AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

AIM 412 – R2

Deep Learning Applications with TensorFlow

Sam Palani
Principal Machine Learning Specialist
AWS

Tim O'Brien
Principal Solutions Architect
AWS

Topics

AWS ML Stack

Art of the Possible

TensorFlow on AWS

Training

Inference

Q&A

Reminder: this is a 400 level talk

The AWS ML Stack

Broadest and most complete set of machine learning capabilities

AI SERVICES

HEALTH AI


Amazon HealthLake


Amazon Transcribe Medical


Amazon Comprehend Medical

INDUSTRIAL AI


AWS Panorama + Appliance


Amazon Monitron


Amazon Lookout for Equipment


Amazon Lookout for Vision

ANOMALY DETECTION


Amazon Lookout for Metrics

CODE AND DEVOPS


Amazon DevOps Guru


Amazon CodeGuru

VISION


Amazon Rekognition

SPEECH


Amazon Polly


Amazon Transcribe

+Medical

TEXT


Amazon Comprehend

+Medical


Amazon Translate


Amazon Textract

SEARCH


Amazon Kendra

CHATBOTS


Amazon Lex

PERSONALIZATION


Amazon Personalize

FORECASTING


Amazon Forecast

FRAUD


Amazon Fraud Detector

CONTACT CENTERS


Contact Lens

Voice ID

For Amazon Connect

ML SERVICES


Amazon SageMaker

Label data

SAGEMAKER STUDIO IDE

Aggregate & prepare data | Store & share features | Auto ML | Spark/R | Detect bias | Visualize in notebooks | Pick algorithm | Train models | Tune parameters | Debug & profile | Deploy in production | Manage & monitor | CI/CD

Human review

SageMaker JumpStart

Model management for edge devices

FRAMEWORKS & INFRASTRUCTURE

TensorFlow | mxnet | PyTorch

GLUON | Keras | scikit learn | HOROVOD | DeepGraphLibrary

Deep Learning AMIs & Containers | GPUs & CPUs | Elastic Inference | Trainium | Inferentia | FPGA

Art of the Possible

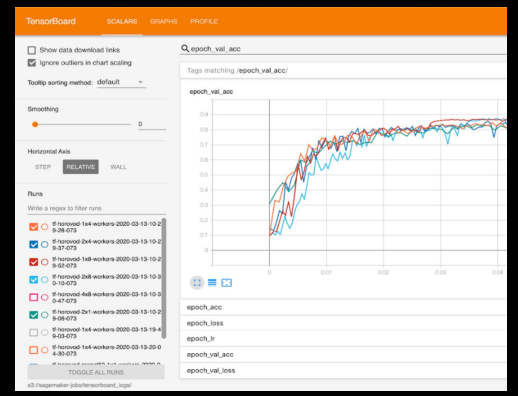
- Text Based Applications
- Image Recognition
- Forecasting and Time series
- Video Detection
- Voice and Sound Recognition

TensorFlow on AWS

https://aws.amazon.com/tensorflow/

Elastic Inference

- AWS optimizations for TensorFlow available on Amazon Elastic Compute Cloud (Amazon EC2) and SageMaker
  - AWS Deep Learning Containers for training and inference
  - AWS Deep Learning AMIs (DLAMI)
- SageMaker benefits for TensorFlow
  - Built-in support for TensorBoard, Debugger, local mode, hyperparameter tuning, managed spot training, Pipe mode, and Amazon Elastic Inference
  - Distributed training – parameter server and Horovod
  - Performance optimizations – GPUs, CPUs, and storage



TensorBoard

aws

SageMaker Training Options

Script Mode

- AWS deep learning containers for training

Custom Container

- Extend pre-built containers
- Bring your own containers

Distributed Training

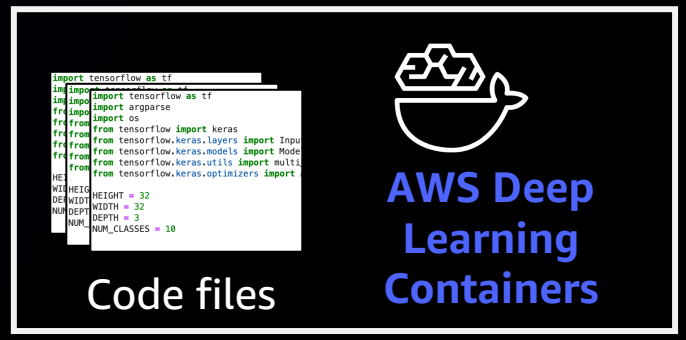
- Data Parallel
- Model Parallel

Training Flow

SageMaker SDK

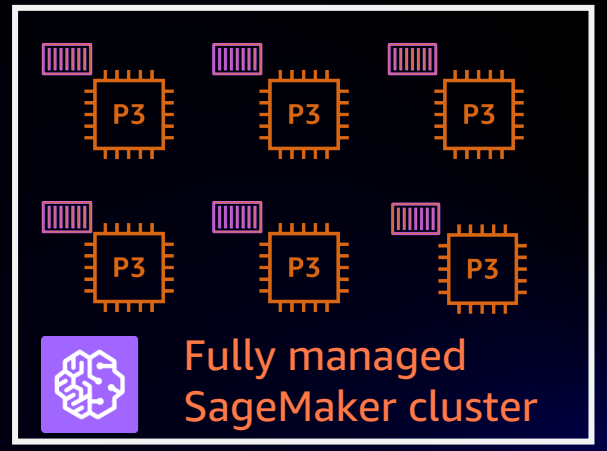
```
from sagemaker.tensorflow import TensorFlow

cifar10_estimator = TensorFlow(entry_point='source/cifar10.py',
                               role=role,
                               framework_version='1.14',
                               train_instance_count=1,
                               train_instance_type='ml.p3.xlarge')
```



Amazon Elastic Container Registry (Amazon ECR)

Container registry



Amazon S3

Distributed Training Solutions

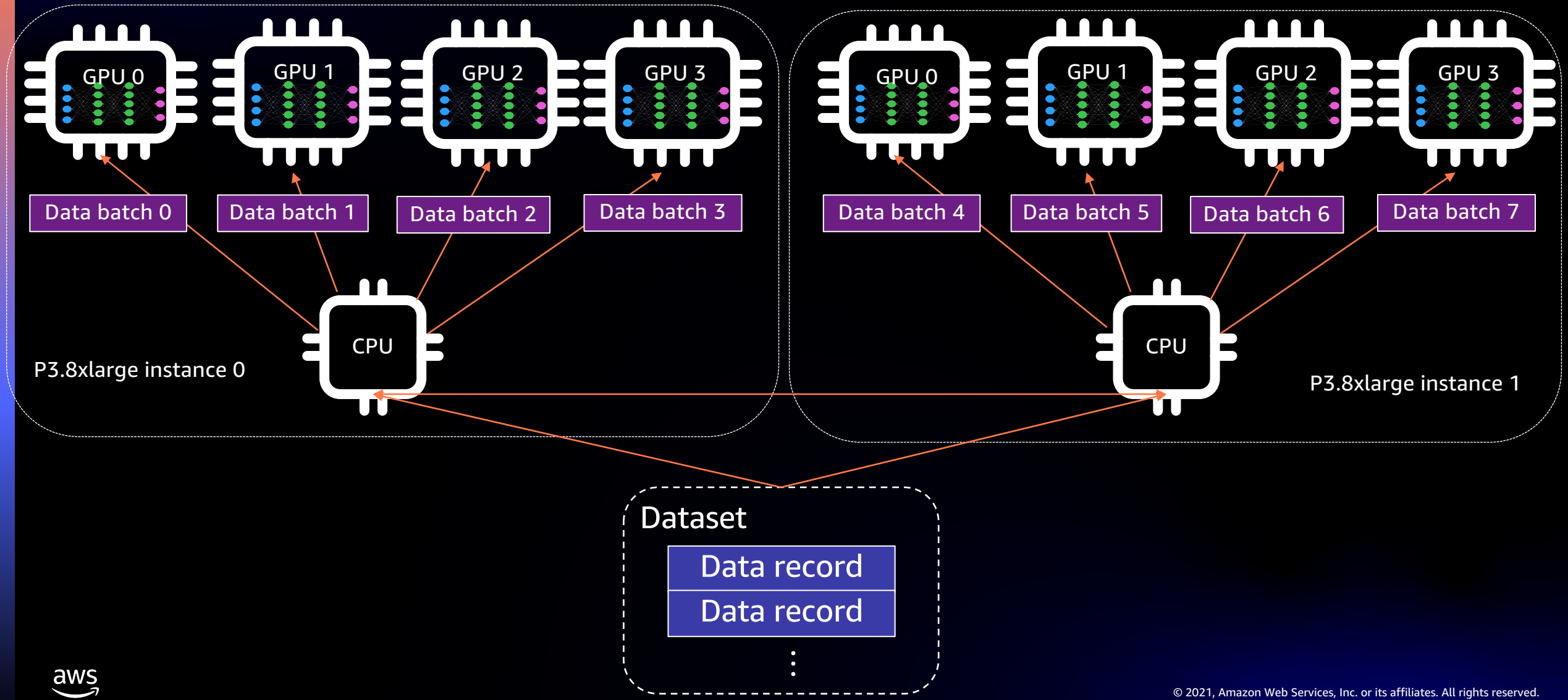
Data Parallelism

- Training dataset is split across multiple nodes.
- Each node has a replica of the model.
- Each node performs a forward & backward pass and synchronizes gradients.
- Still requires to fit the model and at least one record in memory (GPU/CPU).
- Additional complexity of computing gradients.

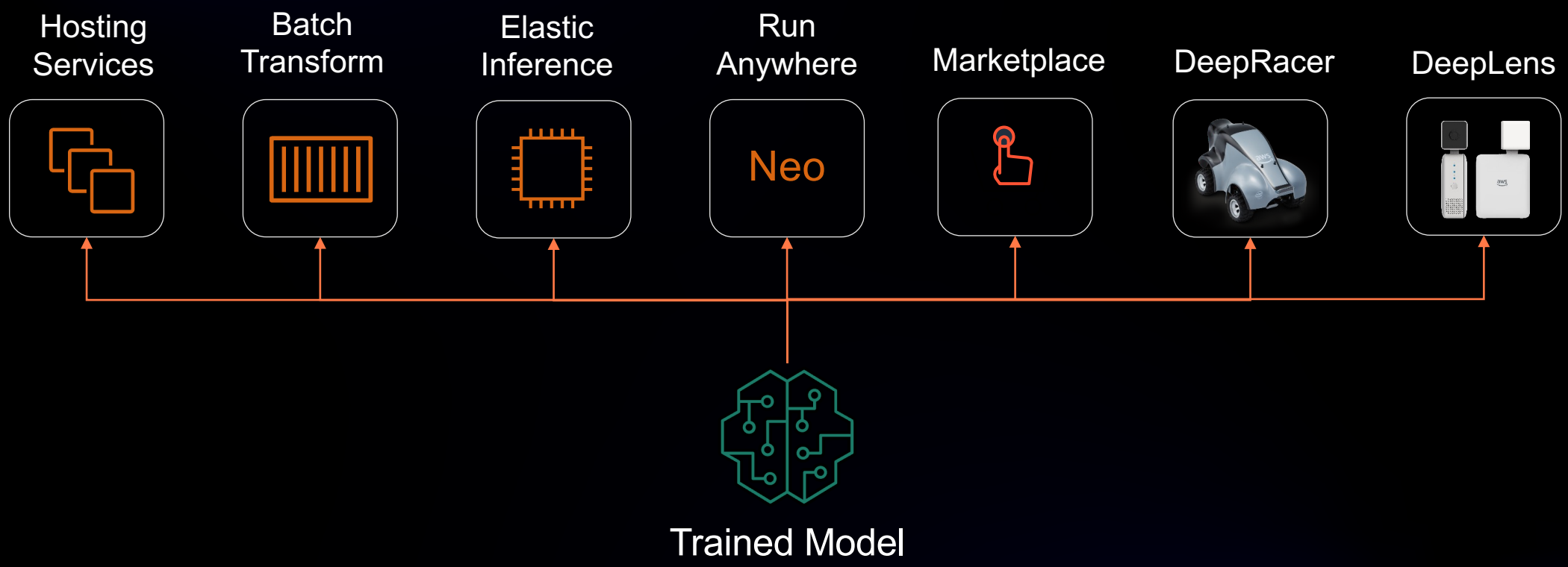
Model Parallelism

- The model is partitioned across multiple nodes.
- Each node contains a subset of the model through which the data flows and the transformations are shared and complied.
- Efficiency driven by a pipeline execution schedule.

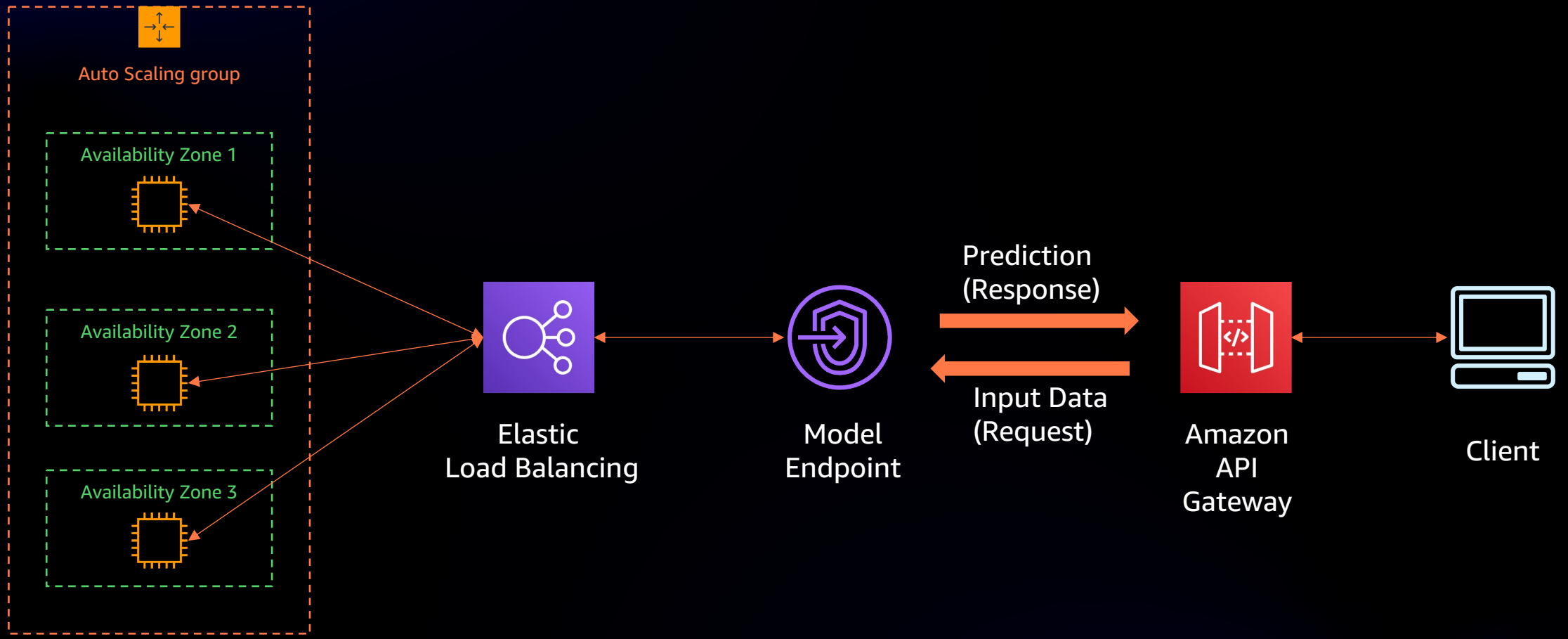
Distributed Training – Data Parallel



Amazon SageMaker Inference Options



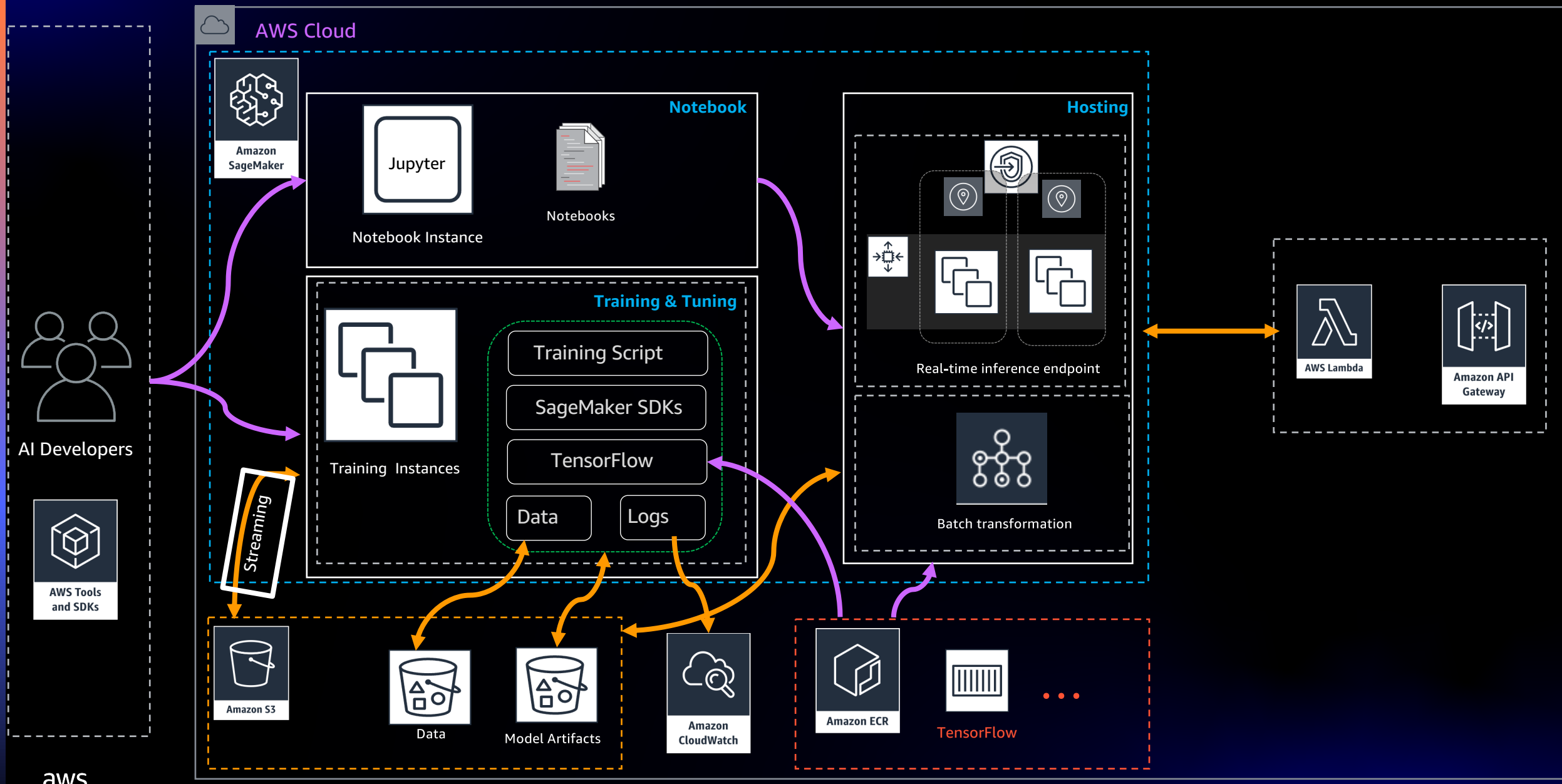
SageMaker Endpoints



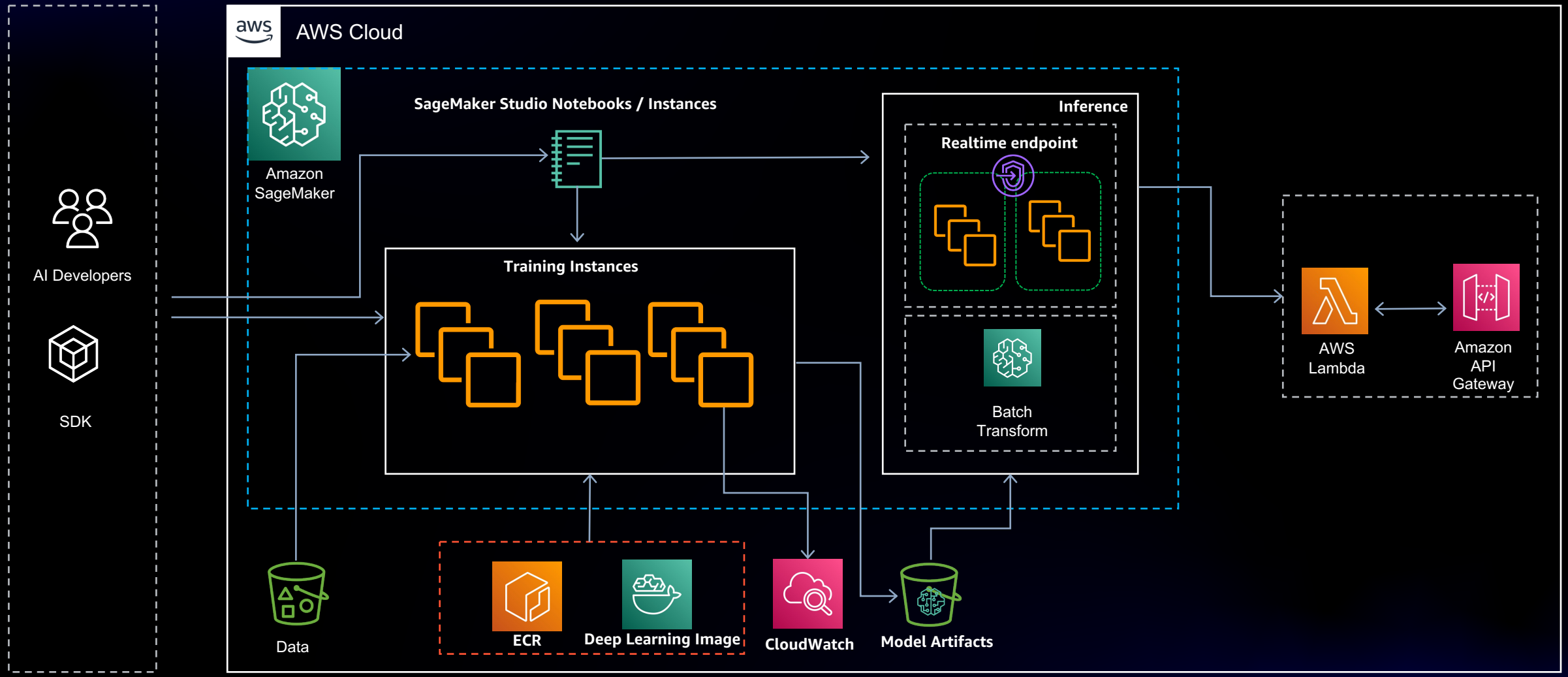
Deployment / Hosting
Amazon SageMaker ML
Compute Instances

aws

End to End Flow



Build, Train and Deploy



Q & A

Thank you!

Sam Palani

@samx18

Tim O'Brien

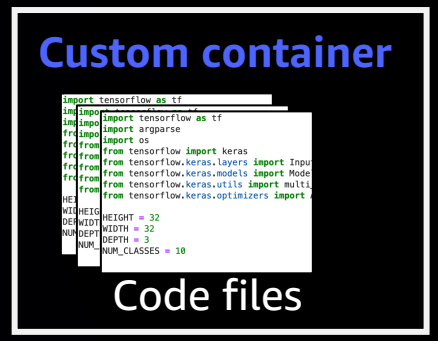
tpobrien@amazon.com
linkedin.com/in/tobrien0

Please complete the session survey

2 Bring your own Docker container

SageMaker SDK

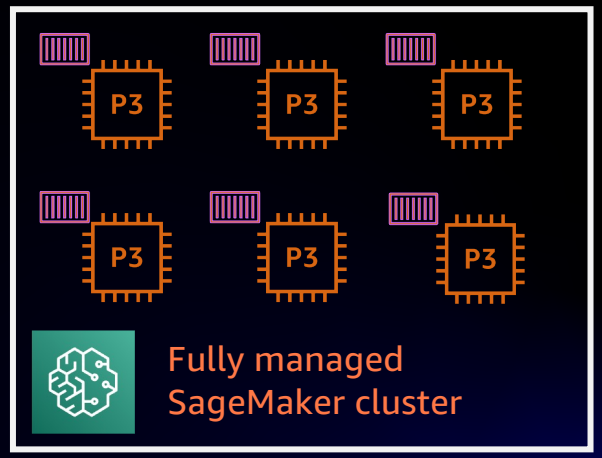
Docker build



Amazon ECR

Container registry

```
ic = sagemaker.estimator.Estimator(training_image,
                                   role,
                                   train_instance_count=1,
                                   train_instance_type='ml.p3.xlarge',
                                   train_volume_size = 50,
                                   train_max_run = 360000,
                                   input_mode= 'File',
                                   output_path=s3_output_location,
                                   sagemaker_session=sess)
```



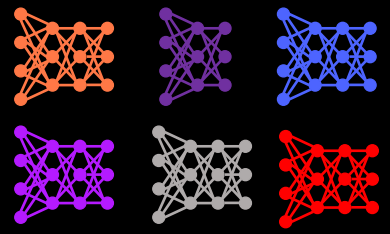
Amazon S3

aws

Challenges with deep learning

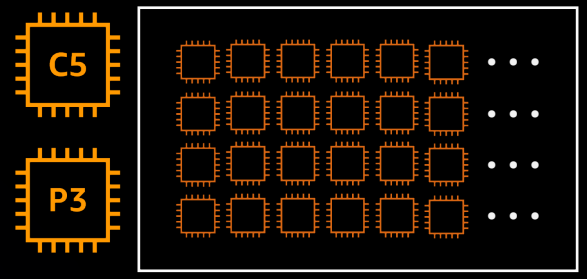
Many model architectures – difficult to get started

VGG, ResNet, ResNeXt, DenseNet, SqueezeNet, R-CNN, Faster R-CNN, SSD, YOLO, seq2seq, Transformers, and custom model architecture



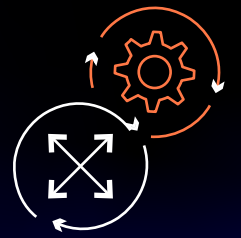
Computationally intensive to train and deploy

- Needs high-performance CPUs and GPUs
- Needs fast access to GBs and TBs of data for training
- Training on hundreds of CPUs and GPUs requires infrastructure management



Difficult to host and manage models in production

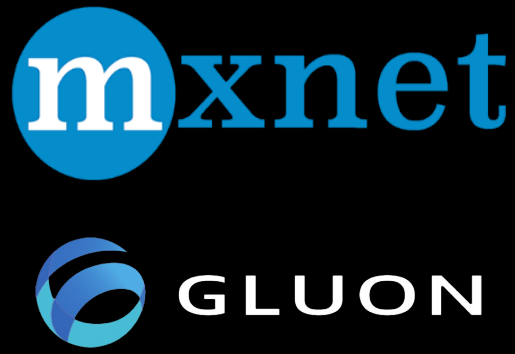
- Difficult to deliver high-performance and low-latency predictions
- Scaling to thousands and millions of users requires infrastructure management



aws

Deep learning frameworks

- Building blocks for designing, training, and validating deep neural networks



- High-level programming APIs with Keras and Gluon
- Performance optimizations to take advantage of GPUs
- Low-level functions for research and development
- Ability to run training at scale (but you will have to manage infrastructure)

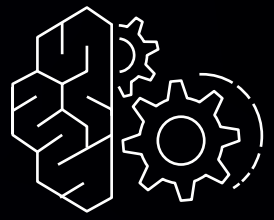
aws

1. Use SMP to automate your model partitioning

ANALYZED MODEL

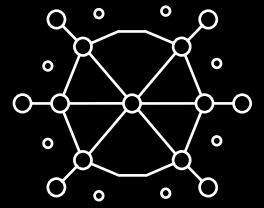
- Graph structure
- Sizes of trainable weights
- Sizes of exchanged tensors (using SageMaker Debugger)

RUN GRAPH PARTITIONING ALGORTIHM



- Balance stored weights and activations
- Minimize communication

PLACE PARTITIONS ON DEVICES

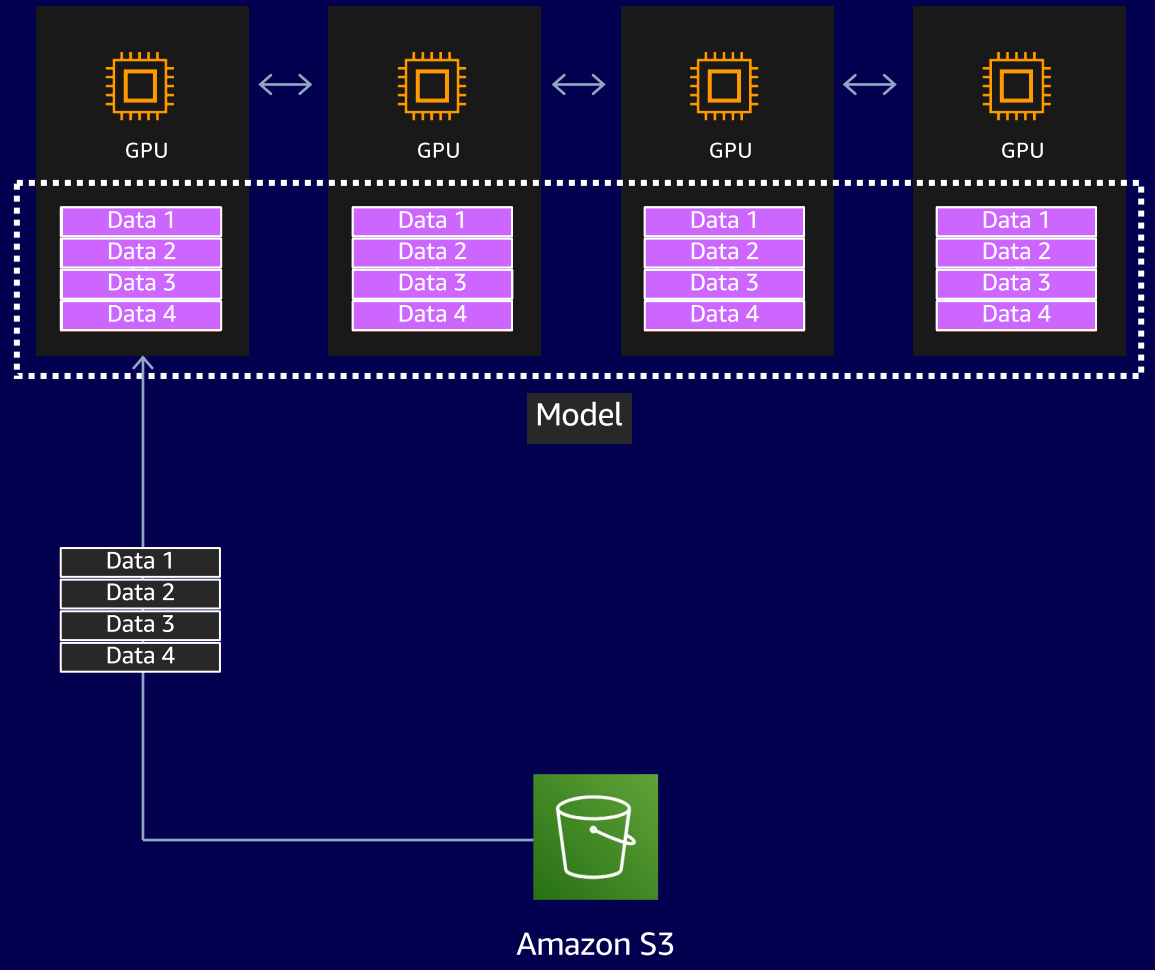


- To be executed in a pipelined manner

aws

2. Interleave pipeline execution to stabilize GPU utilization

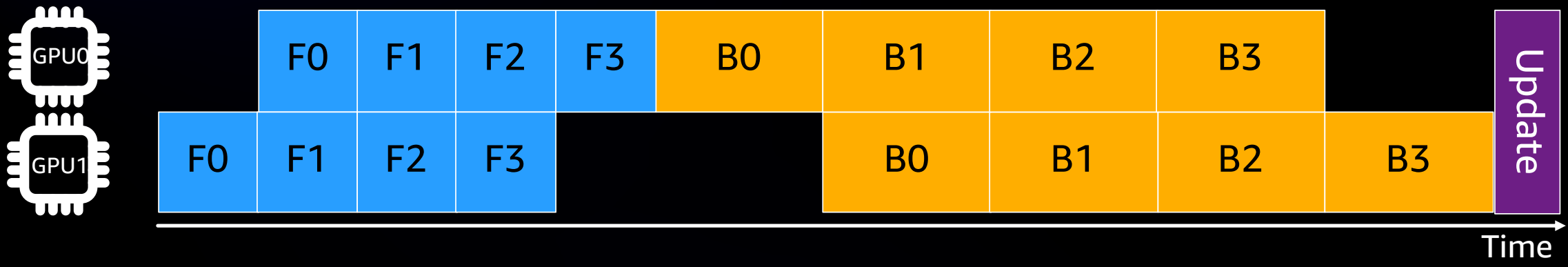
- Split minibatches into N “microbatches”
- Feed microbatches sequentially, but process them to keep GPU utilization more even
- Minimize “idle” time on GPUs



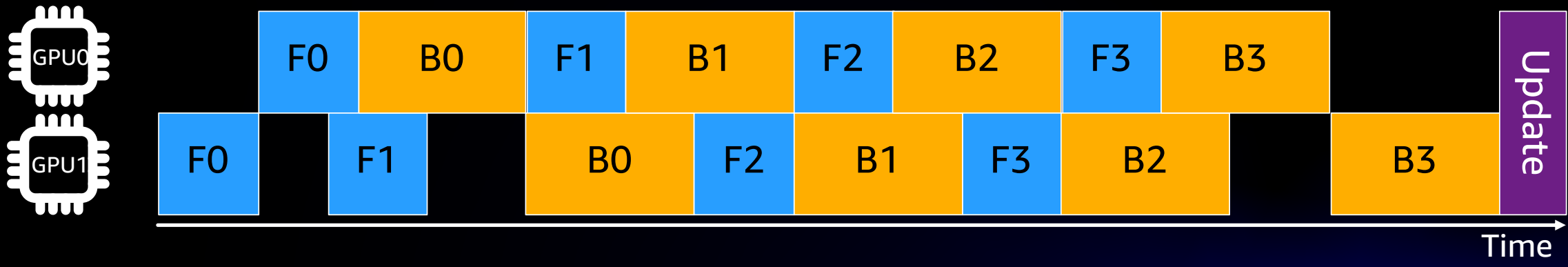
Pipeline execution schedule

Simple

F0 = forward pass, micro-batch 0
B1 = backward pass, micro-batch 1



Interleaved



aws