Accelerating Distributed Deep Learning using HCCL: Habana Collective Communication Library

Karthikeyan Vaidyanathan

Exacomm 2023 Workshop

habana
An Intel Company

Notices and Disclaimers

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies may require enabled hardware, software or service activation. Your costs and results may vary. Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction. No product or component can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Some results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling and are provided to you for informational purposes.

Intel Advanced Vector Extensions (Intel AVX) provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

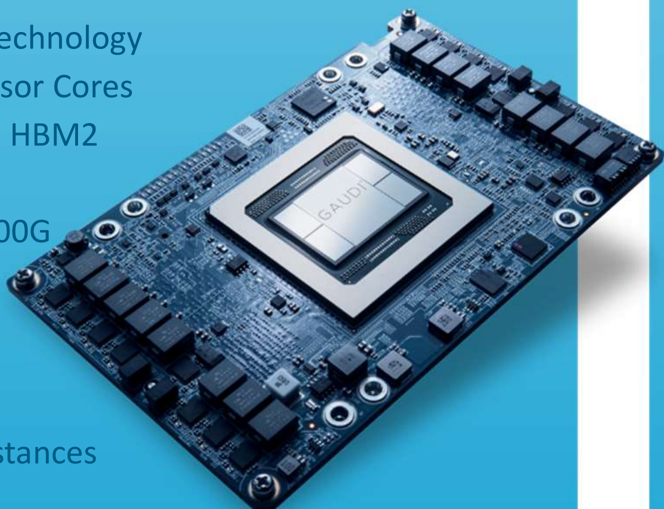
Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Habana Deep Learning Solutions

GAUDI®

High-performance, high-efficiency (price/performance)

- 16nm process technology
- 8 Tensor Processor Cores
- 32 GB on-board HBM2
- 24 SRAM
- 10 integrated 100G Ethernet ports



In the cloud:

- Amazon EC2 DL1 Instances

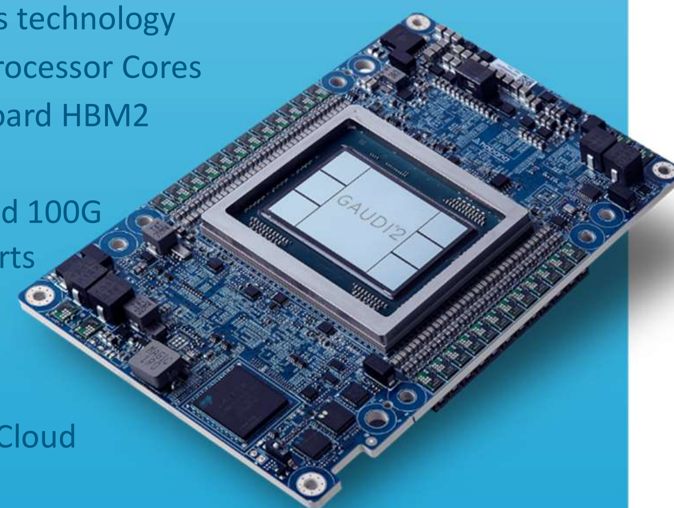
On-premises:

- Supermicro X12 Gaudi Server with 3rd Gen Xeon CPU

GAUDI®2

Higher performance, high-efficiency; optimized speed, memory, scalability for large scale models

- 7nm process technology
- 24 Tensor Processor Cores
- 96 GB on-board HBM2
- 48 SRAM
- 24 integrated 100G Ethernet ports



In the cloud:

- Intel Developer Cloud

On-premises:

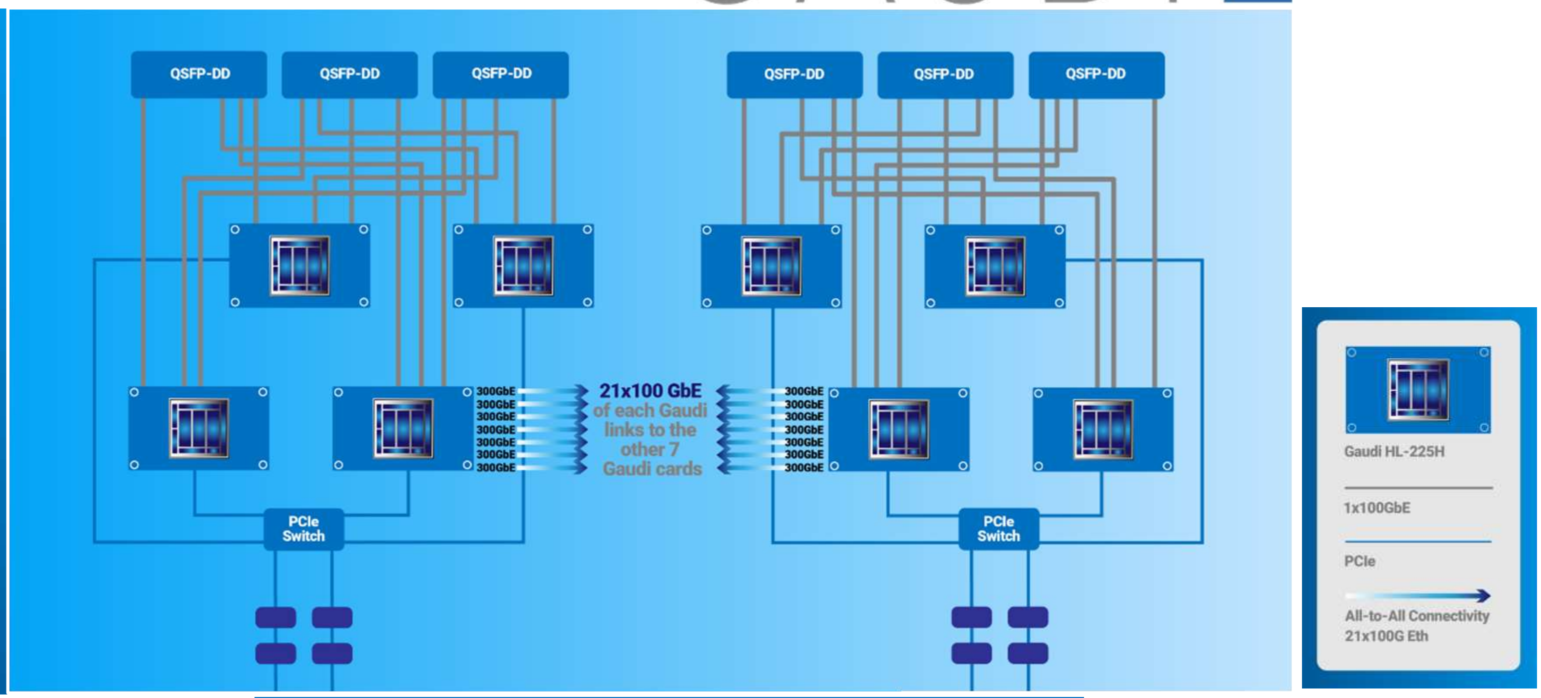
- Supermicro Gaudi2 Server with 3rd Gen Xeon CPU

Gaudi2 Server Delivers Flexible and Efficient Scalability

GAUDI®2

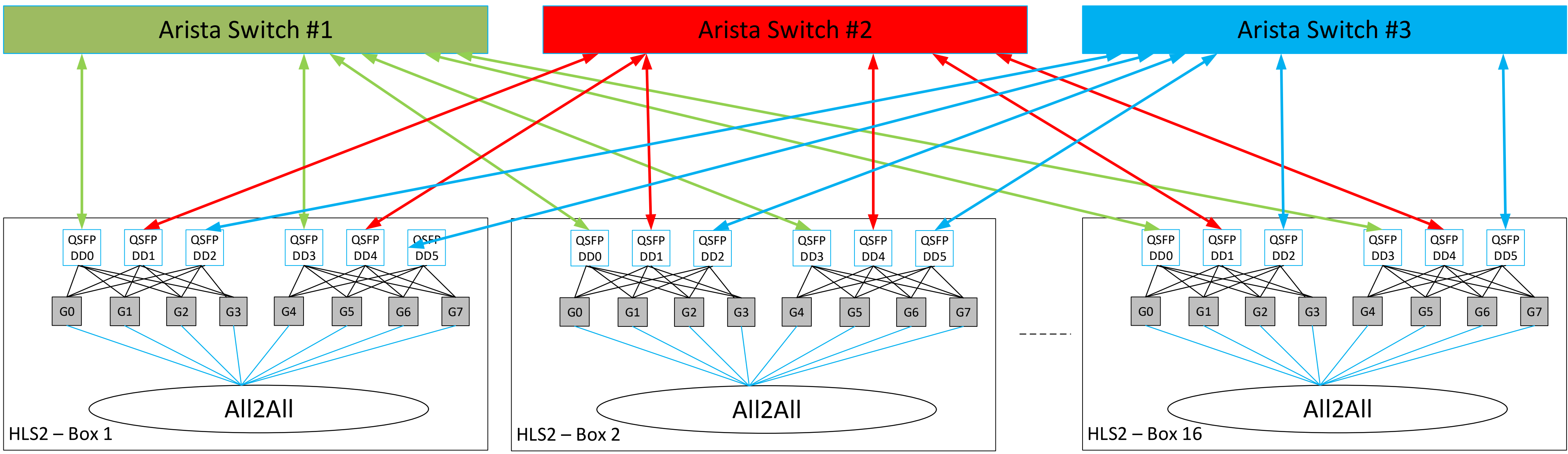
HLS-Gaudi2 Reference Server featuring…

- 8 Gaudi2 mezzanine cards
- 24x 100 GbE ports per card
  - 21 for all-to-all connectivity to other 7 Gaudis within the server
  - 3 to scale out
  - Through 6 QSFP-DD ports
- Dual-socket Host CPU: 3rd Gen Xeon Scalable Processors

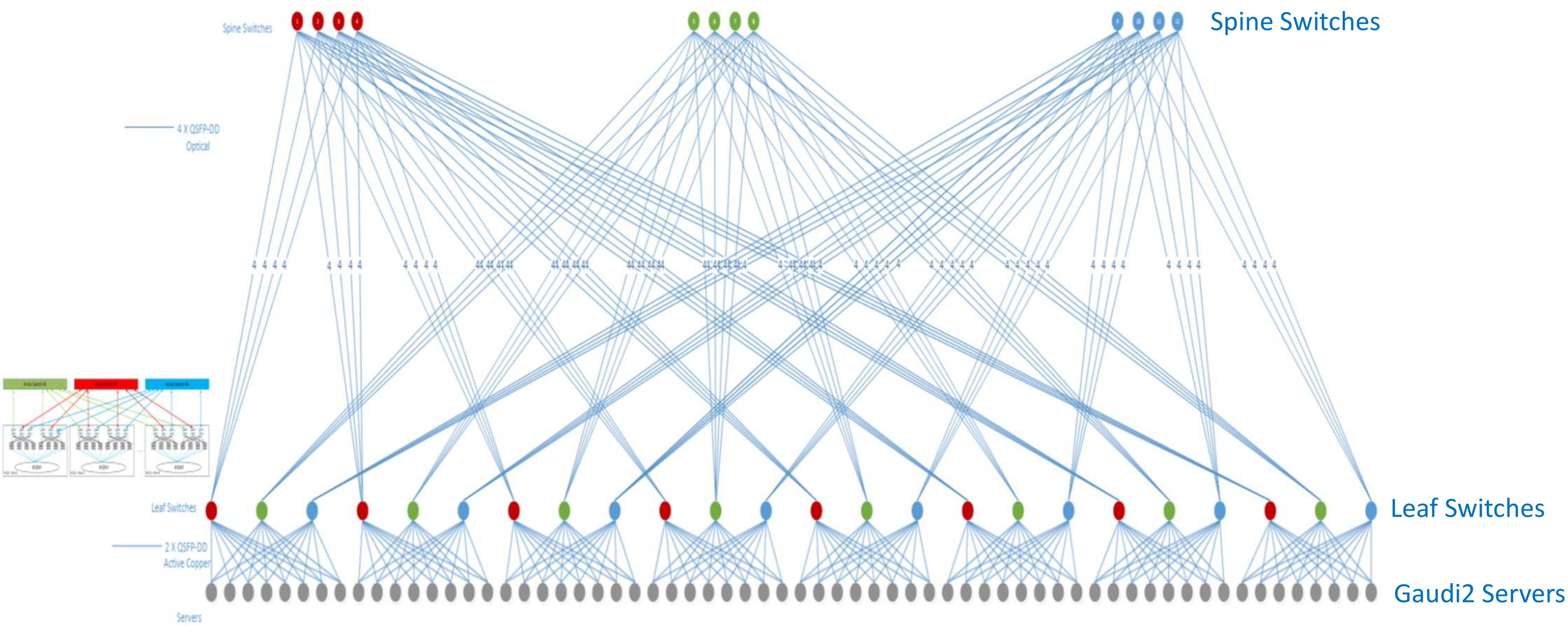


Dual-socket Intel® Xeon® 3rd Gen Scalable Host CPU

16 servers (128 Gaudi2s) datacenter

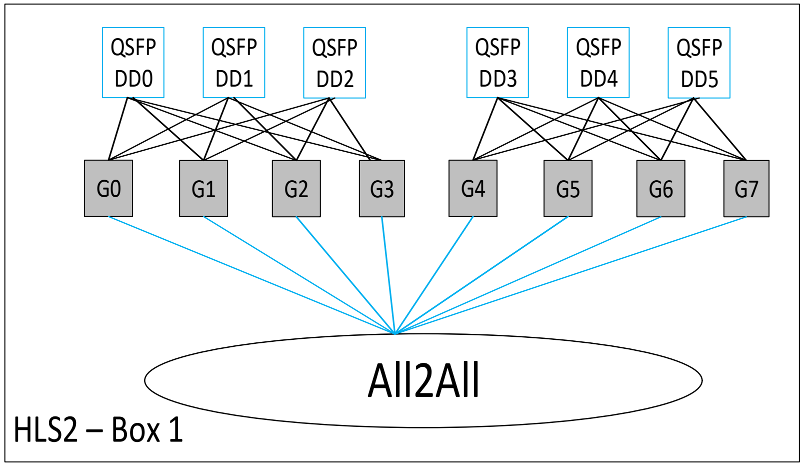


Gaudi Network for 512 Gaudi2s (64 Servers)



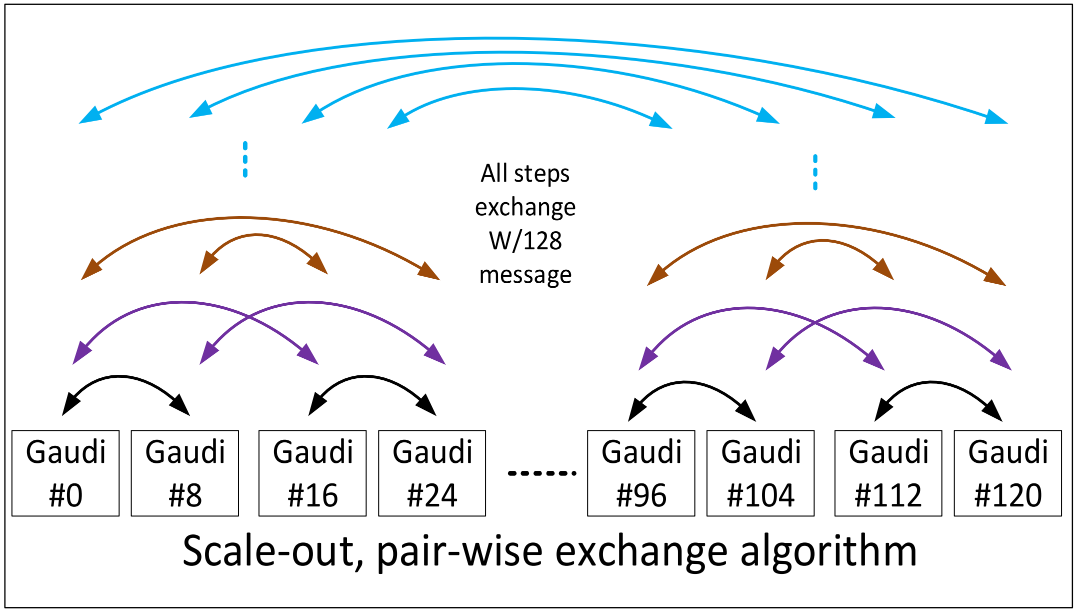
HCCL Hierarchical Collectives

Step 1: Intra-HLS box communication



Scale-up algorithm

Step 2: Inter-HLS box communication



High-level flow of hcclAllreduce in a box

Original

P0	P1	P2	P3	P4	P5	P6	P7
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8

Reduce-Scatter

P0	P1	P2	P3	P4	P5	P6	P7
8							
	16						
		24					
			32				
				40			
					48		
						56	
							64

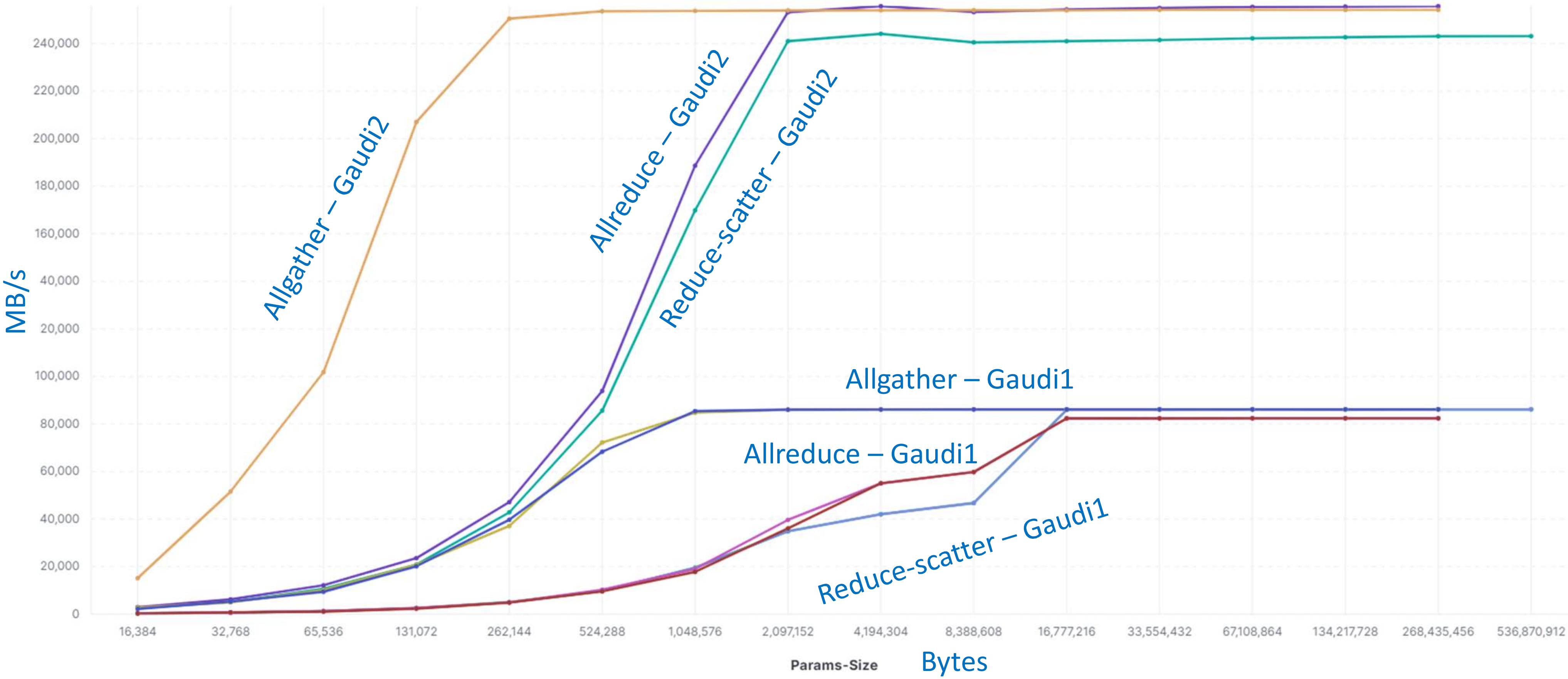
All-gather

P0	P1	P2	P3	P4	P5	P6	P7
8	8	8	8	8	8	8	8
16	16	16	16	16	16	16	16
24	24	24	24	24	24	24	24
32	32	32	32	32	32	32	32
40	40	40	40	40	40	40	40
48	48	48	48	48	48	48	48
56	56	56	56	56	56	56	56
64	64	64	64	64	64	64	64

```
hcclAllReduce(void* sbuff, void* rbuff,
size_t count, hcclDataType_t  datatype,
hcclRedOp_t op, hcclComm_t comm,
synStreamHandle stream_handle);
```

- Address exchange the buffers
- Reduce-scatter + allgather + remainder, slice large allreduce into small ones; on each slice:
- Reduce-scatter
  - (Send) RDMA write 1/8th of the tensor to every Gaudi and Recv 1/8th from every Gaudi
  - SRAM reduction: read from HBM, reduce in SRAM and write result back to HBM
  - Ordering guarantees, reproducible reductions
- Allgather
  - (Send) RDMA write the tensor to every other Gaudi

HCCL Collective Performance (Gaudi1 and Gaudi2) in a box



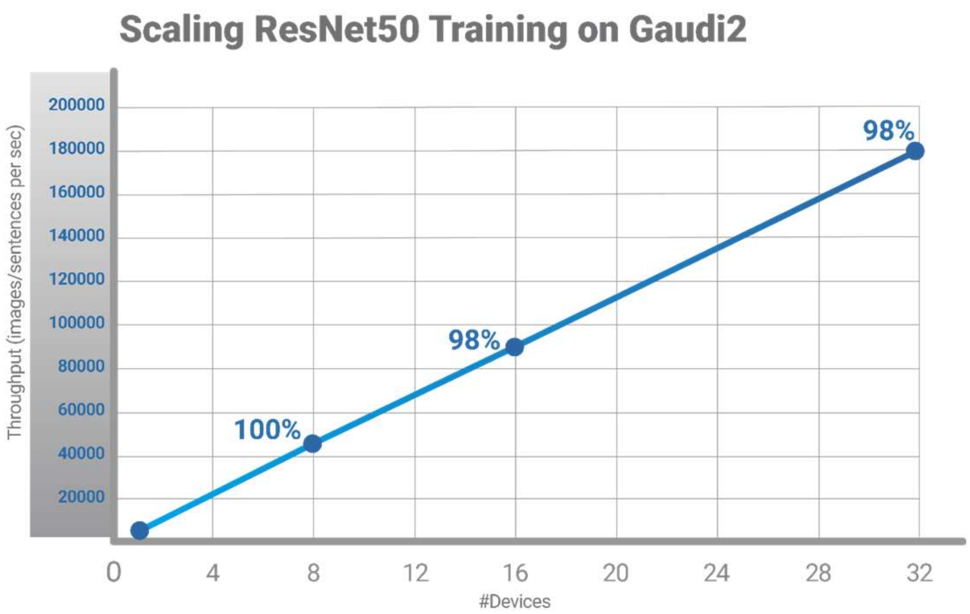
We achieve full utilization in message sizes that are much smaller than other RDMA solutions

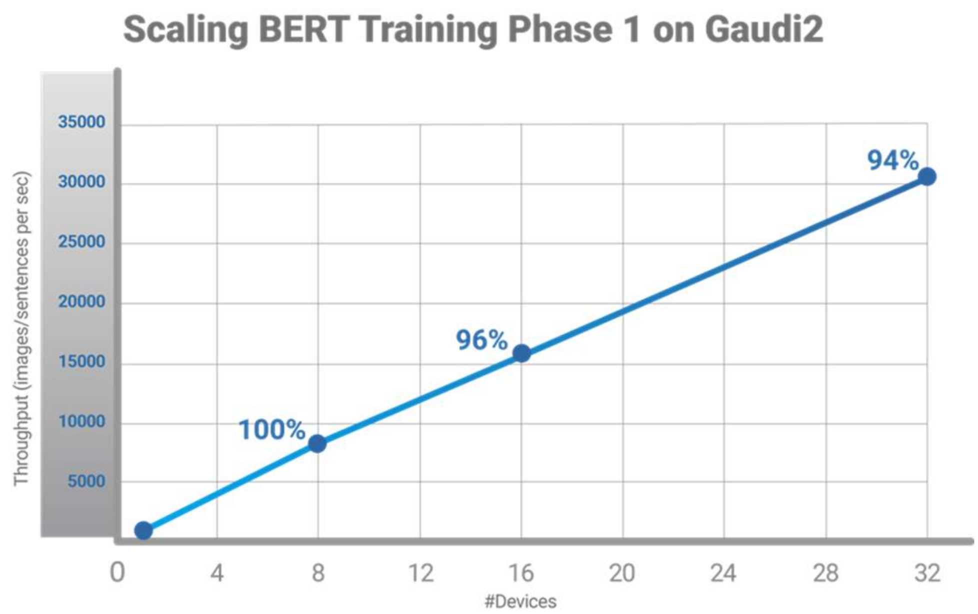
HCCL API

```
// Communicator creation
hcclGetUniqueId(hcclUniqueId* uniqueId);
hcclCommInitRank(hcclComm_t* comm, int nranks, hcclUniqueId 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();
```

Near-linear scaling with Gaudi2

– High scaling efficiency across multiple workloads drives performance and TCO advantages

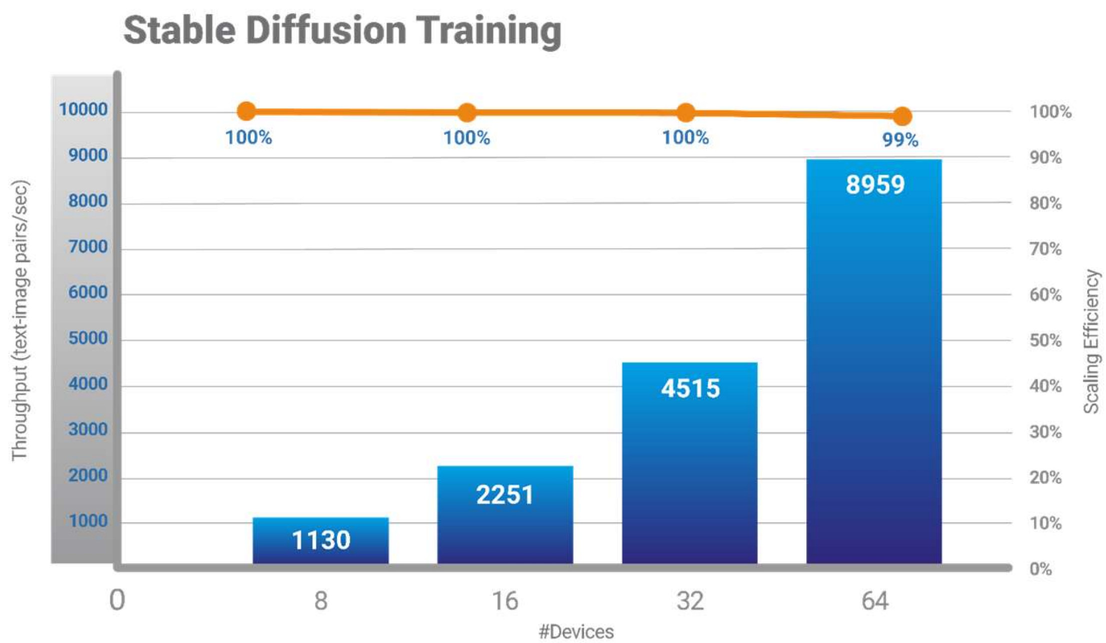




Model scripts available at https://github.com/HabanaAI/Model-References
Performance results available at https://developer.habana.ai/resources/habana-models-performance/
Measurements based on SynapseAI 1.9 (PyTorch 1.13.1)

Stable Diffusion Training on Gaudi2

Linear scaling efficiency > 99% up to 64x cards



Model source: https://github.com/HabanaAI/Model-References/tree/master/PyTorch/generative_models/stable-diffusion-training, Dataset laion2B-en.
Training with BF16, batch size = 16, global batch size = 1024, for 1K iterations. Image size 256x256. Measurements using SynapseAI 1.9.0

Performance of Reference Models

Gaudi Reference Models Training Performance

https://developer.habana.ai/resources/habana-models-performance/

Show 25 entries   Search:

Framework Version	Model	# HPU	Precision	Throughput	Accuracy	TTT	Batch
Select Framework	Filter Model						
TensorFlow 2.8.2	ResNet50 Keras LARS	32					
TensorFlow 2.8.2	ResNet50 Keras LARS	16					
TensorFlow 2.8.2	ResNet50 Keras LARS	8					
TensorFlow 2.9.1	ResNet50 Keras LARS	1					
PyTorch 1.12.0	ResNet50 SGD	16					
PyTorch 1.12.0	ResNet50 SGD	8					
TensorFlow 2.8.2	BERT-Large Pre Training combine	32					
TensorFlow 2.9.1	BERT-Large Pre Training combine	8					
TensorFlow 2.9.1	BERT-Large Pre Training combine	1					
TensorFlow 2.8.2	BERT-Large Pre Training phase 1	32					
TensorFlow 2.9.1	BERT-Large Pre Training phase 1	8					
TensorFlow 2.9.1	BERT-Large Pre Training phase 1	1					
TensorFlow 2.8.2	BERT-Large Pre Training phase 2	32					
TensorFlow 2.9.1	BERT-Large Pre Training phase 2	8					
TensorFlow 2.9.1	BERT-Large Pre Training phase 2	1					
TensorFlow 2.9.1	BERT-Large Fine Tuning (SQUAD)	8					
TensorFlow 2.8.2	BERT-Large Fine Tuning (SQUAD)	1					
PyTorch 1.12.0	BERT-Large Pre Training combine	32					
PyTorch 1.12.0	BERT-Large Pre Training combine	8					
PyTorch 1.12.0	BERT-Large Pre Training combine	1					
PyTorch 1.12.0	BERT-L Pre Training Phase 1	32					
PyTorch 1.12.0	BERT-L Pre Training Phase 1	8					
PyTorch 1.12.0	BERT-L Pre Training Phase 1	1					
PyTorch 1.12.0	BERT-L Pre Training Phase 2	32					
PyTorch 1.12.0	BERT-L Pre Training Phase 2	8					

Framework Version	Model	# HPU	Precision	Throughput	Accuracy	TTT	Batch
Select Framework	Filter Model						
PyTorch 1.12.0	BERT-L Pre Training Phase 2	1					
PyTorch 1.12.0	BERT-L SQUAD Fine Tuning	8					
PyTorch 1.12.0	BERT-L SQUAD Fine Tuning	1					
PyTorch 1.12.0	BERT-XL-1.2B Pre Training Phase 1	8					
PyTorch 1.12.0	BERT-XL-1.2B Pre Training Phase 2	8					
DeepSpeed 0.6.0	BERT 1.5B LANS Pre Training Phase 1	64					
DeepSpeed 0.6.0	BERT 1.5B LANS Pre Training Phase 1	32					
DeepSpeed 0.6.0	BERT 1.5B LANS Pre Training Phase 1	16					
DeepSpeed 0.6.0	BERT 1.5B LANS Pre Training Phase 1	8					
TensorFlow 2.8.2	SSD	8					
TensorFlow 2.9.1	SSD	1					
PyTorch 1.12.0	SSD	8					
PyTorch 1.12.0	SSD	1					
PyTorch 1.12.0	ResNext101	8					
TensorFlow 2.8.2	Resnext-101	8					
TensorFlow 2.8.2	Resnext-101	1					
PyTorch 1.12.0	ResNet152	8					
TensorFlow 2.9.1	UNet2D	8					
TensorFlow 2.9.1	UNet2D	1					
TensorFlow 2.9.1	UNet3D	8					
TensorFlow 2.9.1	UNet3D	1					
Lightning 1.6.4	Unet2D	8					
Lightning 1.6.4	Unet2D	1					
Lightning 1.6.4	Unet3D	8					
Lightning 1.6.4	Unet3D	1					

Framework Version	Model	# HPU	Precision	Throughput	Accuracy	TTT	Batch
Select Framework	Filter Model						
PyTorch 1.12.0	Transformer						
PyTorch 1.12.0	Transformer						
TensorFlow 2.8.2	Transformer						
TensorFlow 2.8.2	Transformer						
TensorFlow 2.9.1	MaskRCNN						
TensorFlow 2.8.2	MaskRCNN						
TensorFlow 2.8.2	Vision Transformer						
TensorFlow 2.8.0	RetinaNet						
TensorFlow 2.9.1	Densenet 121 TFD						
TensorFlow 2.8.2	T5 Base						
TensorFlow 2.9.1	VGG SegNet						
TensorFlow 2.8.2	EfficientDet						
TensorFlow 2.8.2	CycleGAN						
TensorFlow 2.8.2	WideAndDeep						
TensorFlow 2.8.2	Electra Fine Tuning						
TensorFlow 2.9.1	DistilBERT						
PyTorch 1.12.0	GoogLeNet						
PyTorch 1.12.0	DistilBERT						
PyTorch 1.12.0	DistilBERT						
PyTorch 1.12.0	RoBERTa Large						
PyTorch 1.12.0	RoBERTa Large						
PyTorch 1.12.0	RoBERTa Base						
PyTorch 1.12.0	RoBERTa Base						
PyTorch 1.12.0	ALBERT-XXL Fine Tuning						
PyTorch 1.12.0	ALBERT-XXL Fine Tuning						

Framework Version	Model	# HPU	Precision	Th
Select Framework	Filter Model			
PyTorch 1.12.0	ALBERT-Large Fine Tuning	8	bf16	37
PyTorch 1.12.0	ALBERT-Large Fine Tuning	1	bf16	51
PyTorch 1.12.0	BART Fine Tuning	8	bf16	15
PyTorch 1.12.0	BART Fine Tuning	1	bf16	27
PyTorch 1.10.2	MobileNetV2	1	bf16	15
PyTorch 1.12.0	Vision Transformer	8	bf16	66
PyTorch 1.12.0	Vision Transformer	1	bf16	85
PyTorch 1.11.0	ElectraLD FT	8	bf16	21
PyTorch 1.12.0	YOLOv5	8	bf16	56
PyTorch 1.12.0	YOLOv5	1	bf16	10
PyTorch 1.12.0	DINO	8	bf16	92
PyTorch 1.12.0	DINO	1	bf16	15
PyTorch 1.12.0	Wav2Vec 2.0	8	bf16	19
PyTorch 1.12.0	Wav2Vec 2.0	1	bf16	28
PyTorch 1.12.0	YOLOX	8	bf16	31
PyTorch 1.12.0	YOLOX	1	bf16	64
TensorFlow 2.9.1	Unet Industrial	8	bf16	73
TensorFlow 2.8.2	ResNet50 Keras LARS tf.distribute	8	bf16	12
TensorFlow 2.8.2	ResNet50 Keras LARS Host NIC (HVD and Libfabric)	16	bf16	23

Q&A

habana
An Intel Company

Legal Notices and Disclaimers

Performance varies by use, configuration and other factors. See details on performance and other related claims at the QR code below or 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. No product or component can be absolutely secure.

Your costs and results may vary.

Legal Notices and Disclaimers

– For notices, disclaimers, and details about performance claims, visit https://habana.ai/habana-claims-validation

or scan the QR code:

Thank You.

habana
An Intel Company