Training Deep Learning Models on Habana Gaudi®

Milind S. Pandit, Senior Solutions Architect, mpandit@habana.ai
https://www.habana.ai
A little about Habana

• Founded in 2016 to develop purpose-built AI processors
• Launched inference processor in 2018, training processor in 2019
• Acquired by Intel in late-2019
• Fully leveraging Intel’s scale, resources and infrastructure
• Accessing Intel ecosystem and customer partnerships
• Delivering aggressive roadmap optimized for AI data center performance and efficiency
Demand for compute for ML training doubles every 3.4 months

- **Increasing Complexity**
  - 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

Need for dedicated AI processors to address the compute, memory and communication challenges
Today’s Cost to Train: Biggest Barrier to AI Implementation

“Cost is the most significant challenge to implementing AI/ML solutions.”

56% of AI/ML customers

Industry Challenge:
How to give more customers access to more AI?

Source: IDC Semiannual Artificial Intelligence Tracker (2020H1, published Jan 2021)
GAUDI®

Purpose-built for AI training efficiency, usability and scale
DL1 Model Training Cost Savings

Cost savings based on Amazon EC2 On-Demand pricing for P3, P4d and DL1 instances respectively. Performance data was collected and measured using the following resources:

Habana BERT-Large Model: https://github.com/HabanaAI/Model-References/tree/master/TensorFlow/nlp/bert
Habana ResNet50 Model: https://github.com/HabanaAI/Model-References/tree/master/TensorFlow/computer_vision/Resnets/resnet_keras
Habana SynapseAI Container: https://vault.habana.ai/ui/repos/tree/General/gaudi-docker/1.4.0/ubuntu20.04/habanalabs/tensorflow-installer-tf-cpu-2.8.0
Habana Gaudi Performance: https://developer.habana.ai/resources/habana-training-models/


Results published for DGX A100-40G and DGX V100-32G. Results May Vary.
The Habana® Gaudi® AI Training Processor

Designed to optimize AI performance, delivering higher AI efficiency than traditional CPUs and GPUs

Heterogeneous compute architecture enables high-efficiency on large AI workloads
- GEMM engine (MME) excels at matrix multiplication
- While TPC runs non-linear and element wise ops

Software-managed memory architecture
- 32 GB of HBM2 memory

Integrates ten 100Gb Ethernet RoCE ports
- Scaling capacity
- Flexibility based on industry standard
- Cost-efficiency with integrated NIC
Scaling within a Gaudi Server

- 8 Gaudi OCP OAM cards
- 24 x 100GbE RDMA RoCE for scale-out
- Non-blocking, all-2-all internal interconnect across Gaudi AI processors
- Separate PCIe ports for external Host CPU traffic

Example of Integrated Server with eight Gaudi AI processors, two Xeon CPU and multiple Ethernet Interfaces
Software Suite Detail

TensorFlow and PyTorch integrated with SynapseAI and optimized for Gaudi. Developers enjoy the same abstraction that they are accustomed to today.

Graph compiler generates optimized binary code that implements model topology on Gaudi.

LLVM-based TPC-C compiler, simulator and debugger facilitate development of custom TPC kernels TensorFlow and PyTorch.

Rich TPC kernel library, with wide variety of operations (for example, elementwise, non-linear, non-GEMM operators).

Implements efficient scale-up between Gaudi cards within a single node and scale-out across nodes for distributed training. HCL uses Gaudi integrated NICs for both scale-up and scale-out. HCCL allows users to enable Gaudi integrated NIC for scale-up and host NIC for scale-out, using NCCL-compatible API.
Multiple teams across Mobileye have chosen to use Gaudi-accelerated training machines, either on Amazon EC2 DL1 instances or on-prem; Those teams consistently see significant cost-savings relative to existing GPU-based instances across model types, enabling them to achieve much better Time-To-Market for existing models or training much larger and complex models...We’re excited to see Gaudi2’s leap in performance”

Gaby Hayon, EVP R&D, Mobileye
Mobileye journey towards scaling Amazon EKS to thousands of nodes leveraging Intel® Xeon® Scalable Processors and Habana’s Gaudi AI accelerators

Authors: Diego Bailon Humpert, AWS EMEA and Global Automotive GTM Lead & David Peer, Mobileye AI Engineering DevOps specialist & team leader.

Mobileye is a company that develops autonomous driving technologies and advanced driver-assistance systems (ADAS) including cameras, computer chips, and software.
Accelerating Medical Imaging Applications

**Objective**
Demonstrate Gaudi DL1 AI processor cost-efficiency (price-performance ratio) for training deep learning models to detect novel coronavirus pneumonia in frontal chest X-ray images.

**Models**
- Pretraining: CheXNet, to detect and localize multiple kinds of diseases from chest X-ray images.
- Finetuning: COVID-CXNet, to detect novel coronavirus pneumonia in frontal chest X-ray images
  - Transfer learning of CheXNet with a focus on Grad-CAM visualizations.

**Datasets**
- 3200 normal images from NIH CXR dataset excluding age < 18 images based on paper
- 845 COVID-19 images from dataset used in the paper excluding age < 18 and early stage images.

Source: Leidos
## DL1 Cost Savings

### CheXNet-Keras
- **Dataset:** ChestXray-NIHCC
- **Batch size:** 32
- **Precision:** FP32
- **Device count:** 8

<table>
<thead>
<tr>
<th>Instance</th>
<th>On-Demand hourly rate of EC2 instance [$/Hour]</th>
<th>Time per epoch [Seconds]</th>
<th>Cost per epoch [$]</th>
<th>DL1 Cost Savings to EC2 Customers [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x V100-32 GB* (p3dn.24xlarge)</td>
<td>$31.21</td>
<td>4.6</td>
<td>$143.57</td>
<td>59%</td>
</tr>
<tr>
<td>8x Gaudi DL1.24xlarge**</td>
<td>$13.11</td>
<td>4.47</td>
<td>$58.56</td>
<td></td>
</tr>
</tbody>
</table>

### COVID-CXNet
- **Dataset:** COVID-CXNet
- **Batch size:** 16
- **Precision:** BF16
- **Device Count:** 1

<table>
<thead>
<tr>
<th>Instance</th>
<th>On-Demand hourly rate of EC2 instance [$/Hour]</th>
<th>Time per epoch [Seconds]</th>
<th>Cost per epoch [$]</th>
<th>DL1 Cost Savings to EC2 Customers [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x V100-32 GB* (p3dn.24xlarge)</td>
<td>$31.21</td>
<td>718</td>
<td>$6.22</td>
<td>67%</td>
</tr>
<tr>
<td>8x Gaudi DL1.24xlarge**</td>
<td>$13.11</td>
<td>565</td>
<td>$2.06</td>
<td></td>
</tr>
</tbody>
</table>

Results may vary.
Pricing is published by Amazon: [https://aws.amazon.com/ec2/pricing/on-demand/](https://aws.amazon.com/ec2/pricing/on-demand/)
Pricing data accessed: May 2022

Source: Leidos
Summary

- Using Amazon EC2 DL1 instances for Chest X-Ray COVID Detection model pretraining and finetuning resulted in 60%+ savings in cost of training

- Successfully trained deep learning models on EC2 DL1 platform with minimal code changes

- Excellent support and documentation available on Habana Developer Site https://developer.habana.ai and GitHub with reference models

Source: Leidos
Leidos customer using NLP-based deep learning solution to facilitate medical benefit application processing

**Objective:** Demonstrate price/performance of Gaudi based EC2 DL1.24xlarge instance versus GPU based G4DN.12xlarge EC2 instance used by customer

- TensorFlow DistilBERT Model finetuned for a multi-labeling classification task
- Trained with 737k labeled examples and tested against 184k test examples

<table>
<thead>
<tr>
<th>Attribute</th>
<th>G4DN.12xlarge (GPU based)</th>
<th>DL1.24xlarge (Gaudi Based)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>192 GB</td>
<td>768 GB</td>
</tr>
<tr>
<td>Accelerator</td>
<td>4 x Tesla T4</td>
<td>8 x Gaudi</td>
</tr>
<tr>
<td>Accelerator Type</td>
<td>GPU</td>
<td>HPU</td>
</tr>
<tr>
<td>On Demand Cost (per hour)</td>
<td>$3.91</td>
<td>$13.11</td>
</tr>
</tbody>
</table>

Source: Leidos
Results

Cost Performance

- Compared to GPU, training took only 45% of the time with 1x Gaudi processor and only 10% of the time when all 8 Gaudi processors were used
- Although the DL1 instance costs more per hour, due to shorter training time, the total cost for model training ended up being only 1/3 of the baseline, i.e., 66% cost savings.

<table>
<thead>
<tr>
<th>Device</th>
<th>Training Time</th>
<th>Training Cost</th>
<th>DL1 cost savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPU (g4dn)</td>
<td>10 hrs</td>
<td>$39.12</td>
<td></td>
</tr>
<tr>
<td>Gaudi x1 (dl1)</td>
<td>4.5 hrs</td>
<td>$59.00</td>
<td></td>
</tr>
<tr>
<td>Gaudi x8 (dl1)</td>
<td>1 hr</td>
<td>$13.11</td>
<td>66%</td>
</tr>
</tbody>
</table>

User Experience

- Performing single card training is quite intuitive and simple
- Distributed training is simple when using Horovod and OpenMPI

Results may vary.
Pricing is published by Amazon: [https://aws.amazon.com/ec2/pricing/on-demand/](https://aws.amazon.com/ec2/pricing/on-demand/)
Pricing data accessed: May 2022

Source: Leidos
Summary

• Gaudi performs significantly better than NVIDIA Tesla T4 when looking at the time and cost metrics
• DL1 is worthy of strong consideration as part of cloud-forward strategy, especially when an organization anticipates using deep learning models.
• For Leidos’ customers who are interested in continuing training, low cost is a very attractive feature

Areas for Future Work:
• With lower cost to train with Gaudi, one can potentially train/update more complex and accurate models
• Pre-training or continue to pre-train domain specific model, which is a more computing expensive task. This is critical for domain adaptation, which impacts a broad range of NLP tasks and is relevant to lot of our clients.
> 60% cost savings with DL1 vs. GPU instances

“Given Leidos and its customers’ need for quick, easy, and cost-effective training for deep learning models, we are excited to have begun this journey with Intel and AWS to use Amazon EC2 DL1 instances based on Habana Gaudi AI processors.”

Chetan Paul, CTO Health and Human Services at Leidos
Habana AI Powers SDSC’s Voyager Research Program

336 Gaudi Training accelerators with native RoCE scaling and 16 Goya Inference processors

- In service since Fall of 2021
- Funded by $5M grant from National Science Foundation
  - Matching funds targeting community support and operation
- AI research conducted across range of science and engineering domains
  - Astronomy, climate sciences, chemistry, particle physics,
- Announced by SDSC in July 2020, more information [here](#).
Combining Fire Science With AI For Wildfire Mitigation

DL algorithms of satellite images determine land covers across geographies in the context of wildfire management

“With innovative solutions optimized for deep learning operations and AI workloads, Habana accelerators are excellent choices to power Voyager’s forthcoming AI research”

Amit Majumdar, Director of Data Enabled Scientific Computing Division, SDSC
First-generation Gaudi

IN THE CLOUD
- AWS EC2 DL1 Instances
- Leading AWS AI training efficiency

ON PREMISES
- Supermicro X12 Gaudi Server
- DDN AI 400X2 storage solution

MLOps SOFTWARE

aws

ddn
cnvrng.io
import tensorflow as tf

from TensorFlow.common.library_loader import load_habana_module
load_habana_module()

(x_train, y_train), (x_test, y_test) = tf.keras.datasets.mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([tf.keras.layers.Flatten(input_shape=(28, 28)),
                                    tf.keras.layers.Dense(10),
                                ])  

loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
optimizer = tf.keras.optimizers.SGD(learning_rate=0.01)

model.compile(optimizer=optimizer, loss=loss, metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5, batch_size=128)
model.evaluate(x_test, y_test)

Load the Habana® libraries needed to use Gaudi aka HPU device

Once loaded, the HPU device is registered in TensorFlow

When an Op is available for both CPU and HPU, Op is assigned to the HPU

When an Op is not supported on HPU, it runs on the CPU
import pytorch_lightning as pl
from pytorch_lightning.plugins import HPUPrecisionPlugin

# mixed precision distributed training with 8 Gaudios
trainer = pl.Trainer(accelerator="hpu", devices=<n>, precision=16)

All you need is to provide `accelerator="hpu"` parameter to the Trainer class

Select the number of Gaudi devices, n=1..8

For mixed precision training, import `HPUPrecisionPlugin` and set “`precision=16`”

Getting Started With Huggingface On Gaudi

```python
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,
)
trainer = GaudiTrainer(
    model=model,
    gaudi_config=gaudi_config,
    args=args,
    tokenizer=tokenizer,
)
trainer.train()
```

- Uses Optimum Habana library
- Model instantiated the same way as in the Transformers library
- Only difference is to load Gaudi configuration and provide to the Gaudi trainer
Welcome to Habana’s developer site.

Here you will find the content, guidance, tools and support needed to easily and flexibly build new or migrate existing AI models and optimize their performance to meet your AI requirements. You can also access the latest Gaudi software to build or update your infrastructure.

GAUDI®

Today we announced the Gaudi2 processor for training deep learning workloads

LEARN ABOUT GAUDI2 

SynapseAI®

The Habana® Labs team is happy to announce the release of SynapseAI® version 1.5.0.

READ THE BLOG 

Live Webinar: Accelerate Transformer Model Training with Hugging Face and Habana Labs

REGISTER TO WEBINAR 

Get Started

Get access to Habana’s programmable Tensor Processor Core and SynapseAI® software stack with support for TensorFlow and PyTorch frameworks, along with numerous other libraries and tools that enable portable, high-performance AI applications. Go to the Habana Developer Platform today to start your AI journey with Habana.
Welcome to Habana® Gaudi® v1.5 Documentation

Find detailed documentation to learn how to use the Habana Gaudi solutions - first-generation Gaudi and Gaudi2. This will cover the details on how to migrate models to Habana, code samples, diagrams, best practices for debug and optimization, API references, and more.

Getting Started
Start using Habana Gaudi Processors
Click here to get started

Tutorials
Tutorials to show basic examples of how to run on TensorFlow and PyTorch
Click Here

Model Catalog
Start with TensorFlow and PyTorch models already running on Gaudi
Click Here

User Forum
Post questions and get help in the User Forum
Click Here
Habana Developer Software---vault.habana.ai

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Date</th>
<th>Version</th>
<th>License</th>
<th>Versions</th>
<th>Downloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>habanalabs-aeon</td>
<td>03-21-2022 05:42:32 -0700</td>
<td>0:0.15.1-37.amzn2</td>
<td>Unknown</td>
<td>12</td>
<td>2k</td>
</tr>
<tr>
<td>habanalabs-aeon</td>
<td>03-21-2022 05:48:18 -0700</td>
<td>1.3.0-499</td>
<td>Unknown</td>
<td>13</td>
<td>2k</td>
</tr>
<tr>
<td>habanalabs-container-runtime</td>
<td>06-15-2022 21:30:55 -0700</td>
<td>1.5.0-610</td>
<td>Unknown</td>
<td>17</td>
<td>6k</td>
</tr>
</tbody>
</table>
Habana GitHub Repositories---github.com/HabanaAI
# Gaudi Reference Models---August 2022

## TensorFlow

- ResNet50 Keras
- ResNeXt101
- SSD
- Mask R-CNN
- DenseNet
- UNet 2D
- UNet 3D
- UNet Industrial
- CycleGAN
- EfficientDet
- RetinaNet
- SegNet
- Vision Transformer
- MobileNet V2

## PyTorch

- BERT
- DistilBERT
- ALBERT
- Transformer
- T5 Base
- Electra

### TensorFlow Models

- ResNet50
- ResNet50, ResNeXt101
- ResNet152
- MobileNet V2
- UNet 2D
- UNet 3D
- SSD
- GoogleNet
- Vision Transformer
- Swin Transformer
- DINO

### PyTorch Models

- BERT Pretraining
- BERT Finetuning
- DeepSpeed BERT-1.5B
- RoBERTa
- ALBERT
- DistilBERT
- Electra
- Transformer
- BART
- GPT2
Welcome to Habana’s Developer Forum

We are creating a new community for developers and data scientists, IT and systems administrators, who wish to post issues and solutions, and share in creating the next new thing in AI, leveraging a new class of AI accelerators from Habana. Be among the first to join this new way to AI.

<table>
<thead>
<tr>
<th>Category</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Announcements</td>
<td>7</td>
</tr>
<tr>
<td>Training</td>
<td>49</td>
</tr>
<tr>
<td>Inference</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Latest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome to Habana’s developers forum</td>
</tr>
<tr>
<td>Gaudi Torch Cunmax</td>
</tr>
<tr>
<td>Error with convolution layers</td>
</tr>
</tbody>
</table>
Leadership Performance
~2x better throughput vs A100
for popular vision and language models
Gaudi2: Second-generation Training & Inference

Node
- 16nm
- 7nm

Compute
- 8 TPCs
- New
- 24 TPCs
- Media decode & processing
- FP8

Memory
- 1 TB/s
- 32GB HBM2
- 24 MB

Networking
- 2.45 TB/s
- 96 GB HBM2e
- 48 MB SRAM
- 10 x 100
- 24 x 100 GbE

TDP
- 350W
- 600W
Gaudi2 outperformed Nvidia A100 MLPerf submissions on both ResNet and BERT

...and First-gen Gaudi achieved near-ideal linear scale on 128- and 256-accelerators

**Gaudi2 Time-to-Train (TTT) improved by 3 to 4.7x compared to First-gen Gaudi**
MLPerf Press Coverage

Intel Habana overtakes Nvidia in latest MLPerf results

Since Habana’s Last MLPerf submission...

Much has happened at Habana since our last MLPerf submission in November 2021. We launched Gaudi2, our second-generation deep learning training processor, and Greco™, our second-generation inference processor. We expanded our software functionality with SynapsaAI, which supports the latest versions of PyTorch, PyTorch Lightning, Tensorflow and OpenMPI. Our operator and model coverage is continually expanding and we have added many popular reference models to the HabanaAI GitHub repository with latest examples, including Vision transformers and DeepSpeed BERT. Habana also joined forces with HuggingFace to integrate SynapseAI into the Optimum open-source library, as well as LIGHTING to integrate SynapsaAI with PyTorch Lightning. We have also integrated SynapsaAI with the c-ai-norm MLPerf platform.

All the above is aimed at enabling data scientists and machine learning engineers to accelerate model training and inference not only on Habana processors but also on a wide range of other hardware architectures.

NVIDIA Loses The AI Performance Crown, At Least For Now

Nvidia has been king of AI training workloads due to their flexible, easy to program, powerful hardware. This may be changing as AI is very dynamic and various different AI workloads are bifurcating. Training isn’t a monolithic entity, and therefore the hardware and software solution best suited for your workload may not be the same in that of another workload. Combined with the rapid pace of model evolution, some AI training hardware is starting to find a niche.
Deep Learning Inference Efficiency
Greco: Second-Generation Inference for Deep Learning

Node: 16nm to 7nm

Memory:
- 40GB/s DDR4
- 16GB LPDDR5
- 50MB
- 128MB on chip SRAM

Compute: BF16, FP16, INT4
- Media decode and processing

Form Factor: Dual-slot PCIe > Single-slot HHHL

TDP:
- 200w
- 75w
TensorFlow integration with SynapseAI

SynapseAI receives a computational graph of the model from the framework.

It identifies subgraphs (blue nodes) that can be accelerated by Gaudi.

The rest of the graph runs on CPU (yellow node).

The original graph is modified to replace the Gaudi subgraphs with encapsulated nodes (blue).

The framework runtime executes the modified graph.

For each encapsulated node, SynapseAI generates optimized binary code that runs on Gaudi.
Software Installation and Deployment

Setup_and_Install repository on Habana GitHub provides instructions on how to setup your environment with the SynapseAI software stack.

**SynapseAI Orchestration**
(Kubernetes Gaudi plugin, Kubeflow mpi-operator)

**SynapseAI TensorFlow Container Image**
(TensorFlow frontend, horovod, open-mpi)

**SynapseAI Base Installation Image**
(OS, Gaudi linux kernel driver, user mode driver, graph compiler, HCL/HCCL & embedded tools)

**Gaudi Server**

Gaudi-optimized Docker container images with all necessary dependences*

Official releases publicly available on Habana Vault

<table>
<thead>
<tr>
<th>Orchestration</th>
<th>Kubernetes (1.19)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frameworks</td>
<td>TensorFlow2 and PyTorch</td>
</tr>
<tr>
<td>Operating Systems</td>
<td>Ubuntu 18.04 and 20.04</td>
</tr>
<tr>
<td>Container Runtimes</td>
<td>Docker (Docker CE version 18.09)</td>
</tr>
<tr>
<td>Distributed Training Schemes</td>
<td>TensorFlow with Horovod and tf.distribute</td>
</tr>
<tr>
<td></td>
<td>PyTorch distributed (native)</td>
</tr>
</tbody>
</table>

* Habana GitHub will have repository with Dockerfiles to “build your own” Docker images
DL1 Vision Model Training Performance


Habana SynapseAI Container: [https://vault.habana.ai/ui/repos/tree/General/gaudi-docker/1.2.0/ubuntu20.04/habanalabs/tensorflow-installer-tf-cpu-2.7.0](https://vault.habana.ai/ui/repos/tree/General/gaudi-docker/1.2.0/ubuntu20.04/habanalabs/tensorflow-installer-tf-cpu-2.7.0)


A100 / V100 Performance Source: [https://ngc.nvidia.com/catalog/resources/nvidia:resnet_50_v1_5_for_tensorflow/performance](https://ngc.nvidia.com/catalog/resources/nvidia:resnet_50_v1_5_for_tensorflow/performance), results published for DGX A100-40G and DGX V100-32G
Vision Model Training Scalability

ResNet 50 Training Throughput

Number of Gaudi Processors

images/second

Habana Gaudi  Linear
**DL1 NLP Model Training Performance**

Habana BERT-Large Model: [https://github.com/HabanaAI/Model-References/tree/master/TensorFlow/nlp/bert](https://github.com/HabanaAI/Model-References/tree/master/TensorFlow/nlp/bert)

Habana SynapseAI Container: [https://vault.habana.ai/ui/repos/tree/General/gaudi-docker/1.2.0/ubuntu20.04/habanalabs/tensorflow-installer-tf-cpu-2.7.0](https://vault.habana.ai/ui/repos/tree/General/gaudi-docker/1.2.0/ubuntu20.04/habanalabs/tensorflow-installer-tf-cpu-2.7.0)


A100 / V100 Performance Sources: [https://ngc.nvidia.com/catalog/resources/nvidia-bert_for_tensorflow/performance](https://ngc.nvidia.com/catalog/resources/nvidia-bert_for_tensorflow/performance), results published for DGX A100-40G and DGX V100-32G
NLP Model Training Scalability

BERT Large Pretraining Throughput
(Phase 1 & Phase 2)

BERT Large Finetuning Throughput
(Phase 1)

BERT Large Pretraining Throughput
(Phase 2)