Intel® Gaudi®2 AI Accelerator for Deep Learning Training and Inference

Karthikeyan Vaidyanathan
November 2023
Notices and Disclaimers

Performance varies by use, configuration and other factors. Learn more at https://habana.ai/habana-claims-validation/

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

© 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.
Agenda

- Gaudi2 programming model and recent MLPERF results
- Experience at scale
  - Use case1: SWIFT congestion control
  - Use case2: Packet/message spraying
Train and deploy large scale GenAI and LLMs

Gaudi2 Clusters and systems for models from billions to trillions of parameters

Fine tune and run thousands of domain specialized models with targeted curated data sets from the data center and the factory floor to devices

Intel Xeon for fine tuning and inferencing models up to tens of billions of parameters
Intel® Gaudi® Accelerator Roadmap

Available Now

**GAUDI®**
Native RoCE Scaleup & out
Available via:
- HLS-1 Server (x8)
- SMC Server (x8)
- SDSC
- Public Cloud AWS: EC2

Available Now

**GAUDI®2**
(7nm)
Native RoCE Scaleup & out
Available via:
- HLS-Gaudi2 Server (x8)
- SMC Server (x8)
- Aivres/IEI Server (x8)
- Intel Dev Cloud

In Development

**GAUDI®3**
(5nm)
Native RoCE Scaleup & out

In Development

Next Generation AI Accelerator:
Falcon Shores 1
Native RoCE Scaleup & out

2024

2025
Intel delivers increasingly competitive Training Performance

- One of only three accelerators submitting GPT-3 results: Intel, Nvidia, Google

- Xeon continues to be the only CPU to submit training results on the MLPerf Benchmark.
Intel® Gaudi®2 Accelerator Performance Doubled with FP8

- Intel Gaudi team projected to customers +90% performance gain with FP8
- Delivered more than promised: 103% on GPT-3 industry benchmark

MLPerf Training 3.1 GPT-3 Benchmark
- Lower is better -

For complete results information and configurations, see MLCommons publication: https://mlcommons.org/en/inference-datacenter-31/

See backup for workloads and configurations. Results may vary.
Intel® Gaudi®2 performance advances strengthen competitive price-performance vs. H100

- Gaudi2 performance on ResNet near that of H100.
- H100 with FP8 outperformed Gaudi2 with BF16 on BERT.
- Vs.TPU, Gaudi2 delivered 3x performance on GPT-3.
- Given its significantly lower server cost vs. H100 server cost, Intel Gaudi2 delivers price-performance advantage vs. H100 across models.

For complete results information and configurations, see MLCommons publication: [https://mlcommons.org/en/inference-datacenter-31/](https://mlcommons.org/en/inference-datacenter-31/)

<table>
<thead>
<tr>
<th>Workload</th>
<th>Nvidia H100 (Nov)</th>
<th>Intel Gaudi2 (Nov)</th>
<th>Intel Gaudi2 (May)</th>
<th>Google TPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-3</td>
<td>0.5x</td>
<td>1</td>
<td>2.03x</td>
<td>3.09x</td>
</tr>
<tr>
<td>Stable Diffusion</td>
<td>0.5x</td>
<td>1</td>
<td>0.85x</td>
<td>0.41x</td>
</tr>
<tr>
<td>ResNet</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>GPT-3</th>
<th>Stable Diffusion</th>
<th>ResNet</th>
<th>BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPUs/G2s/TPUs</td>
<td>512x/384x/4096</td>
<td>64x/8x</td>
<td>8x/8x</td>
<td>8x/8x</td>
</tr>
<tr>
<td>FP8/H100/F16</td>
<td>G2s/H100/F16</td>
<td>G2/BF16</td>
<td>G2/BF16</td>
<td>G2/BF16</td>
</tr>
</tbody>
</table>

Lower is better

See backup for workloads and configurations. Results may vary.
Outstanding Intel® Gaudi®2 AI Accelerator performance on MLPerf v3.1 Inference Benchmark

Intel Gaudi2 Accelerator with FP8: near-parity performance on GPT-J (Server) with H100

- Gaudi 2 inference performance on GPT-J: -9% (Server) and -28% (Offline) vs H100
- Gaudi 2 outperformed A100 by 2.4x (Server) and 2x (Offline)
- Gaudi 2 employed FP8 and reached 99.9% accuracy

For complete results information and configurations, see MLCommons publication: https://mlcommons.org/en/inference-datacenter-31/

See backup for workloads and configurations. Results may vary.
SynapseAI Software: Optimized for Intel® Gaudi® Performance and Ease of Use

- Shared software suite for training and inference
- Start running on Intel Gaudi accelerators with minimal code changes
- Integrated with PyTorch and TensorFlow
- Rich library of performance-optimized kernels
- Advanced users can write their custom kernels
- Docker container images and Kubernetes orchestration
- Habana Developer Site & HabanaAI GitHub
- Habana Developer Forum
Distributed Pytorch

- Trainer (Main Process)
- Framework (Main Process)
- Bridge (spawned Process 1)
  - Hpu_mid_layer_backend (4/2 threads)
  - Events/device
  - Processgroup_hccl (1 thread)
- HPU (spawned Process 1)
  - Synapse
  - HCCL
- 1 HLS
  - Gaudi
  - ...
- Bridge (spawned Process 2)
  - Hpu_mid_layer_backend (4/2 threads)
  - Events/device
  - Processgroup_hccl (1 thread)
- HPU (spawned Process 2)
  - Synapse
  - HCCL
HCCL API

// Communicator creation
hcclGetUniqueId(hcclUniqueld* uniqueld);
hcclCommInitRank(hcclComm_t* comm, int nranks, hcclUniqueld commId, int rank);

// Communicator destruction
hcclCommDestroy(hcclComm_t comm);

// Collectives communication
hcclReduceScatter(void* sbuff, void* rbuff, size_t recvcount, hcclDataType_t datatype, hcclRedOp_t op, hcclComm_t comm, synStreamHandle stream_handle);
hcclAllReduce(void* sbuff, void* rbuff, size_t count, hcclDataType_t datatype, hcclRedOp_t op, hcclComm_t comm, synStreamHandle stream_handle);
hcclBroadcast(void* sbuff, void* rbuff, size_t count, hcclDataType_t datatype, int root, hcclComm_t comm, synStreamHandle stream_handle);
hcclAllGather(void* sbuff, void* rbuff, size_t sendcount, hcclDataType_t datatype, hcclComm_t comm, synStreamHandle stream_handle);
hcclReduce(void* sbuff, void* rbuff, size_t count, hcclDataType_t datatype, hcclRedOp_t op, int root, hcclComm_t comm, synStreamHandle stream_handle);
hcclAlltoAll(...);

// Point-to-point communication
hcclSend(void* sbuff, size_t count, hcclDataType_t datatype, int peer, hcclComm_t comm, synStreamHandle stream);
hcclRecv(void* rbuff, size_t count, hcclDataType_t datatype, int peer, hcclComm_t comm, synStreamHandle stream);

// Aggregation/Composition
hcclGroupStart();
hcclGroupEnd();
Incast Congestion

- PFC is great but does not work for multi-tenant and multi-level switches
- When packet drops occur, utilization is poor
SWIFT congestion control for Habana Gaudi2

<table>
<thead>
<tr>
<th>7:1 congestion, No PFC</th>
<th>#packets dropped</th>
<th>Bandwidth utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>276787</td>
<td>10-50%</td>
</tr>
<tr>
<td>SWIFT</td>
<td>~1</td>
<td>~98%</td>
</tr>
</tbody>
</table>

for 4KB/8KB MTU (targetdelay=20usecs, ai=2, beta=0.5,min_cwnd=2, max_cwnd=32)

50% equal share on 2 flows
33% equal share on 3 flows

SWIFT: https://dl.acm.org/doi/pdf/10.1145/3387514.3406591
Packet collision at large-scale

Collision due to mapping to same output port \( \rightarrow \) leads to performance degradation

**Spine Switches**
- ECMP hashing
  - src ip
  - dst ip
  - src port
  - dst port
  - protocol

**Leaf Switches**

**Gaudi2 Servers**

G0  G4  G16  G20
Only certain output ports have traffic and rest are idle and unutilized.
Solution: Packet spraying

ECMP hashing
- src ip
- dst ip
- src port
- dst port
- protocol

Spine Switches

Leaf Switches

Gaudi2 Servers
## Packet spraying solution

<table>
<thead>
<tr>
<th>HCCL collectives BW</th>
<th>With collisions</th>
<th>Packet spraying</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>All2All</td>
<td>22 GB/s</td>
<td>64 GB/s</td>
<td>65 GB/s</td>
</tr>
<tr>
<td>Allgather</td>
<td>183 GB/s</td>
<td>272 GB/s</td>
<td>272 GB/s</td>
</tr>
</tbody>
</table>
Almost all ports are utilized
Developer Resources

Gaudi Developer Site: developer.Habana.ai

Habana GitHub

Habana Optimum Library on Hugging Face Hub
Summary

- Intel Gaudi2 continues to be the only viable alternative to NVIDIA’s H100 for GenAI/LLM compute, with a significant price-performance advantage.
- 4th Gen Intel Xeon processors help customers train small- to mid-sized deep learning models, as well as fine tuning and transfer learning.
- Intel is well positioned to address every phase of the AI continuum across AI workloads, from large to small models—giving customers choice.
Thank you