



T1ALML001

Training ML models at scale with Amazon SageMaker

Vatsal Shah

Sr. Solutions Architect
AWS India

Session overview

- Benefits and challenges of large-scale machine learning
- Amazon SageMaker accelerates training ML models at scale
- Distributed training
- A sample walkthrough of training a LLM
- Useful resources



Rise of large-scale models

"a picture of a very clean living room"



2017

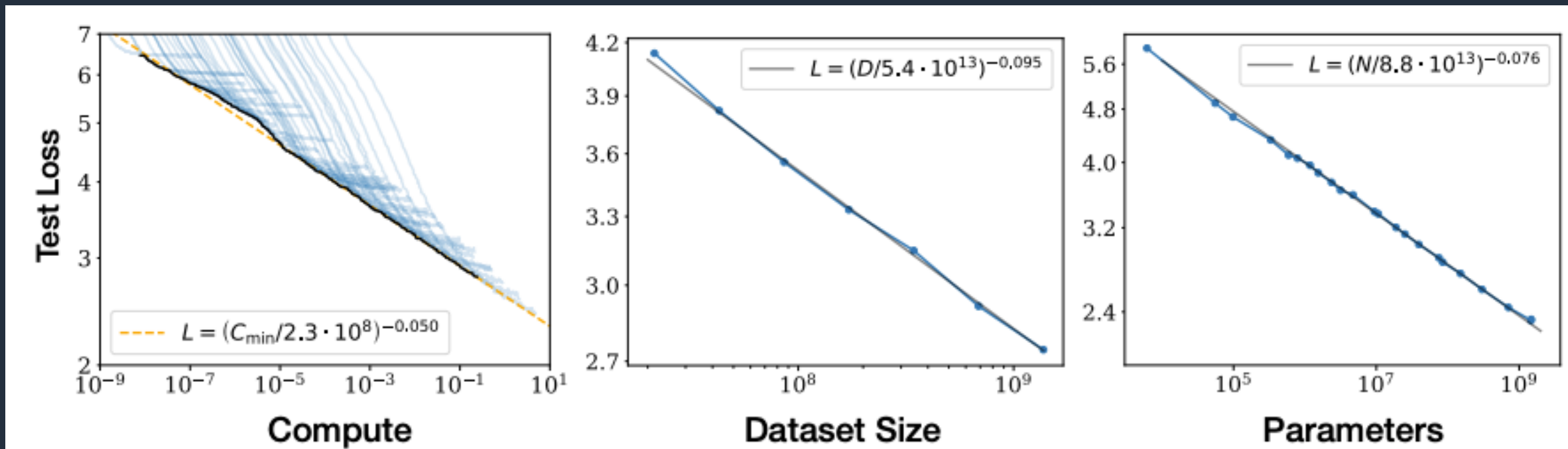
StackGAN,
Zhang et al.



2022

Stable Diffusion,
Rombach et al.

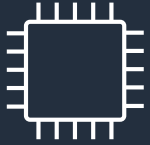
Large-scale models lead to better results



Scaling Laws for Neural Language Models

Kaplan et al., 2020

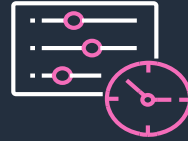
Challenges with training large-scale models



Hardware



Health checks



Orchestration



Data



Scaling up



Cost



Amazon SageMaker accelerates large-scale model training



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

Large-scale training on Amazon SageMaker

OPTIMIZED DISTRIBUTED TRAINING LIBRARIES & FRAMEWORKS

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

AMAZON SAGEMAKER TRAINING

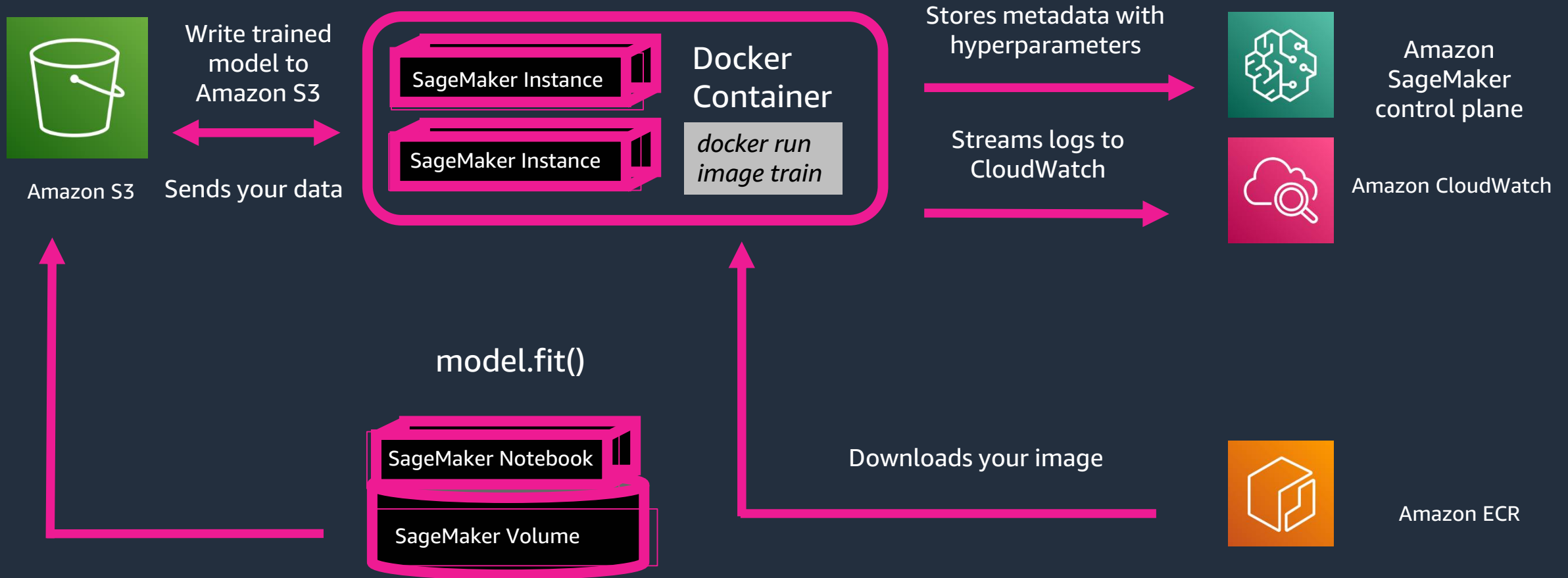
Large Scale Cluster Orchestration	NCCL Health Checks	Resilient training	SageMaker Compiler	Warm pools	SSH to container
Data loading	Debugger	Profiling	Experiment tracking	Hyperparameter optimization	Pay for what you use

ML COMPUTE INSTANCES & ACCELERATORS

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



Amazon SageMaker ephemeral training clusters



Train with your own deep learning model

```
from sagemaker.pytorch import PyTorch

estimator = PyTorch(entry_point = './cifar10.py',
                    role = role,
                    framework_version = '1.13',
                    py_version = 'py38',
                    instance_count = 1,
                    instance_type = 'ml.g5.xlarge',
                    hyperparameters = {'epochs': 50, 'batch_size': 32},
                    metric_definitions = [{'Name': 'train:loss', 'Regex': 'loss: (.*?)'}])

estimator.fit("s3://bucket/path/to/training/data")
```

Replicate experimental results by default

pytorch-training-2022-04-14-20-33-18-654

Clone

Create model package

Stop

Create model

Job settings

Job name	Status	SageMaker metrics time series	IAM role ARN
pytorch-training-2022-04-14-20-33-18-654	<div>Completed</div> <div>View history</div>	Enabled	arn:aws:iam::524898879256:role/service-role/AmazonSageMaker-ExecutionRole-20210323T152430
ARN	Creation time	Training time (seconds)	
arn:aws:sagemaker:us-west-2:524898879256:training-job/pytorch-training-2022-04-14-20-33-18-654	Apr 14, 2022 20:33 UTC	464	
	Last modified time	Billable time (seconds)	
	Apr 14, 2022 20:42 UTC	464	
		Managed spot training savings	
		0%	
		Tuning job source/parent	
		-	

Algorithm

Algorithm ARN	Instance type	Additional volume size (GB)	Volume encryption key
-	ml.g5.xlarge	30	-
Training image	Instance count	Maximum runtime (s)	
763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-training:1.10.0-gpu-py38	1	86400	
Input mode		Maximum wait time for managed spot training(s)	
File		-	



Easy and performant data loading



Amazon Simple Storage
Service (S3)

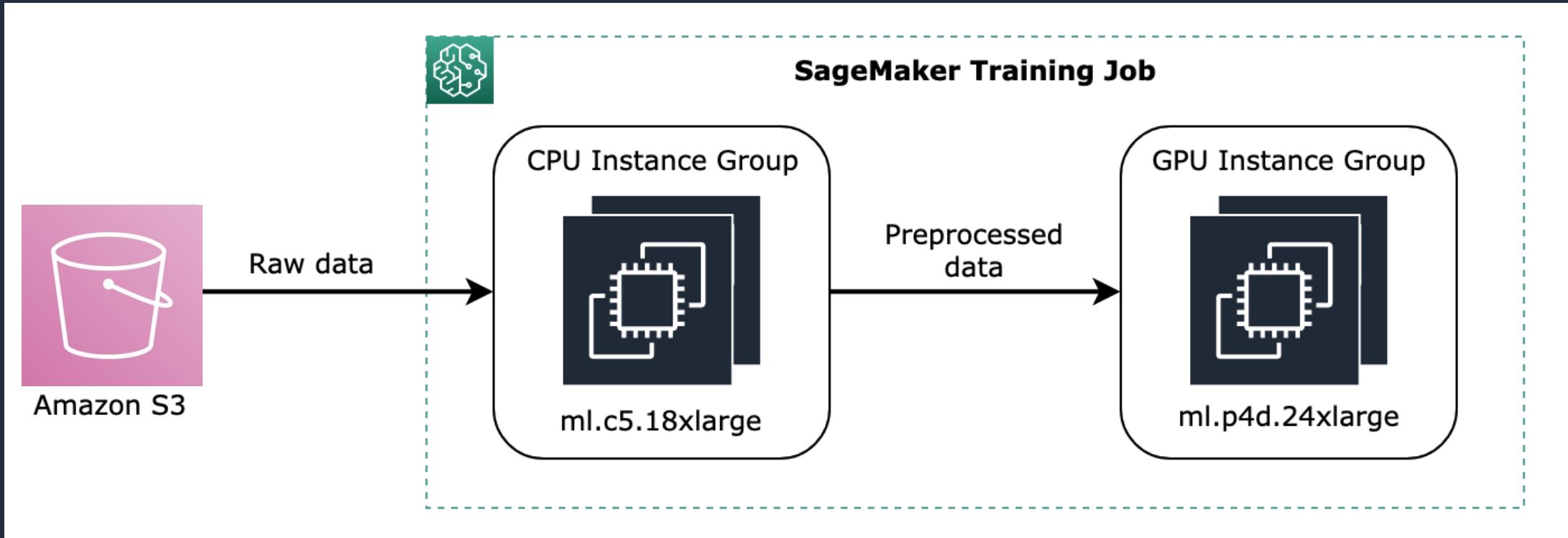


Amazon Elastic File
System (EFS)



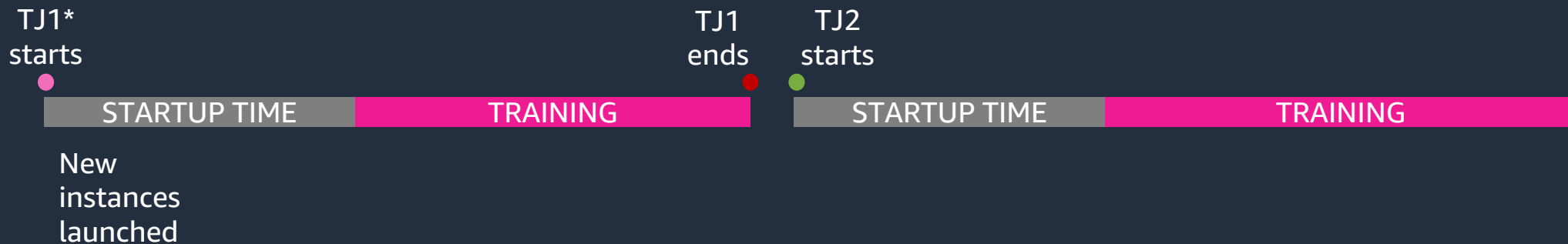
Amazon FSx
for Lustre

Heterogeneous clusters: Better GPU utilization



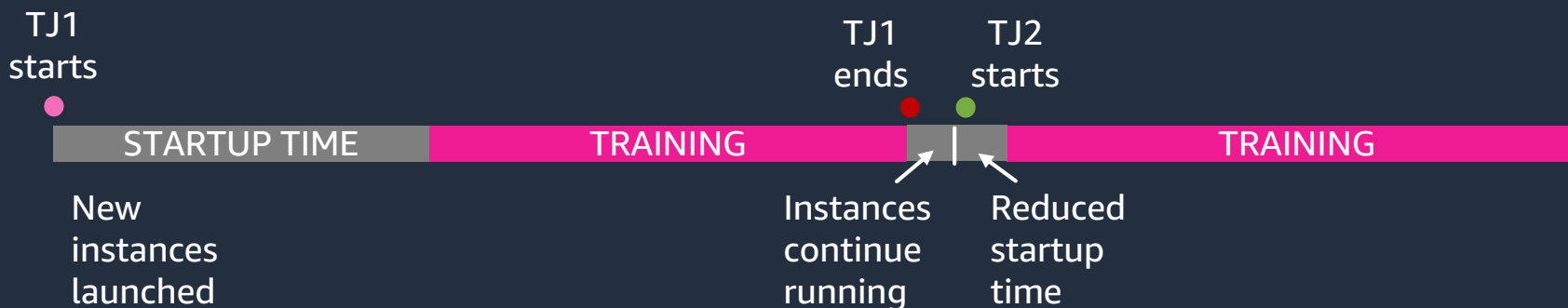
Warm Pools: Faster startup time

Before: Multi-minute wait between script updates



After: Multi-*second* wait between script updates

`Keep_alive_period_in_seconds=600`



Distributed Training with Amazon SageMaker



Why do we need distributed training?

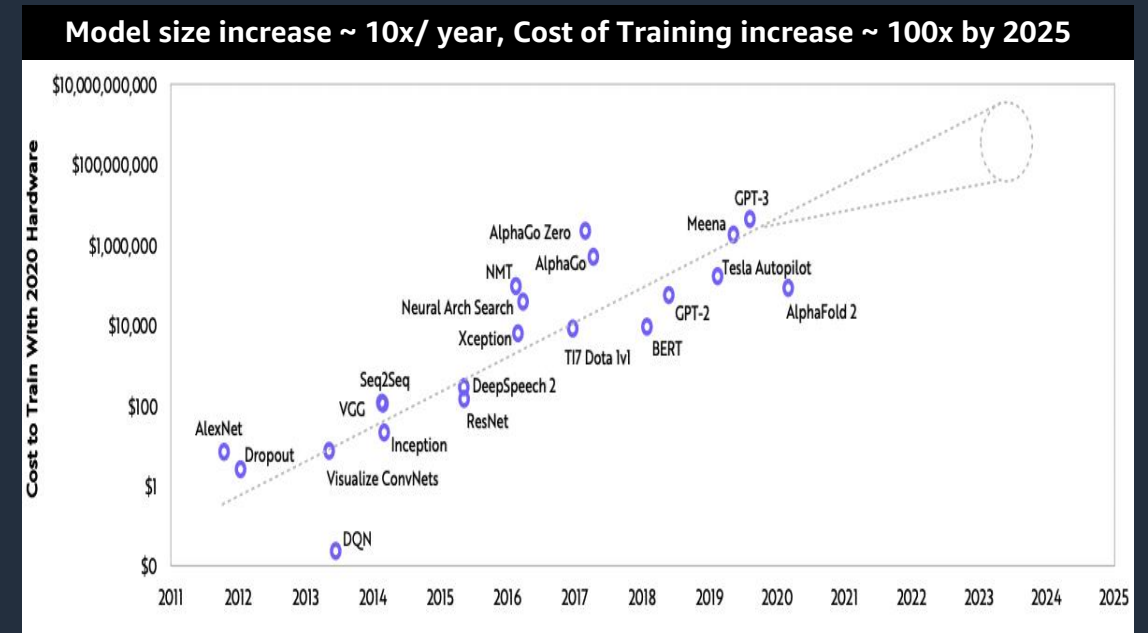
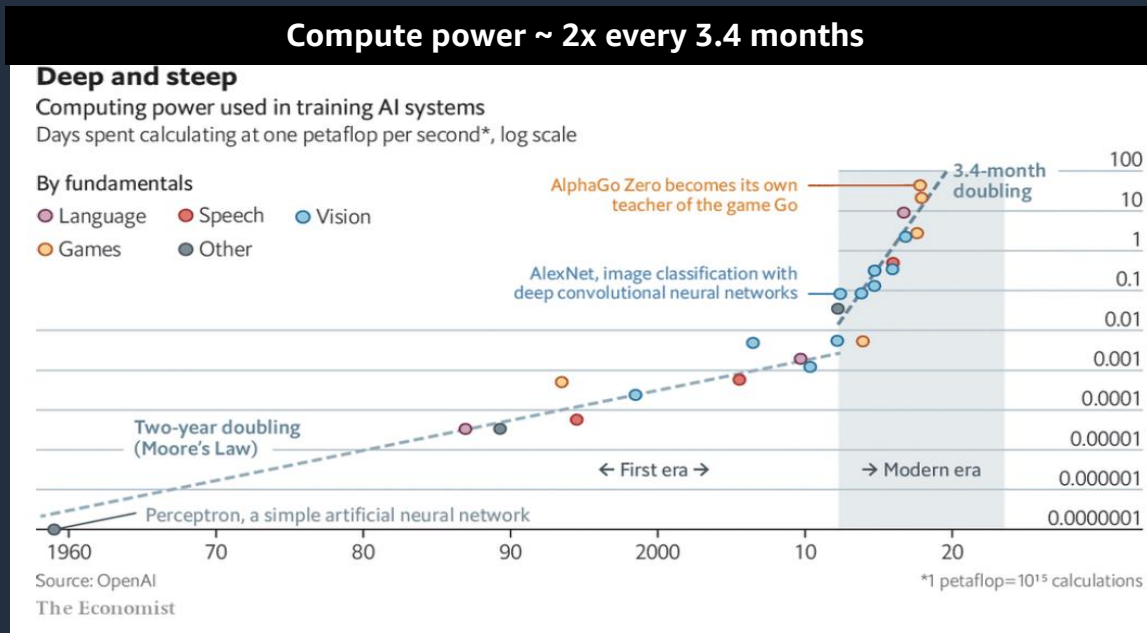
INCREASING COMPLEXITY

MODELS GROW FASTER THAN HARDWARE, LEADING TO BOTTLENECKS

- Businesses need higher precision in their model predictions
- Results in larger and more complex models
- Requires frequent retraining of models

INCREASING COSTS

- Increasing compute power required for frequent training of larger models drives up cost to train
- Becomes a barrier for innovation and growth



What are large language models (LLMs)?



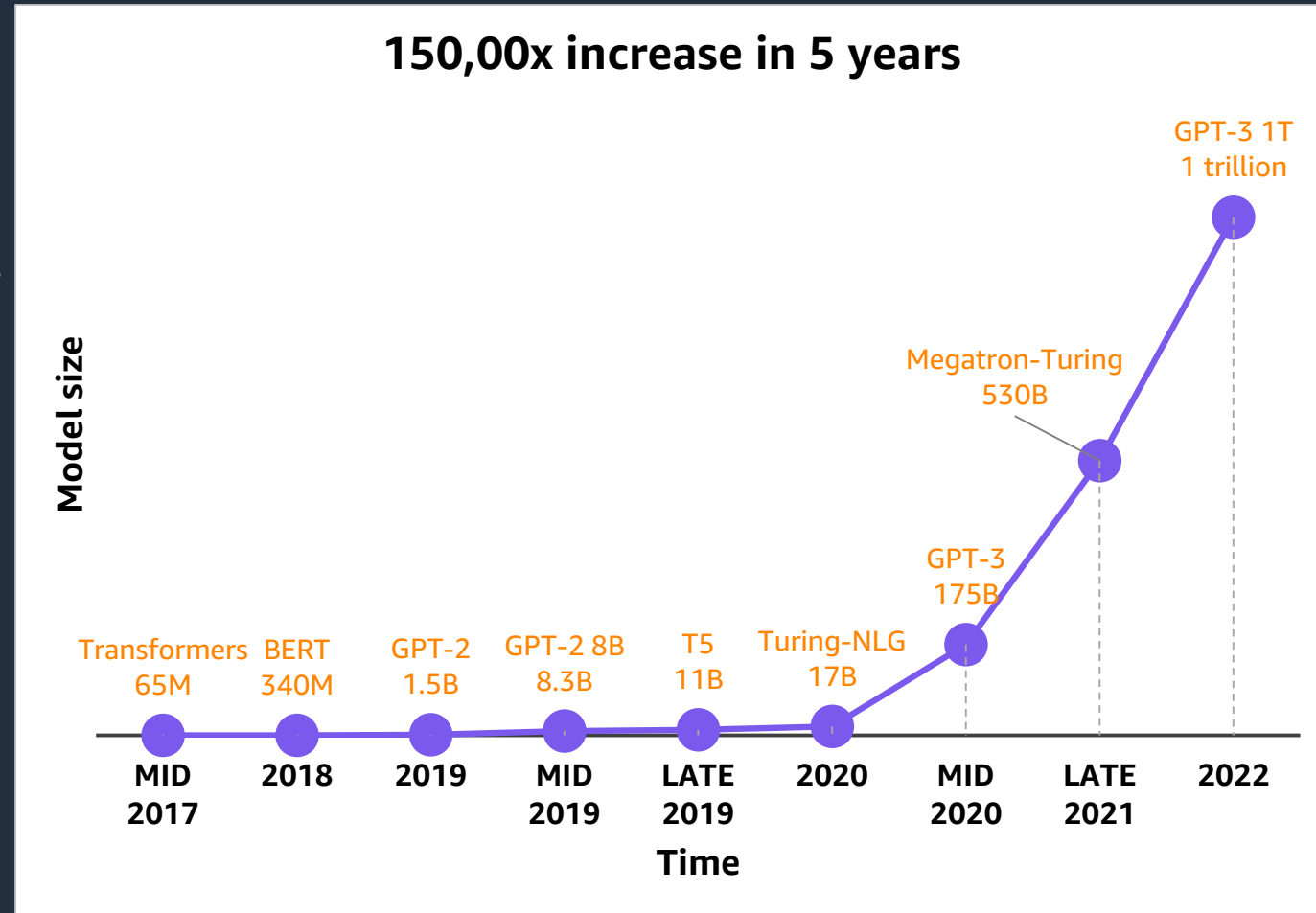
Size of large NLP models is increasing

Models are becoming more complex, with ML practitioners moving from classical ML to deep learning

State-of-the-art deep learning models are getting larger and larger as we find that larger models generalize better

They can be applied to a range of tasks to create more powerful digital assistants and generate better search results and product recommendations

They can be used for multi-modal use cases like answering visual questions from people who are vision-impaired, answering questions visually, emotion recognition, and many more



Pretraining vs. fine-tuning LLMs

	Pretraining	Fine tuning
Training duration (and cost)	Weeks to months	Minutes to hours
Customization	FULL <ul style="list-style-type: none">• NN architecture and size• Vocabulary size• Context length• Training data	Some <ul style="list-style-type: none">• Specific task tuning• Added domain-specific training data
Expertize needed	High	Low

Challenges in pretraining LLMs



Time to
train

Cost to
train

Expertise
required

Training LLMs in Amazon SageMaker

DISTRIBUTED TRAINING LIBRARIES

 TensorFlow	 PyTorch	SageMaker Distributed Training	OSS libraries
--	---	--------------------------------	---------------

AMAZON SAGEMAKER PLATFORM

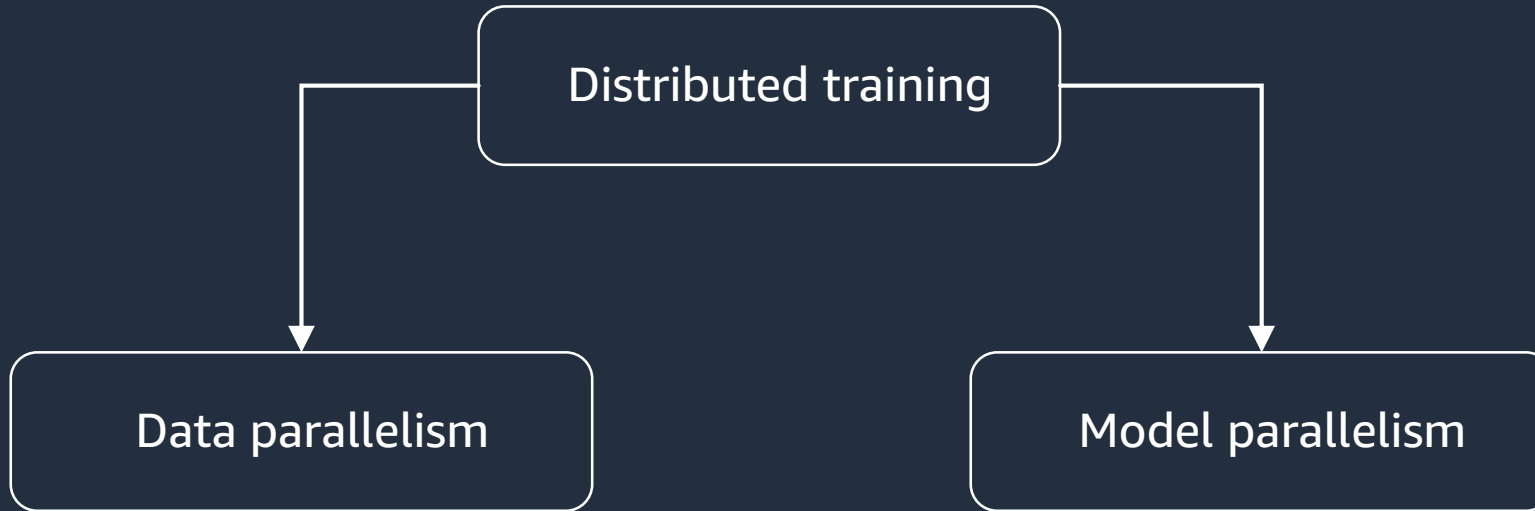
Large Scale Cluster Orchestration	Fault tolerant	 Hugging Face integration	SageMaker Training Compiler
-----------------------------------	----------------	--	-----------------------------

ML COMPUTE INSTANCES & ACCELERATORS

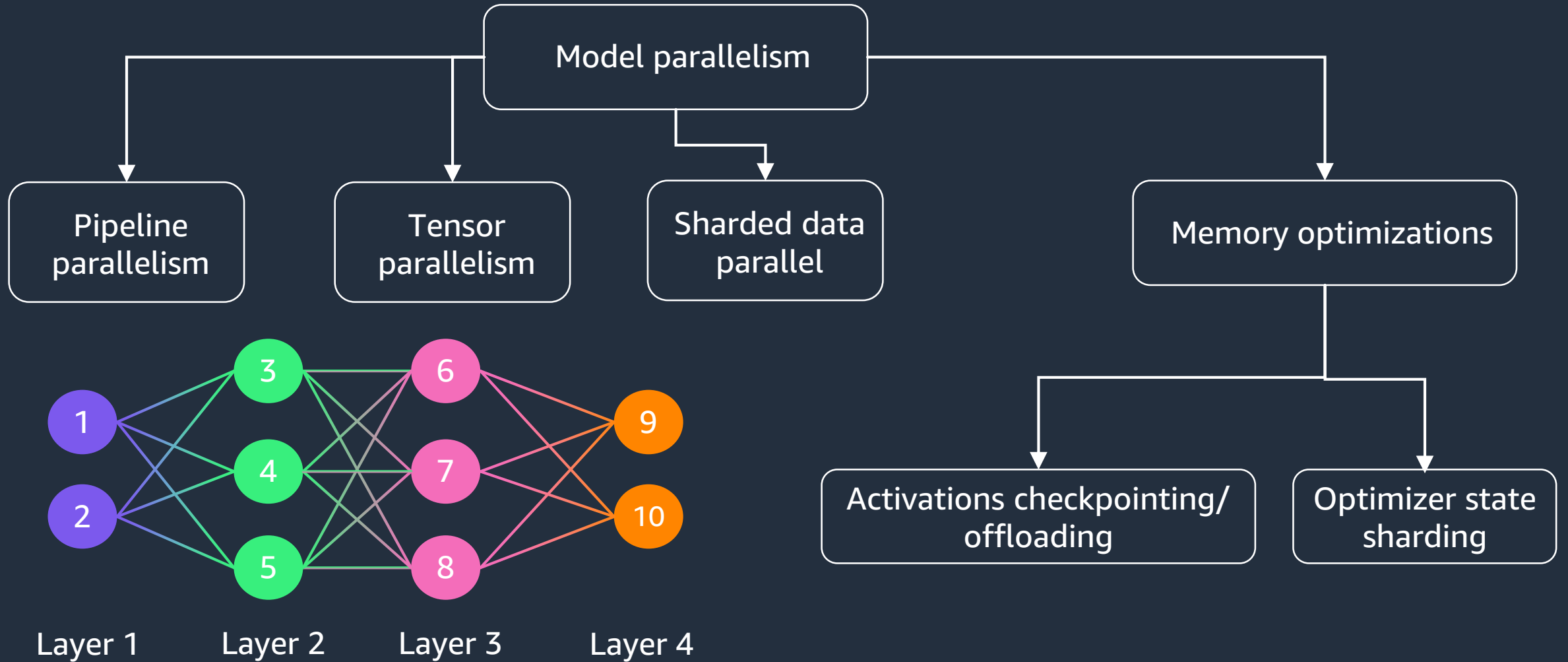
400/800 Gbps EFA networking	NVIDIA A100 GPUs 40 GB/80 GB	AWS Trainium	Habana Gaudi
-----------------------------	---------------------------------	--------------	--------------



Distributed training methods



Amazon SageMaker model parallelism options



**Data parallel,
think “massive data”**



Historical approaches to distributed gradient descent



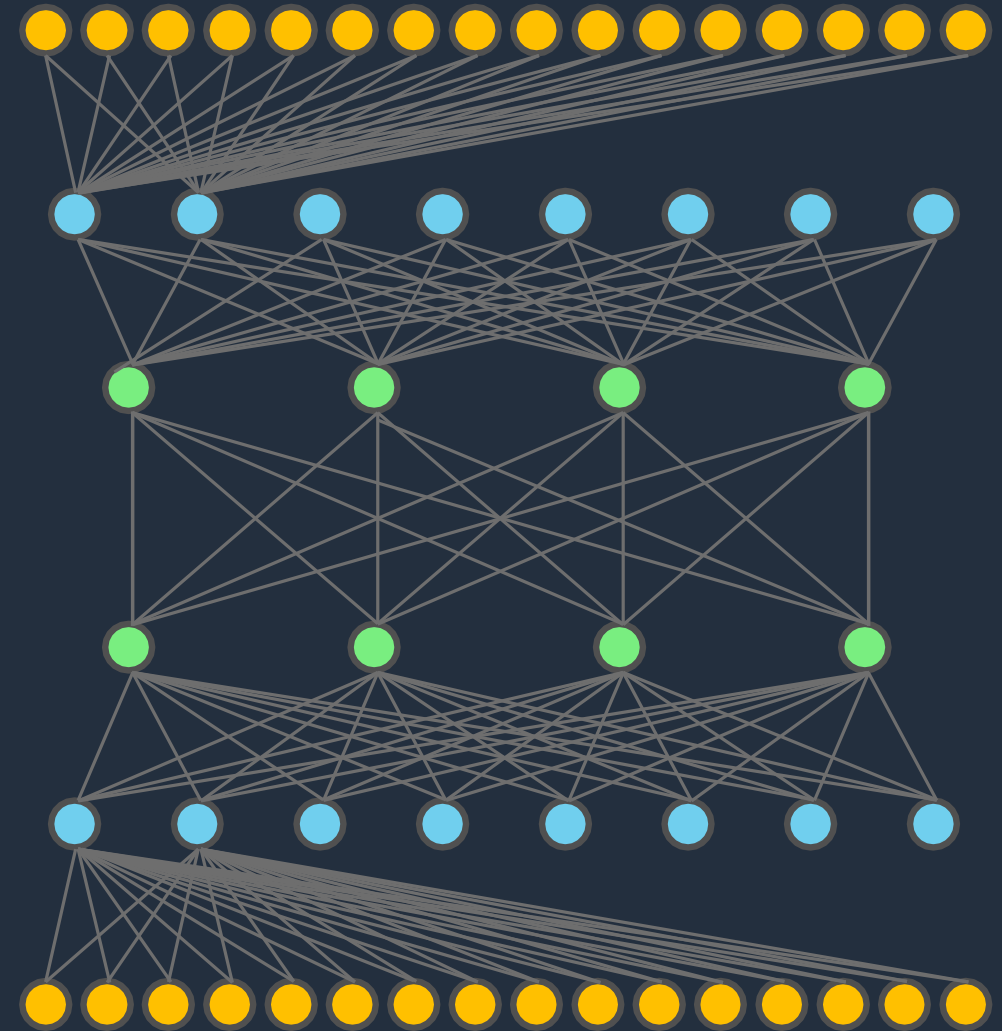
Parameter server
TensorFlow
ParameterServerStrategy



MPI AllReduce
Horovod
PyTorch
DistributedDataParallel

Amazon SageMaker Distributed Data Parallel

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



Run PyTorch Lightning and native PyTorch DDP on Amazon SageMaker Training, featuring Amazon Search

by Emily Webber, Abhinandan Patni, Mayank Jha, Karan Dhiman, Eiman Elnahrawy, and Vishwa Karia | on 18 AUG 2022 | in [Amazon SageMaker](#) | [Permalink](#) | [Comments](#) | [Share](#)

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

- 1 line of code to plug in PyTorch Lightning with SageMaker Distributed Data Parallel backend
- Easily run native PyTorch DDP on SageMaker Training
- Amazon Search saw 7.3x speedup moving to distributed training on SageMaker

Use the SageMaker Distributed Data Parallel Library as the Backend of `torch.distributed`

To use the SageMaker distributed data parallel library, the only thing you need to do is to import the SageMaker distributed data parallel library's PyTorch client (`smdistributed.dataparallel.torch.torch_smddp`). The client registers `smddp` as a backend for `torch.distributed`. When you initialize the PyTorch distributed process group using the `torch.distributed.init_process_group` API, make sure to pass `'smddp'` to the `backend` argument.

```
import smdistributed.dataparallel.torch.torch_smddp
import torch.distributed as dist

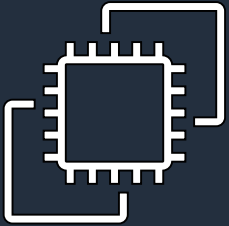
dist.init_process_group(backend='smddp')
```

```
ddp = DDPStrategy(
    cluster_environment=env,
    process_group_backend="smddp",
    accelerator="gpu")
```

Model parallel, think “massive models”



Model parallelism on Amazon SageMaker (SMP)



**Automated
model partitioning**



**Interleaved
pipelined training**



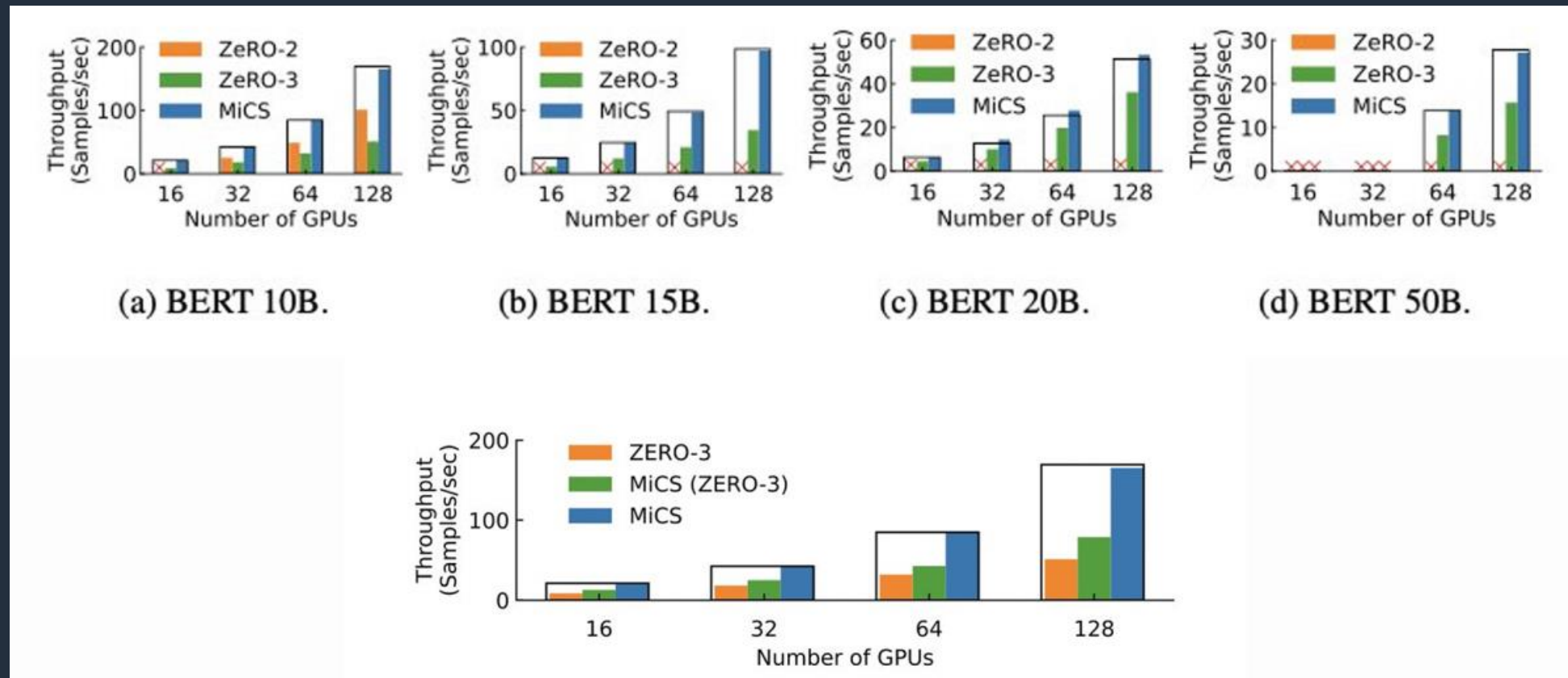
**Managed
Amazon SageMaker
training**



**Clean
framework integration**

Scale near linearly with *Sharded Data Parallelism*

MiCS achieves 169 TFLOPS per GPU with 175B parameter model on AWS p4de.24xlarge instances



- MiCS hits 99.4% of linear-scaling efficiency from 128 to 512 GPUs

- DeepSpeed hits only 72% , saturates at 62 TFLOPS per GPU

Available within Amazon SageMaker Model Parallel
2.8x faster than DeepSpeed



Steps to train 10 TB on 200 SageMaker GPUs

Data download

18 parallel CPU jobs
SageMaker Training

Local dev and test

2 CPUs <=> 96 CPUs
SageMaker Studio

Lightweight job test

1 Small GPU
SageMaker Training

FSx for Lustre

Set up VPC
Establish data repository
SageMaker Training

Build data index

2 CPUs <=> 96 CPUs
SageMaker Studio

Run final job

200 GPUs
SageMaker Training
10x faster than 1 node baseline
Training loop completed in 15 minutes



1/ Download data with CPU job parallelism


```
for p_file in parquet_list[:18]:  
    part_number = p_file.split('-')[1]  
    output_dir = "s3://{}/data/part-{}/".format(bucket, part_number)  
    if is_open(output_dir):  
        est = get_estimator(part_number, p_file, output_dir)  
        est.fit({"parquet": "s3://{}/meta/{}/".format(bucket, p_file)}, wait=False)
```

2/ Create Lustre in minutes from S3 path




Calculate total size [Info](#) Close

ⓘ The information below will no longer be available after you navigate away from this page.

Summary		
Source	Total number of objects	Total size
s3://laion-5b	1,541,244	9.5 TB

Specified objects						
<input type="text" value="Find objects by name"/>						
< 1 >						
Name	Type	Last modified	Size	Total number of objects	Error	
 data/	Folder	-	9.5 TB	1541244	-	

FSx > File systems

File systems (1)									
<input type="text" value="Filter file systems"/>									
< 1 > ⚙									
	File system name	File system ID	File system type	Status	Deployment type	Storage type	Storage capacity	Throughput capacity	Creation time
	laion	fs-08506f15b97761a0f 	Lustre	 Available	Persistent 2	SSD	12,000 GiB	12000 MB/s	2022-11-02T09:27:15-07:00

3/ Run 96 CPUs on Studio to develop and test

The screenshot displays the Amazon SageMaker Studio interface. On the left sidebar, there are sections for 'RUNNING INSTANCES', 'RUNNING APPS', 'KERNEL SESSIONS', and 'TERMINAL SESSIONS'. The 'RUNNING INSTANCES' section shows two Jupyter Notebook instances: 'ml.c5.24xlarge' (96 vCPU + 192 GiB) and 'ml.t3.medium' (2 vCPU + 4 GiB). The 'RUNNING APPS' section shows two 'datascience-1.0' applications. The 'KERNEL SESSIONS' section shows a 'Stable_Diffusion.ipynb' kernel session on an 'ml.c5.24xlarge' instance. The 'TERMINAL SESSIONS' section shows a 'terminals/1' terminal session.

The main area displays the 'Stable_Diffusion.ipynb' Jupyter Notebook. The notebook has two tabs: 'finetune.py' and 'Stable_Diffusion.ipynb'. The 'Stable_Diffusion.ipynb' tab is active, showing a code cell with the following code:

```
[4]: dataset = load_index_dataset('')  
[5]: dataset  
[5]: DatasetDict({  
      train: Dataset({  
            features: ['image', 'caption'],  
            num_rows: 50000000  
          })  
})  
[6]: !head -10 data_index.jsonl
```

The output of the code cell shows the first 10 lines of the 'data_index.jsonl' file, which are JSON objects containing image paths and captions:

```
{"image": "/opt/ml/input/data/training/laion-fsx/part-00018/07846037.jpg", "caption":  
"Join Ms. Monica for Virtual Storytime each Tuesday and Thursday at 10:30 a.m."}  
{"image": "/opt/ml/input/data/training/laion-fsx/part-00018/09755358.jpg", "caption":  
"Free Printable Ace Hardware Coupons"}  
{"image": "/opt/ml/input/data/training/laion-fsx/part-00018/03759897.jpg", "caption":  
"A physical distancing sign is seen during a media tour of Hastings Elementary school i  
n Vancouver on September 2, 2020. THE CANADIAN PRESS/Jonathan Hayward"}  
{"image": "/opt/ml/input/data/training/laion-fsx/part-00018/16803873.jpg", "caption":  
"Trendy Ruby Red Crystal Rhinestone Christmas Tree Holiday Brooch Pendant - Trendy Chr  
istmas Trees"}
```

4/ Scale to 200 GPUs with SageMaker Training

```
est = HuggingFace(entry_point='finetune.py',
                  source_dir='src',
                  image_uri=image_uri,
                  sagemaker_session=sess,
                  role=role,
                  output_path="s3://laion-5b/output/model/",
                  instance_type='ml.p4dn.24xlarge',
                  keep_alive_period_in_seconds = 60*60,
                  py_version='py38',
                  base_job_name='fsx-stable-diffusion',
                  instance_count=24,
                  enable_network_isolation=True,
                  encrypt_inter_container_traffic = True,
                  # all opt/ml paths point to SageMaker training
                  hyperparameters = hyperparameters,
                  distribution={"smdistributed": { "dataparallel": { "enabled": True } }},
                  max_retry_attempts = 30,
                  max_run = 4 * 60 * 60,
                  debugger_hook_config=False,
                  disable_profiler = True,
                  **kwargs)

est.fit(inputs=data_channels, wait=False)
```

5/ Host on SageMaker Real-Time Endpoints

```
import sagemaker
from sagemaker.deserializers import JSONDeserializer
from sagemaker.serializers import JSONSerializer

sess = sagemaker.Session()

endpoint_name = 'sd-inference-gpu-2022-10-16-14-46-56-112'

pred = sagemaker.predictor.Predictor(endpoint_name,
                                       sagemaker_session=sess,
                                       serializer=JSONSerializer(),
                                       deserializer=JSONDeserializer())

prompt = "a Christmas tree in Las Vegas"

output = pred.predict({'inputs':prompt})
process_result(output['images'][0])
```



Start your LLM journey today



Get started now
It's free!



Jurassic-1: Technical Details and Evaluation
White paper covering Large and Jumbo



Read the docs
API documentation, guides and more



Introducing J1-Grande
Blog post with benchmark results and samples



Build a sentiment analysis dashboard
Extract insights from reviews in 15 minutes



Build a CV Profile Generator
Teach Jurassic-1 Grande to generate a profile for a given role and skills from just a few examples



Getting started



Product page

aws.amazon.com/sagemaker/train/



Technical Documentation

docs.aws.amazon.com/sagemaker/latest/dg/how-it-works-training.html



SageMaker examples on GitHub

github.com/aws/amazon-sagemaker-examples



AWS ML blogs

<https://aws.amazon.com/blogs/machine-learning/>





Thank you!

Vatsal Shah

vatsalbshah 

