intel ai Summit 英特爾 AI 科技論壇 Bringing Al Everywhere

Accelerating GenAl & LLM with Intel Gaudi Solutions

Leo Zhao Al Software Evangelist March 27th, 2024





Brief of Intel[®] Gaudi[®] Solution





GenAl and LLMs require...



Deep learning performance



Flexible, efficient scalability







Intel[®] Gaudi[®] Al Accelerator: Built for the Deep Learning and Large-Scale Eras





Intel[®] Al Portfolio



Intel[®] Gaudi[®] AI Accelerator Roadmap





Intel[®] Gaudi[®] 2 Accelerator: Architected for GenAI and Large Language Models

Heterogenous Al-custom compute engine

- 24 Tensor processor cores
- Dual matrix multiplication engines

Leading on-chip memory capacity

- 96 GB HBM2E
- 48 MB SRAM

Massive, flexible on-chip networking

24x 100 GbE RoCE ports on every Intel Gaudi2 chip





Intel[®] Gaudi[®] 2 Server: Designed for Flexible, Efficient Scalability

HLS-Gaudi 2 Reference Server featuring...

- 8 Intel Gaudi 2 mezzanine cards
- 24x 100 GbE ports per card
 - 21 for all-to-all connectivity to other 7 Intel Gaudi processors within the server
 - Three to scale out
 - Through six QSFP-DD ports
- Dual-socket Host CPU: Intel[®] Xeon[®] Scalable processor



PCle

All-to-All Connectivity 21x100G Eth

Dual-socket Intel Xeon Scalable Processor Host CPU



Near-Infinite System Scale-Out Possibilities to Meet Demands for the Largest GenAl Models







Intel[®] Gaudi[®] Performance in MLPerf





Intel[®] Gaudi[®] 2 Accelerator Performance Doubled with FP8

We projected for customers +90% performance gain with FP8



Committed to credible, reliable performance projections and delivering on them





Performance source: MLPerf Training 3.1 results https://mlcommons.org/benchmarks/training/

40-50% Better Price-Performance on GPT-3 vs. H100

Based on the most recent MLPerf Training benchmark and estimated range of server pricing, Intel Gaudi2 AI accelerator delivers between 40 and 50% better price performance than Nvidia H100.

- 1 of only 2 merchant silicon submissions for GPT-3
- H100/FP8 outperformed Gaudi2/BF16 on BERT
- Intel[®] Gaudi[®] 2 ResNet result near H100 submission





Pricing information for range of servers based on Semianalysis GPU Utils for cloud (with pricing) and for servers as of December 2023. Performance source: MLPerf Training 3.1 results <u>https://mlcommons.org/benchmarks/training/</u>

Near-Parity GPT-J Inference Performance vs. H100

Intel[®] Gaudi[®] 2 with FP8 achieved accuracy of

99.9%

- Intel Gaudi 2 throughput:
 -9% (server) and
 -28% (offline) vs H100
- Vs. A100: 2.4x (Server) and 2x (Offline)









Intel[®] Gaudi[®] Software Ecosystem





Software Optimized for Intel[®] Gaudi[®] AI Accelerator Performance and Usability

SynapseAl°

Will evolve to merge with OneAPI, delivering forward compatibility for Gaudi software to Falcon Shores





Access to Thousands of Al Models to Ease Development Process





Developer Resources

Intel® Gaudi® Developer Site

*hab <u>ana</u> D	eveloper	Home	Resources v	Documentation ~	Catalog 🗸	Forum	Explore More 🗸
ome » Resources » Hat	oana Model Performance	Data					
Habana N See the latest perform are currently integrate	AODEL Perfo nance data for Gaudi2 trai d with Habana's Synapse)rmanc ining, Gaudi2 in Al software su	CE Data ference, Gaudi trai iite visit the Haban	ning and Gaudi inferer a catalog.	nce. For informa	tion on mo	dels and container
TRAINING	INFERENCE						
Gaudi2 MLPerf™ 3.0 Training Performance							
These performance	e numbers have been gen	erated with the	latest version of S	ynapseAl and are imp	rovements over	the official	ly submitted numb

Framework Version	Model	# HPU	Precision	Time To Train
PyTorch 2.0.1	MLPerf 3.0 - GPT3	256	bf16	442.5 min
PyTorch 2.0.1	MLPerf 3.0 - BERT	64	bf16	2.2 min
PyTorch 2.0.1	MLPerf 3.0 - BERT	8	bf16	13.3 min
PyTorch 2.0.1	MLPerf 3.0 - ResNet	8	bf16	16.4 min
PyTorch 2.0.1	MLPerf 3.0 - 3D U-Net	8	bf16	21.3 min
TensorFlow 2.12.1	MLPerf 3.0 - ResNet	8	bf16	15.9 min
TensorFlow 2.12.1	MLPerf 3.0 - BERT	8	bf16	14.5 min

Gaudi2 Reference Models Training Performance

how 25 v entries						Search:	
Framework Version	Model	# HPU	Precision	Throughput	Accuracy	Time To Train	
Select Framework	Filter Model						
DeepSpeed 0.9.4	Megatron-DeepSpeed BLOOM 13B	64	bf16	64.37 sent/sec			
DeepSpeed 0.9.4	Megatron-DeepSpeed LLaMA 13B	64	bf16	55.12 sent/sec			
Lightning 2.0.4	Stable Diffusion	64	bf16	6820.62 img/sec			
Lightning 2.0.4	Stable Diffusion	8	bf16	1202.97 img/sec			
Lightning 2.0.4	Stable Diffusion	1	hf16	151.88 img/sec			

Intel Gaudi GitHub



E README.md

Please visit this page for performance information.

This repository is a collection of models that have been ported to run on Habana Gaudi AI accelerator. They are intended as examples, and will be reasonably optimized for performance while still being easy to read.

Computer Vision

Models	Framework	Validated on Gaudi	Validated on Gaudi2
ResNet50, ResNeXt101	PyTorch	Training	Training, Inference
ResNet50 for PyTorch Lightning	PyTorch Lightning	Training	Training
ResNet152	PyTorch	Training	
MobileNetV2	PyTorch	Training	
UNet 2D, Unet3D	PyTorch Lightning	Training, Inference	Training, Inference
SSD	PyTorch	Training	Training
GoogLeNet	PyTorch	Training	
Vision Transformer	PyTorch	Training	
DINO	PyTorch	Training	
VOLOX	PyTorch	Training	

Audio

Models	Framework	Validated on Gaudi	Validated on Gaudi2
Wav2Vec2ForCTC	PyTorch	Inference	Inference
Hubert	PyTorch		Training

Generative Models

Mandala	Common de	Well-band and Could	Malidated and Caudin
Models	Framework	validated on Gaudi	validated on Gaudiz
V-Diffusion	PyTorch	Inference	
Stable Diffusion	PyTorch Lightning	Training, Inference	Training, Inference
Stable Diffusion FineTuning	g PyTorch	Training	Training
Stable Diffusion v1.5	PyTorch	Inference	Inference
Stable Diffusion v2.1	PyTorch	Inference	Inference
	BERT Fretraining and Fin	letuning Pytorch	training, interence
	actor recounting and rin	ryioici	maning, meren

Intel Gaudi Optimum Library on Hugging Face Hub

	Validated Models						
	The following model architecture	es, tasks and device s single-card, multi-	distributions have I card and DeepSpee	een valio d have a	lated for 😑 Optimum Habana: II been validated.		
README.md							
Detimum H	+	hak	oan	a	classification stion answering usge modeling stion answering usge modeling stion answering		
Optimum Habana is occessor (HPU). It prov U settings for differen n try other models an	the interface between the a Transfor ides a set of tools enabling easy model nt downstream tasks. The list of official d tasks with only few changes.	mers and Diffusers lib I loading, training and ly validated models an	raries and Habana's G inference on single- a id tasks is available he	audi and multi- rre. Users	usge modeling stion answering usge modeling usge modeling		
/hat is a Habar /Us offer fast model tr out BERT pre-training amples. If you are not sk at our conceptual g	na Processing Unit (HPU) aining and inference as well as a great and this article benchmarking Habana familiar with HPUs and would like to k juide.	price-performance rat Gaudi2 versus Nvidia now more about them	tio. Check out this bla A100 GPUs for concr n, we recommend you	g post ete take a	generation generation uage modeling		
stall					generation juage modeling		
install the latest stabl	e release of this package: e-strategy eager optimum[habana]			Ø	generation generation		
		LoRA	LoRA	• text	generation		
	StableLM	×	Single card	• text	generation		
	Falcon	×	Single card	• text	generation		
	CodeGen	×	Single card	• text	generation		
	MPT	×	 Single card 	• text	generation		
	TS	~	~	• sum • tran • que	marization slation stion answering		



Train on Gaudi2

Native PyTorch

based model migration

Tools:

- GPU migration tools
- HPU graph
- FP8

References:

Habana Model References

Megatron & DeepSpeed

based model migration

Tools:

- 3D Parallelism
- DP Zero
- LLM pretrain
- FP8

References:

- Habana Megatron-DeepSpeed
- Habana DeepSpeed

Huggingface

based model migration

Tools:

- HF API compatibility
- One step train
- DP Zero
- FP8

References:

Optimum Habana



Inference on Gaudi2

Model / Framework based	Tools	References
Native PyTorch	GPU migration tools DeepSpeed Tensor parallelism FP8 quantization tool HPU graph	Habana Model References
Huggingface	HF API compatibility One step inference DeepSpeed Tensor parallelism FP8 quantization tool HPU graph	Optimum Habana
vLLM / TGI	To be released soon	Habana vLLM / TGI-gaudi
Ray.io	To be released soon	



Easily Get Started with PyTorch Models

import torch import torch.nn as nn import torch.optim as optim import torch.nn.functional as F import torchvision import torchvision.transforms as transforms import os

Import Habana Torch Library
import habana frameworks.torch.core as htcore

neural network model
class SimpleModel(nn.Module):
...

training loop
def train(net,criterion,optimizer,trainloader,device):
...

loss.backward()

API call to trigger execution
 htcore.mark_step()

optimizer.step()

API call to trigger execution
 htcore.mark_step()

def main():

• • •

Target the Gaudi HPU device
device = torch.device("hpu")

Minimal code to start using Gaudi



Migrating Python APIs with GPU dependencies

import torch
import torch.nn as nn

import torch.optim as optim import torch.nn.functional as F

import torchvision

import torchvision.transforms as transforms import os

Import GPU Migration Package: import habana_frameworks.torch.gpu_migration

Import Habana Torch Library import habana frameworks.torch.core as htcore

neural network model
class <u>SimpleModel(nn.Module)</u>:

. . .

training loop
def train(net.criterion.optimizer.trainloader.device):

. . .

loss.backward()

API call to trigger execution

htcore.mark step()
optimizer.step()

API call to trigger execution
htcore.mark step()

def main():

. . .

Target the Gaudi HPU device
 device = torch.device("hpu")

Simplifies replacing Python API calls that have dependencies on GPU libraries with HPU-specific API calls

Specific API calls from following Python libraries are mapped to equivalents in SynapseAI:

• torch.cuda

- torch APIs with GPU related parameters. For example, torch.randn(device="cuda").
- pytorch lightning.
- apex
- pynyml



Getting Started With Hugging Face on Gaudi

from optimum.habana import GaudiConfig, GaudiTrainer, GaudiTrainingArguments

from transformers import BertTokenizer, BertModel

tokenizer = BertTokenizer.from pretrained("bert-base-uncased")

model = BertModel.from pretrained("bert-base-uncased")

gaudi config = GaudiConfig.from pretrained("Habana/bert-base-uncased")

args = GaudiTrainingArguments (

output dir="/tmp/output dir",

use habana=True,

use lazy mode=True,

ĝ.

trainer = GaudiTrainer(

model=model,

gaudi config=gaudi config,

args=args,

tokenizer=tokenizer,

trainer.train()

Optimum-Habana documentation: https://huggingface.co/docs/optimum/habana_index

Supported Model Architectures include:

- NLP: BERT, AlBERT, DistilBERT, RoBERTa, T5, GPT2, BLOOM, LLaMA, MPT, Mixtral,...
- Vision: VIT, SWIN, Stable-diffusion-diffusers
- Audio: Wav2vec2



Train models with 4D parallelism on Gaudi2

By using Megatron-DeepSpeed, train LLaMA on large scale Gaudi2 systems.

- Data Parallelism
- Tensor Parallelism
- Pipeline Parallelism
- Sequence Parallelism



Example: HL_HOSTSFILE=scripts/hostsfile HL_SEQ_PARALLEL=1HL_MICRO_BATCH=1HL_NUM_NODES=32 HL_PP=8 HL_TP=8 HL_DP=4 scripts/run_llamav2.sh



https://github.com/HabanaAI/Model-References/tree/master/PyTorch/nlp/DeepSpeedExamples/Megatron-DeepSpeed

FP8 Training with Intel Gaudi Transformer Engine

1. Import TE and use TE modules in your model, e.g. a te.Linear

import torch
import habana_frameworks.torch.hpex.experimental.transformer_engine as te

Set dimensions. in_features = 768 out_features = 3072 hidden_size = 2048

Initialize model and inputs.
model = te.Linear(in_features, out_features, bias=True)
inp = torch.randn(hidden_size, in_features, device="hpu")

2. Wrap the forward pass of the training with fp8_autocast

from habana_frameworks.torch.hpex.experimental.transformer_engine import recipe

Create an FP8 recipe. Note: All input args are optional.
fp8_recipe = recipe.DelayedScaling(margin=0, interval=1)

Enable autocasting for the forward pass
with te.fp8_autocast(enabled=True, fp8_recipe=fp8_recipe):
 out = model(inp)

intel[°]Ai summit loss = out.sum()
loss.backward()

https://docs.habana.ai/en/latest /PyTorch/PyTorch_FP8_Traini ng/index.html



Customer examples in GenAI & LLM





Intel Gaudi 2 Training New MPT Model from MosaicML: Outperforms H100 for price-performance

3rd party evaluation by Databricks' MosaicML:

https://www.databricks.com/blog/llm-training-and-inference-intel-gaudi2-ai-accelerators

- Delivers 260 TFLOPs-per-Gaudi2: ~55% training performance relative to H100
 - With its cost advantage over H100, Gaudi 2 outperforms H100 for price-performance training MPT



"The MI300x and Gaudi 3 are not yet profiled but are expected to be competitive with H100. "



The MPT-7B model is the first entry in MosaicML's Foundation Series. MPT-7B is a transformer trained from scratch on 1T tokens of text and code. It is open source, available for commercial use, and MosaicML claims it matches the quality of LLaMA-7B. MPT-7B was trained on the MosaicML platform in 9.5 days with no human intervention. See configuration info in backup.

See back-up for test configurations. Results may vary.

MosaicML Evaluation: Gaudi2 near-linear scaling training MPT

"MPT-7B multi-node training performance: As we scale from 8xGaudi2 to 160xGaudi2, we see a near-linear increase in throughput and nearly constant TFLOP/s/device. Note that the global train batch size is held constant at 1920 samples, so these plots demonstrate strong scaling." <u>https://www.databricks.com/blog/Ilm-training-andinference-intel-gaudi2-ai-accelerators</u>





See back-up for test configurations. Results may vary.



MosaicML Evaluation: Gaudi2 Inference of LLama2-70B

https://www.databricks.com/blog/llm-training-and-inference-intel-gaudi2-ai-accelerators

Llama2-70B: Time Per Output Token per User lower is better



📕 8x A100-80GB 📕 8x H100-80GB 📕 8x Gaudi2-96GB

"The TPOT for Gaudi2 is better than H100 under most of the user loads." Llama2-70B: Model Bandwidth Utilization

Higher is better



For decoding latency, the most expensive phase of LLM inference:

With 2450 GB/s of HBM2 memory bandwidth, Gaudi 2 matches H100 decoding latency with 3350 GB/s of HBM3 bandwidth

See back-up for test configurations. Results may vary.

Stability.ai Cluster Built on Intel[®] Gaudi[®] 2 Al Accelerator

"Today, we're excited to announce that we just secured a design win that's a pretty big deal. A large AI supercomputer will be built entirely on Intel's Xeon processors and 4,000 Intel Gaudi2 AI Hardware accelerators. Stability AI is now our anchor customer. Given the magnitude of the Stability AI build out, it will be a top 15 AI supercomputer in the world."

– Pat Gelsinger

intel, Gaudi



Stability.ai: Benchmarking Compute Solutions

Training throughput of 2B MMDiT on Stable Diffusion 3

Device	Attention	# Nodes	# Accelerators (total)	Batch Size per Accelerator	Total Batch Size	Images / sec (100-MA)
Gaudi2	FusedSDPA	2	16	32	512	1,254
Gaudi2	FusedSDPA	2	16	16	256	927
H100-80GB	xFormers	2	16	16	256	595
A100-80GB	xFormers	2	16	16	256	381
						, 0
Device	# Nodes	# Accelerators (total)	Batch Size per Accelerator	Total Batch Size	Images / sec	Images / sec / device
Gaudi2	32	256	16	4096	12,654	49.4
A100-80GB	32	256	16	4096	3,992	15.6



https://stability.ai/news/putting-the-ai-supercomputer-to-work

LLaMA2-70B Training on Gaudi2

Train LLaMA2-70B on 256x, 512x, 1024x Gaudi2





Notices and Disclaimers

For notices, disclaimers, and details about performance claims, visit <u>www.intel.com/PerformanceIndex</u> or scan the QR code:



© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.



Intel® Summer Thank You!

