multi-GPU still required for larger models e.g. Falcon 40B >40GB memory

https://arxiv.org/abs/2106.09685

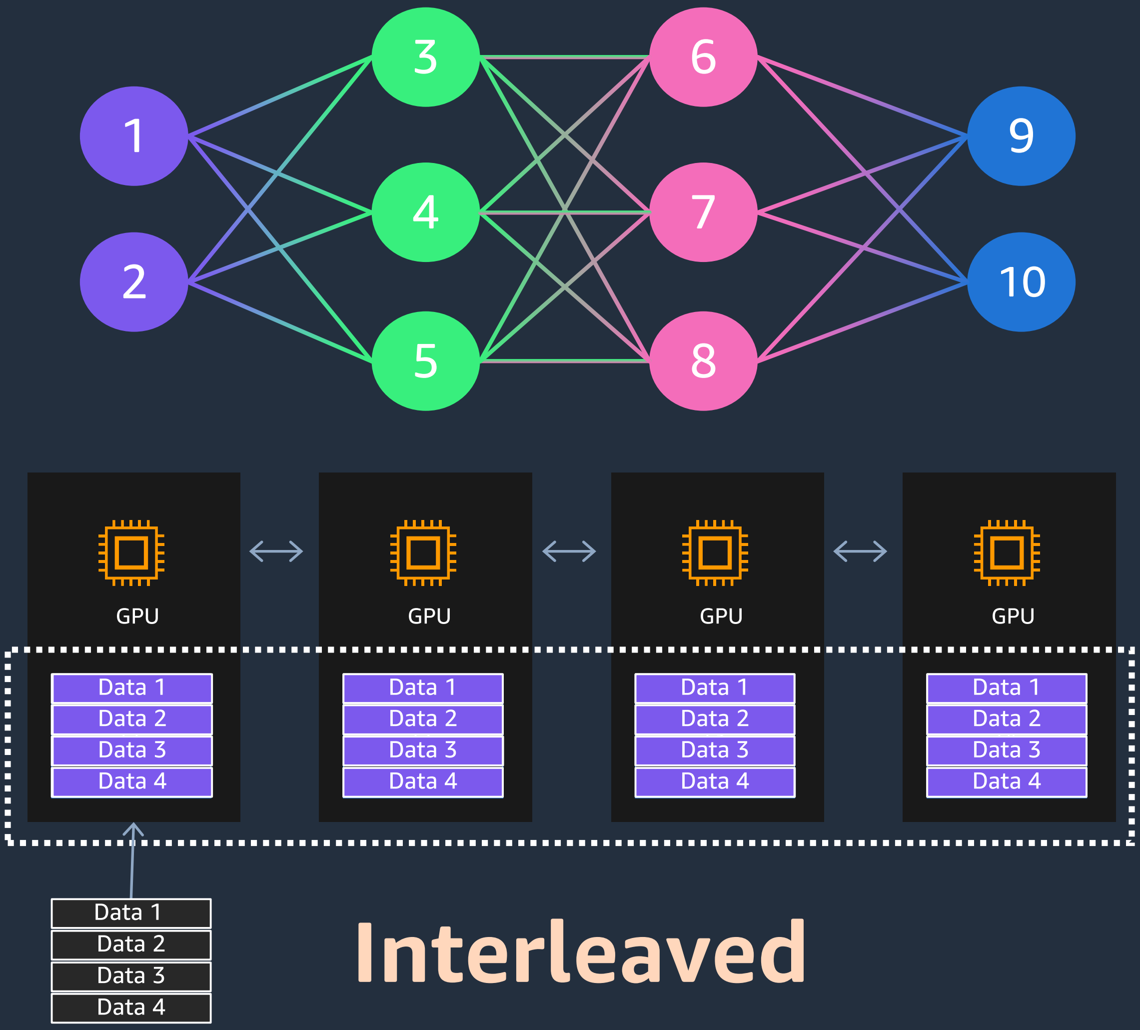
# Model parallelism options



# Pipeline parallelism - partitions model layers across GPUs

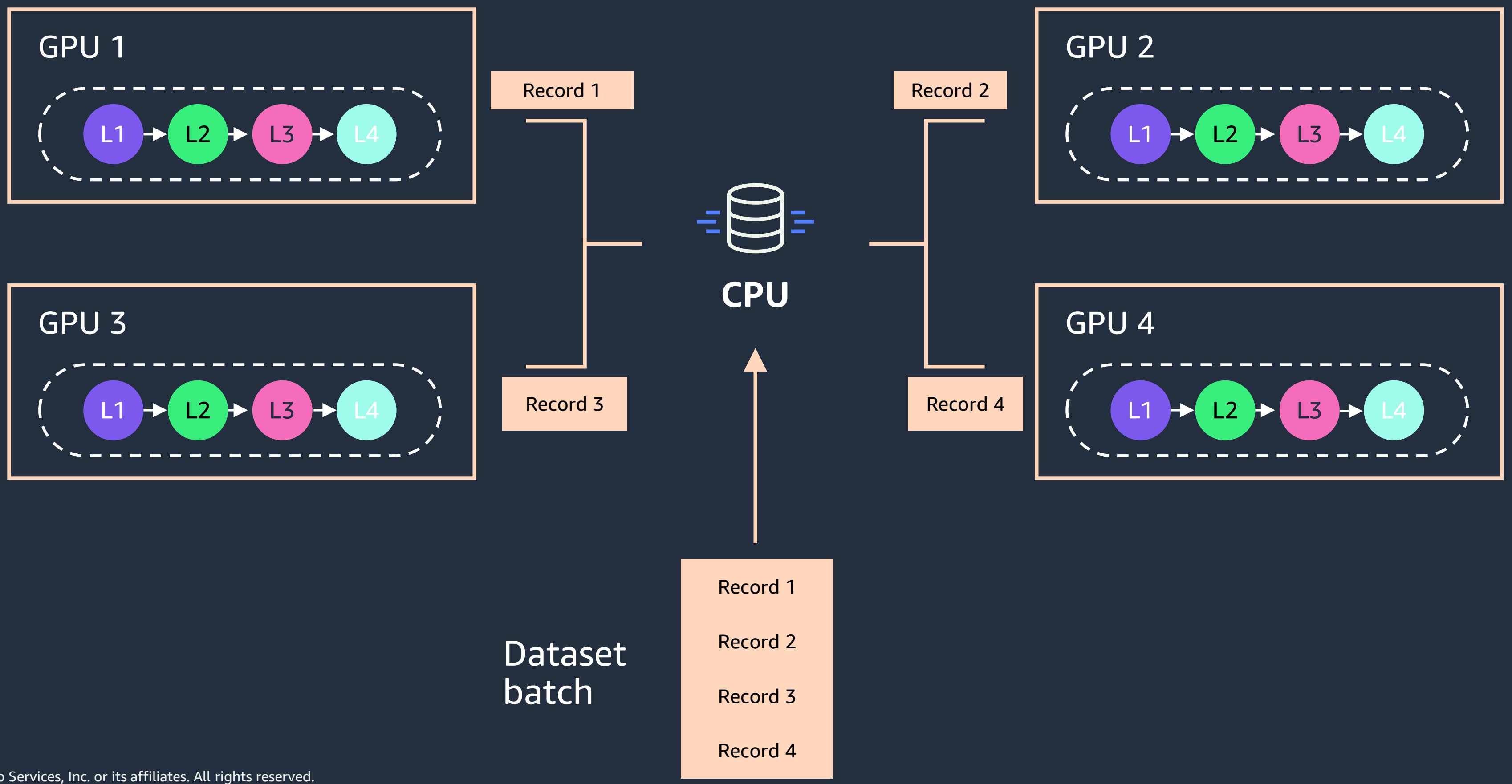


# Pipeline parallelism - partitions model layers across GPUs



# Sharded data parallelism

Splits the state of a model across GPUs and shares them during forward and backward pass



# Frameworks on Amazon SageMaker (PyTorch)

## Pipeline parallelism

- SageMaker Model Parallel (SMP) – Pipeline Parallel
- PyTorch pipeline parallel
- DeepSpeed pipeline parallel

AWS Machine Learning Blog

**Train gigantic models with near-linear scaling using sharded data parallelism on Amazon SageMaker**

by Emily Webber, Can Karakus, Erin Ho, Rahul Huilgol, and Suhit Kodgule | on 31 OCT 2022 | in Amazon SageMaker, Artificial Intelligence, Expert (400) | Permalink | Comments | Share

## Sharded data parallelism

- SageMaker Model Parallel (SMP) - Sharded Data Parallel
- PyTorch Fully Sharded Data Parallel (FSDP)
- DeepSpeed Zero Stage 3

**27.5%** speed up
(October 2022)

# Simplify distributed training with Hugging Face

## Hugging Face

Hugging Face is the most popular Open Source company providing state of the art NLP technology

## AWS

SageMaker offers high performance resources to train and use NLP Models

# Amazon SageMaker – Hugging Face example #1
## Minimal code changes for distributed training

- Pipeline parallelism - PyTorch, naive
- Falcon 40B
- Efficient fine-tune (QLoRA)
- g5.12xlarge (4 x 24GB GPU-s)

```
from transformers import AutoModelForCausalLM, BitsAndBytesConfig

model = AutoModelForCausalLM.from_pretrained(
    "tiiuae/falcon-40b",
    trust_remote_code=True,
    quantization_config=BitsAndBytesConfig(
        load_in_4bit=True,
        bnb_4bit_use_double_quant=True,
        bnb_4bit_quant_type="nf4",
        bnb_4bit_compute_dtype=torch.bfloat16
    ),
    device_map="auto"
)
```

AWS Machine Learning Blog

**Interactively fine-tune Falcon-40B and other LLMs on Amazon SageMaker Studio notebooks using QLoRA**

by Sean Morgan, Philipp Schmid, and Lauren Mullennex | on 29 JUN 2023 | in Amazon Machine Learning, Amazon SageMaker, Artificial Intelligence, Generative AI, Technical How-To | Permalink | Comments | Share

# Amazon SageMaker – Hugging Face example #2

## Minimal code changes for distributed training

- Sharded Data Parallelism - PyTorch FSDP
- GPT-NeoXT-Chat-Base-20B
- Full fine-tune
- 2 x ml.p4d.24xlarge
(2 x 8 x 40 GB GPU-s)

```
from sagemaker.huggingface import HuggingFace

huggingface_estimator = HuggingFace(
    entry_point='run_clm.py',
    source_dir='./scripts',
    instance_type="ml.p4d.24xlarge",
    instance_count=2,
    volume_size=200,
    role=role,
    job_name=job_name,
    transformers_version='4.26.0',
    pytorch_version='1.13.1',
    py_version="py39",
    hyperparameters=hyperparameters,
    distribution={
        "torch_distributed":
            {"enabled": True}
    }
)
```

```
from transformers import TrainingArguments, \
    Seq2SeqTrainer

training_args = TrainingArguments(
    output_dir=output_dir,
    per_device_train_batch_size=8,
    bf16=False,
    num_train_epochs=1,
    logging_strategy="steps",
    logging_steps=10,
    fsdp="full_shard auto_wrap",
    fsdp_transformer_layer_cls_to_wrap="GPTNeoXLayer",
)
trainer = Seq2SeqTrainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
    data_collator=data_collator,
)

trainer.train()
```

# AWS Professional Services

# Why ProServe?

**Our Purpose**

Our existence is singular: accelerating customer outcomes through the innovative adoption of the AWS platform. We help the customer to be self-sufficient.

**Our Provenance**

As an Amazonian business, our customer centricity fosters an unrelenting pursuit of customer outcomes.

**Our Position**

Our proximity with AWS product teams, customers, and Partners - not only harnesses unparalleled AWS technical skills - it allows us to convey customer learnings to influence AWS product roadmaps.

**Our Pace**

Our approach is infectious. We foster a high-touch, proactive, ‘hands-on’, agile and iterative work ethic, which is essential, to avoid inertia.

Working alongside Partners in Big Data, Analytics, AI/ML, GenAI etc.

Track record of bringing/optimising workloads in various environments

Facilitate use case Discovery / PoC and technical validation

Deploying highly scalable analytics for deeper insights and wide platform adoption

Portfolio of ETIP Packaged offerings

aws professional services

Ensuring high quality of security, compliance and resiliency

## Emerging Technologies & Intelligent Platforms (ETIP)

Align → Launch → Scale → Optimise

# Generative AI is transforming all industries

Financial Services

Healthcare and Life Sciences

Automotive

Manufacturing

Media & Entertainment

Telecom

Energy

# Generative AI can be used for a wide range of use cases

Chatbots & Virtual assistants
Agent Assist
Contact Center Analytics
Personalisation

**Enhance customer experience**

Conversational search
Content Localisation
Text, image, video generation
Text summarisation
Code generation

**Boost employee productivity**

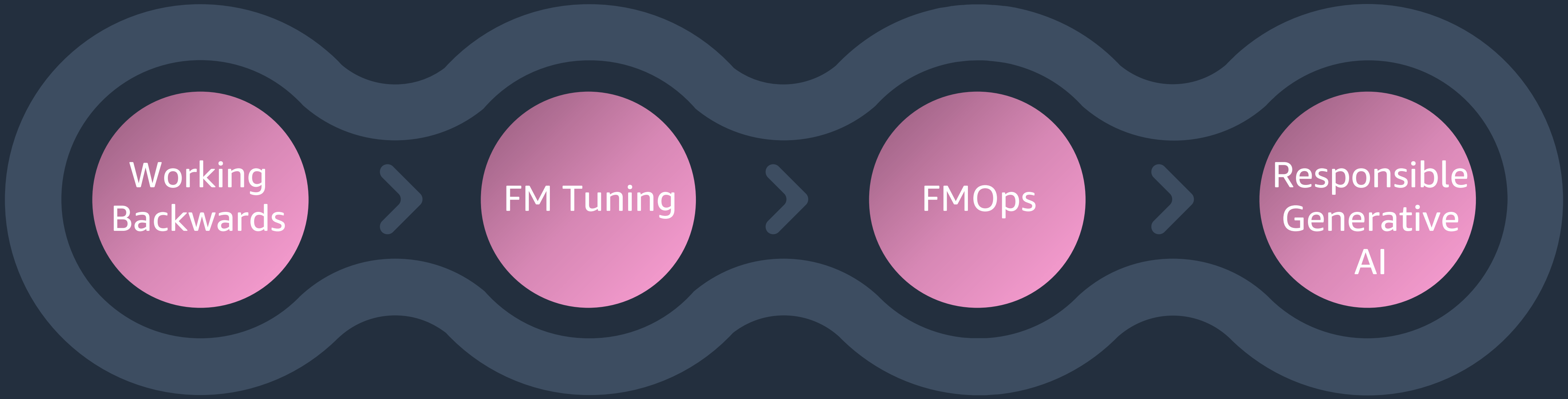
Document processing
Content moderation
Synthetic data creation
Maintenance assistance
Anomaly detection

**Improve business operations**

Image generation for web pages
Video enhancement
Music creation
Image enhancement
Creating animations

**Creativity**

# As you build a Generative AI roadmap




**Business value**

Identify use case opportunities to leverage generative AI for business value

**Explore models**

Custom and domain-specific FM tuning; white-glove implementation services to train and build a FM targeted to your use cases

**Path to production**

- Ongoing FM fine-tuning and Model Compression
- Refining FM knowledge and prompts
- Auto-labeling of training data

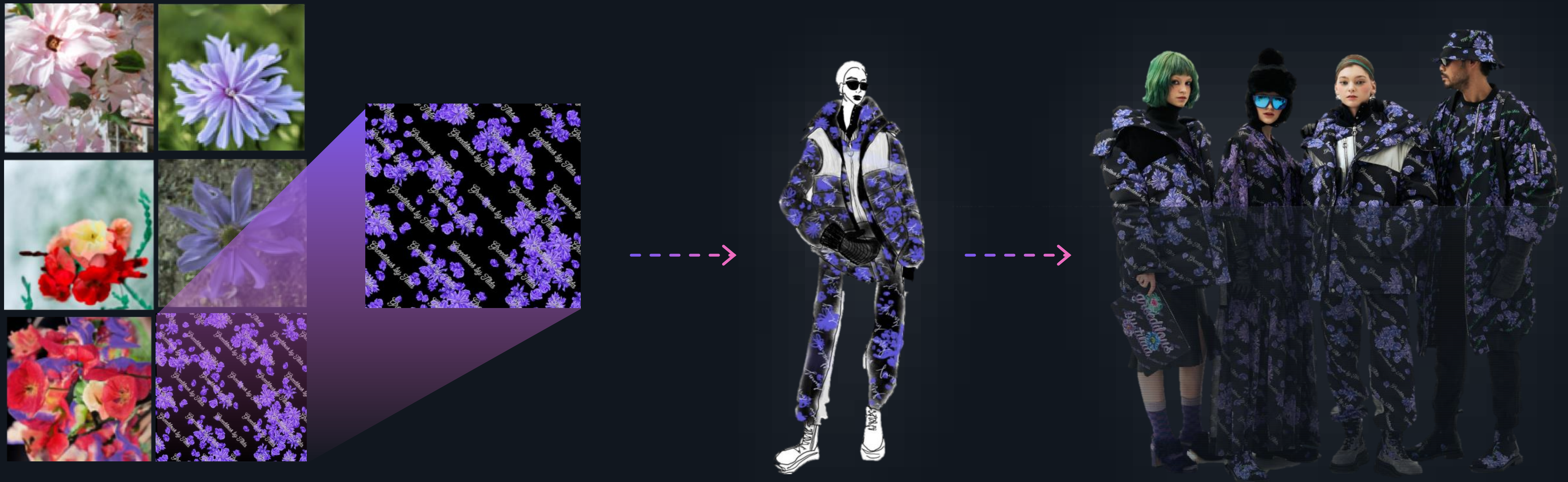
**Generative AI guidance**

Provide an approach for building and launching trustworthy GAI-based products and solutions from principle to practice

# LG AI Research developed FM using Amazon SageMaker

“We could optimise distributed training and were able to train the model faster by 59% (than without Amazon SageMaker) ”

Seung Hwan Kim
Vice President, Vision Lab Leader, LG AI Research



LG AI Research’s Tilda, the AI artist powered by EXAONE

# Thank you!

Daniel Zagyva
zagyvad@amazon.com

Laurens van der Maas
laurensv@amazon.com

Somita Yogi
somyogi@amazon.com

Please complete the session survey