Train and deploy Stable Diffusion using AWS Trainium & AWS Inferentia

Vijay Niles
Senior Solutions Architect
Amazon Web Services

Steven Alyekhin
Senior Solutions Architect
Amazon Web Services
Introduction to generative AI and stable diffusion

Why build generative AI on AWS?

Demo

AWS AI accelerators: AWS Trainium and AWS Inferentia2

Q&A
Generative AI is changing the way we work
Question: What is generative AI?

Generative AI is a branch of AI that focuses on creating new data. It is a subset of machine learning. The goal of generative AI is to create new data that is similar to the data that was used to train the model.
The road to generative AI

Word embeddings

Language models

Word2vec

Natural language processing (NLP)
  Text summarization
  Question answering
  Sentiment analysis
  Speech language understanding
Applications for generative AI

- Text generation (many types)
- Game design
- Industrial design
- Drug design research
- Image generation . . .
Text-to-image models
(e.g., Stable Diffusion)

Language models
(word embeddings)

Image models
(image embeddings)
Stable Diffusion examples

Diffusion models

"Marigold the puppy"

Stable Diffusion training

"a photo of Marigold the puppy as 8bit art"
Why build generative AI on AWS?
The AI/ML flywheel

1. Innovation (new/better algorithms)
2. Applications (more use-cases)
3. Demand (wider adoption)
4. Investment (research, optimization)
The AI/ML flywheel

AWS AI/ML Services
(Chips, Servers, Services) → Better $/perf
New Capabilities

Investment
(research, optimization)

Demand
(wider adoption)

AI/ML Growth

Innovation
(new/better algorithms)

Applications
(more use-cases)
AI models are getting bigger

Model size
(# of parameters)

Perceptron 1
AlexNet 62M
VGG16 138M
YOLO, GNMT 210M
BERT-L 340M

YEAR

1957 ... 2012 ... 2014 2016 ... 2018 2019 2020 2021 2022

SWITCH-C 1.6T
GPT-3 175B
GPT-2 1.5B

10x/year growth

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.
What’s not going to change?

- High performance
- Cost effectiveness
- Ease of use
AWS purpose-built accelerators for generative AI

**AWS Trainium**
- Cost-efficient, high-performance training of LLMs and diffusion models
- Up to 50% cost-to-train savings

**AWS Inferentia2**
- High performance at the lowest cost per inference for LLMs and diffusion models
- Up to 40% better price performance for generative AI
Demo – Train and deploy Stable Diffusion on Trn1 and Inf2
AWS Trainium
Amazon EC2 Trn1 and Trn1n instances powered by AWS Trainium

COST-EFFECTIVE, HIGH-PERFORMANCE DL TRAINING INSTANCES

High performance on training of popular NLP models on AWS

Up to 50% cost-to-train savings

Up to 4x network bandwidth

<table>
<thead>
<tr>
<th>Instance size</th>
<th>vCPUs</th>
<th>Instance memory</th>
<th>Trainium chips</th>
<th>Accelerator memory</th>
<th>NeuronLink</th>
<th>Instance networking</th>
<th>On-demand price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trn1.2xlarge</td>
<td>8</td>
<td>32 GB</td>
<td>1</td>
<td>32 GB</td>
<td>N/A</td>
<td>Up to 10 Gbps</td>
<td>$1.34/hr</td>
</tr>
<tr>
<td>Trn1.32xlarge</td>
<td>128</td>
<td>512 GB</td>
<td>16</td>
<td>512 GB</td>
<td>Yes</td>
<td>800 Gbps</td>
<td>$21.5/hr</td>
</tr>
<tr>
<td>Trn1n.32xlarge</td>
<td>128</td>
<td>512 GB</td>
<td>16</td>
<td>512 GB</td>
<td>Yes</td>
<td>1600 Gbps</td>
<td>$24.78/hr</td>
</tr>
</tbody>
</table>

Trn1 available now in US-East-1 (N. Virginia) and US-West-2 (Oregon)
Trn1 delivers high performance

**BEST-IN-CLASS THROUGHPUT AND COST**

**Trn1 single node:** 1.2x faster  
**Trn1 cluster:** 1.5x faster

**Throughput**

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>Seq/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,000</td>
</tr>
<tr>
<td>16</td>
<td>62,000</td>
</tr>
</tbody>
</table>

**Training cost**

<table>
<thead>
<tr>
<th>Number of nodes</th>
<th>Training cost per million seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.00</td>
</tr>
<tr>
<td>16</td>
<td>5.75</td>
</tr>
</tbody>
</table>

**Comparable Training-optimized Amazon EC2 Instance**

- Trn1 single node: 1.8x lower cost  
- Trn1 cluster: 2.3x lower cost

Hugging Face BERT-Large Ph1 Pretrain; 128 Seqlen  
Trn1 – BF16(SR), p4d – Mixed Precision (BF16/FP32)
Ease of use

Bring your own model

JIT compile to Trainium

```python
import os
... import torch
import torch_xla
import torch_xla.core.xla_model
from transformers import BertForPreTraining

model = BertForPreTraining.from_pretrained('bert-large-uncased')

def train_loop_fn(model, optimizer, train_loader, device, epoch, global_step, training_ustep, running_loss):
    max_grad_norm = 1.0
    for i, data in enumerate(train_loader):
        training_ustep += 1
        input_ids, segment_ids, input_mask, masked_lm_labels, next_sentence_labels = data
        outputs = model(input_ids=input_ids,
                         attention_mask=input_mask,
                         token_type_ids=segment_ids,
                         labels=masked_lm_labels,
                         next_sentence_label=next_sentence_labels)
        loss = outputs[0].loss / flags.grad_accum_usteps
        loss.backward()
        running_loss += loss.detach()

        if (training_ustep + 1) % flags.grad_accum_usteps == 0:
            xm.mark_step()
            running_loss_cpu = running_loss.detach().cpu().item()
            running_loss.zero_()
            torch.nn.utils.clip_grad_norm_(model.parameters(), max_grad_norm)
            xm.optimizer_step(optimizer)

            optimizer.zero_grad()
            scheduler.step()
            global_step += 1
            if global_step >= flags.steps_this_run:
                break

            return global_step, training_ustep, running_loss
```
Rich data type selection

Choose the right data type for your workload

**Mantissa** (Precision)

- **Mantissa** (Precision)
- **Exponent (Range)**

- **Float32**
- **TFloat32**
- **BFLOAT16**
- **Float16**
- **C-Float8**
- **UInt8**

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Rich data type selection

CHOOSE THE RIGHT DATA TYPE FOR YOUR WORKLOAD

Performance

Ease of use
Rich data type selection

CHOOSE THE RIGHT DATA TYPE FOR YOUR WORKLOAD

PFLOPS per data type

- BFLOAT16: 1.4x
- TFLOAT32: >2.5x
- FLOAT16: >5x

© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.
AWS Inferentia2
Amazon EC2 Inf2 instances powered by AWS Inferentia2

**High performance at the lowest cost for generative AI models**

- Up to 4x higher throughput and 10x lower latency
- 9.8 TB/s aggregated accelerator memory bandwidth
- Support for ultra-large generative AI models

<table>
<thead>
<tr>
<th>Instance size</th>
<th>vCPUs</th>
<th>Instance memory</th>
<th>Inferentia2 chips</th>
<th>Accelerator memory</th>
<th>NeuronLink</th>
<th>Instance networking</th>
<th>On-demand price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inf2.xlarge</td>
<td>4</td>
<td>16 GB</td>
<td>1</td>
<td>32 GB</td>
<td>N/A</td>
<td>Up to 15 Gbps</td>
<td>$0.76/hr</td>
</tr>
<tr>
<td>Inf2.8xlarge</td>
<td>32</td>
<td>128 GB</td>
<td>1</td>
<td>32 GB</td>
<td>N/A</td>
<td>Up to 25 Gbps</td>
<td>$1.97/hr</td>
</tr>
<tr>
<td>Inf2.24xlarge</td>
<td>96</td>
<td>384 GB</td>
<td>6</td>
<td>192 GB</td>
<td>Yes</td>
<td>50 Gbps</td>
<td>$6.49/hr</td>
</tr>
<tr>
<td>Inf2.48xlarge</td>
<td>192</td>
<td>768 GB</td>
<td>12</td>
<td>384 GB</td>
<td>Yes</td>
<td>100 Gbps</td>
<td>$12.98/hr</td>
</tr>
</tbody>
</table>

Inf2 available now in US-East-1 (N. Virginia) and US-East-2 (Ohio)
**AWS Inferentia2 LLM performance**

**OPT-30B throughput (tokens/sec)**
- Comparable inference-optimized Amazon EC2 instance: 368.6 tokens/sec
- inf2.48xl: 619.6 tokens/sec
  - **65% higher**

**OPT-30B cost per million (USD)**
- Comparable inference-optimized Amazon EC2 instance: $122.7
- inf2.48xl: $59.15
  - **52% lower**

**OPT-66B throughput (tokens/sec)**
- Comparable inference-optimized Amazon EC2 instance: 65% lower
- inf2.48xl: 351 tokens/sec
  - **Out of memory**
Full stack AI/ML integration

Accelerated compute

Amazon EC2 Trn1/Trn1n, Inf2, Inf1
Amazon EC2 Trn1 UltraClusters

Storage & networking

Amazon Simple Storage Service (Amazon S3)
Amazon Elastic Block Store (Amazon EBS)
Amazon FSx for Lustre
Amazon Elastic File System (Amazon EFS)
Elastic Fabric Adapter

Frameworks & services

ML frameworks
Amazon SageMaker
AWS Deep Learning AMIs
AWS Deep Learning Containers
Amazon Elastic Kubernetes Service (Amazon EKS)
Amazon Elastic Container Service (Amazon ECS)
Getting started with AWS Trainium and AWS Inferentia

Launch instances (Trn1, Inf2, Inf1)
- AWS Deep Learning AMIs
- AWS Deep Learning Containers

Bring your own model
- PyTorch
- TensorFlow
- Hugging Face
- OpenXLA

Modify a few lines of code → run
- Neuron SDK

Monitor, tune, scale
- Neuron SDK

PyTorch, the PyTorch logo and any related marks are trademarks of The Linux Foundation
Getting started with Hugging Face Optimum Neuron

Hugging Face + Neuron SDK

Documentation & tutorials

Hugging Face documentation

Code examples

Access GitHub
Your time is now
Build in-demand cloud skills your way
Thank you!

Vijay Niles
vnniles@amazon.com

Steven Alyekhin
salyekh@amazon.com

Please complete the session survey in the mobile app.