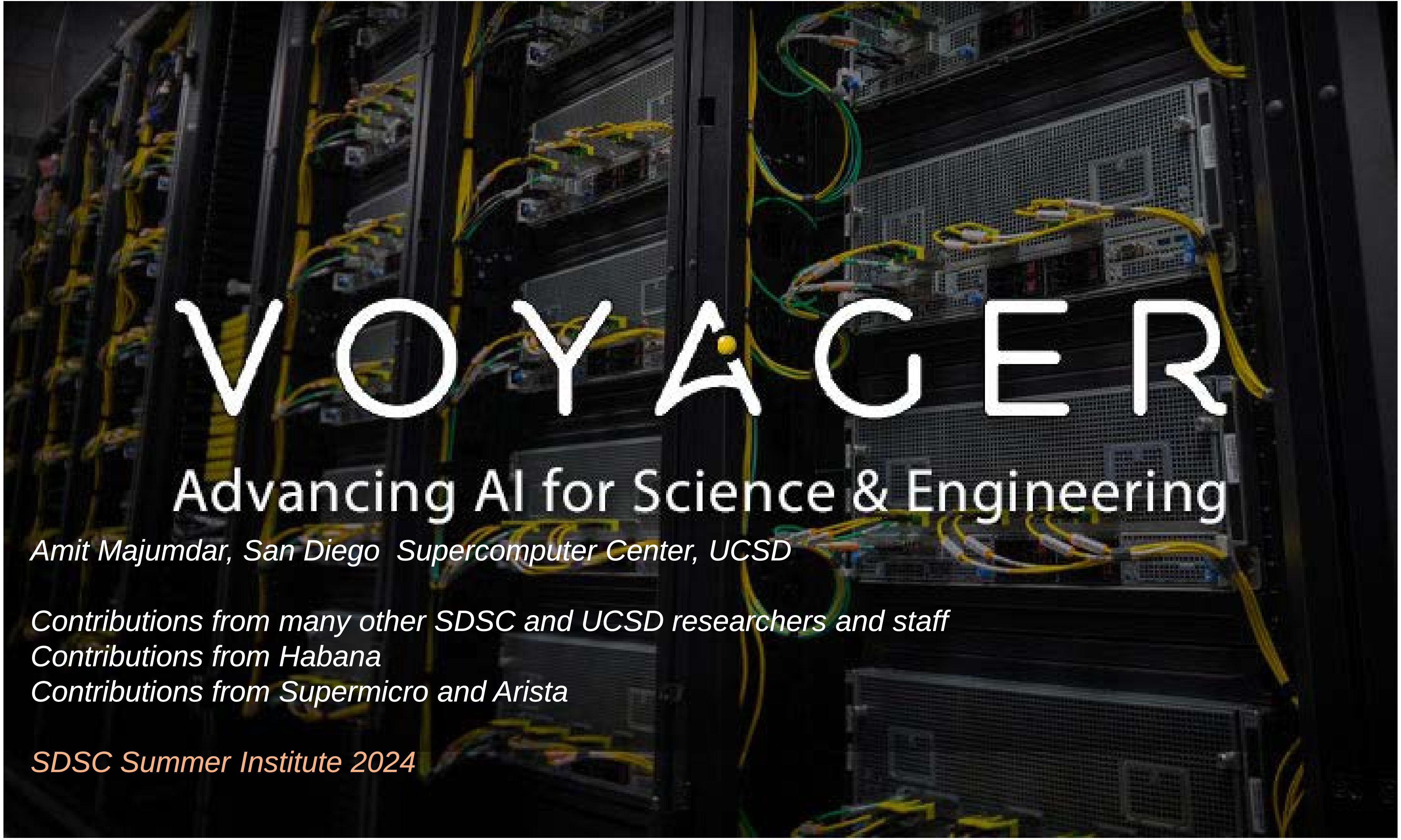


VOYAGER

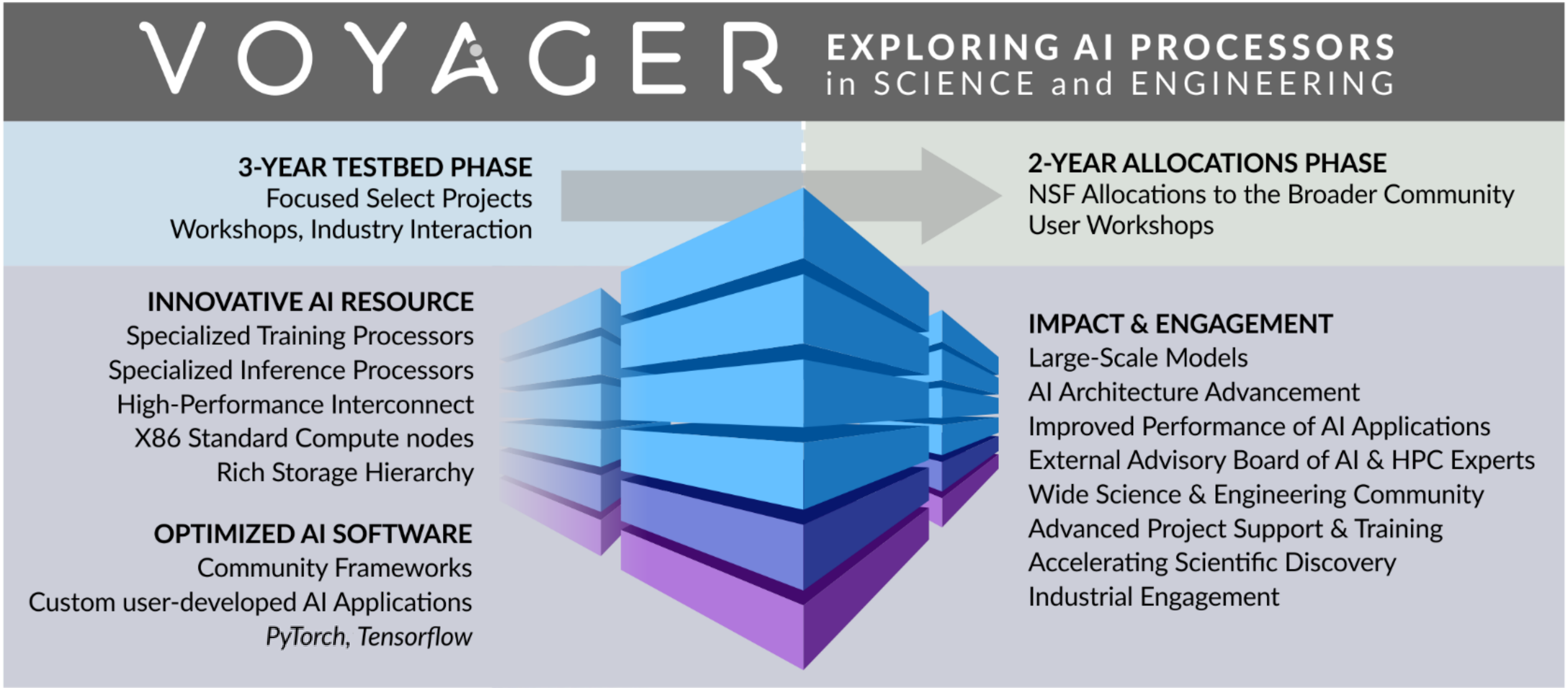
Advancing AI for Science & Engineering

Amit Majumdar, San Diego Supercomputer Center, UCSD

Contributions from many other SDSC and UCSD researchers and staff
Contributions from Habana
Contributions from Supermicro and Arista

SDSC Summer Institute 2024

- **Voyager Project**
- Voyager Architecture
- Voyager Applications



VOYAGER

EXPLORING AI PROCESSORS in SCIENCE and ENGINEERING

3-YEAR TESTBED PHASE
Focused Select Projects
Workshops, Industry Interaction

2-YEAR ALLOCATIONS PHASE
NSF Allocations to the Broader Community
User Workshops

INNOVATIVE AI RESOURCE
Specialized Training Processors
Specialized Inference Processors
High-Performance Interconnect
X86 Standard Compute nodes
Rich Storage Hierarchy

OPTIMIZED AI SOFTWARE
Community Frameworks
Custom user-developed AI Applications
PyTorch, Tensorflow

IMPACT & ENGAGEMENT
Large-Scale Models
AI Architecture Advancement
Improved Performance of AI Applications
External Advisory Board of AI & HPC Experts
Wide Science & Engineering Community
Advanced Project Support & Training
Accelerating Scientific Discovery
Industrial Engagement

Category II System, NSF Award # 2005369

PI: Amit Majumdar (SDSC); Co PIs: Rommie Amaro (UCSD), Javier Duarte (UCSD), Mai Nguyen (SDSC), Robert Sinkovits (SDSC)

Arista 7808R
400 GbE switch

6x Gaudi training
nodes/rack (7 racks total)

Voyager: NSF Category II Award to UCSD

Amit Majumdar, SDSC, PI
Rommie Amaro, UCSD Chemistry and Biochemistry Department, Co-PI
Javier Duarte, UCSD Physics Department, Co-PI
Mai Nguyen, SDSC, Co-PI
Robert Sinkovits, SDSC, Co-PI
Shawn Strande, SDSC Deputy Director

External Advisory Board

- Rich Loft, HPC Consultant, ex-NCAR
- DK Panda OSU
- Linda Petzold UCSB
- Rick Stevens ANL

Science Use Case Researchers
Industrial Relations – SDSC Advanced Technology Lab (ATL)

Christopher Irving, SDSC, HPC Systems Manager
Haisong Cai
Trevor Cooper
Dmitry Mishin
Fernando Silva

Tom Hutton, HPC Networking
Scott Sakai, Securities group
Mary Thomas, EOT

Mahidhar Tatineni, SDSC, User Services Manager
Marty Kandes
Paul Rodriguez
Nicole Wolter
Javier Hernandez Nicolau
Madhu Gujral

The project is structured in two phases: a 3-year testbed, followed by a 2-year allocations phase

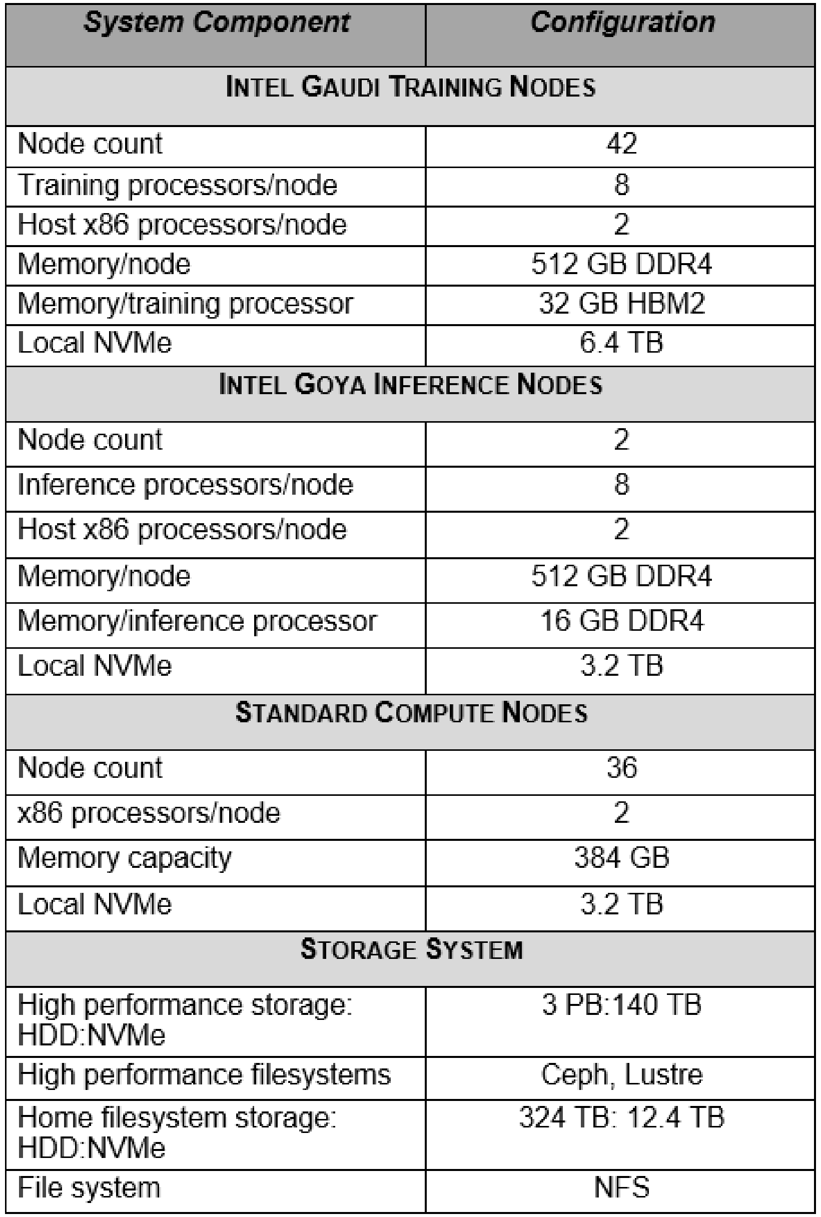
- **Testbed Phase – started May 2022**
  - Work closely with select research groups – deep user engagement
  - Evaluate *Voyager's* innovative DL hardware, software, libraries, ML application porting/performance
  - Semiannual workshops, user forums to share lessons learned, bring researchers together
  - Develop knowledge base, best use cases for future users, allocation policies
  - External Advisory Board to help recruit research groups, provide guidance to project

- **Allocations Phase**
  - Allocate via NSF-approved process
  - Lessons learned from Testbed phase inform documentation and training
  - Regular and advanced user support
  - Semiannual workshops
  - Industry engagement for similar technology evaluation

- Voyager Project
- **Voyager Architecture**
- Voyager Applications

Voyager is a heterogeneous system designed to support complex AI workflows

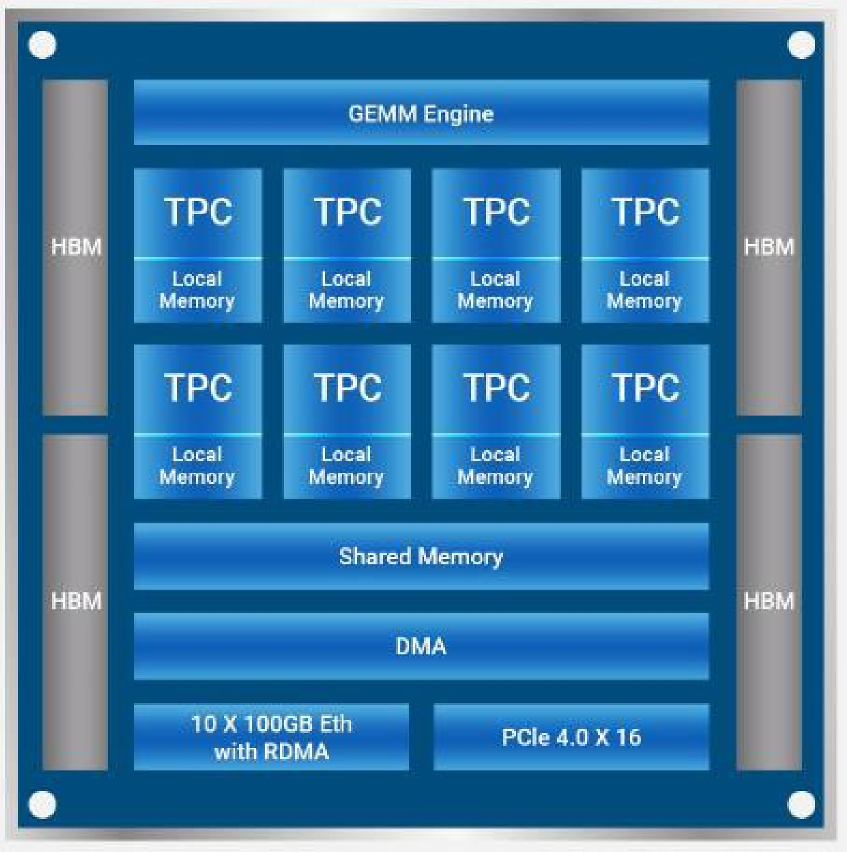
- **42x Intel Habana Gaudi** training nodes, each with 8 training processors (**336 in total**); all-to-all network between processors on a node
- Gaudi processors feature specialized hardware units for AI, HBM2, and on-chip high-speed Ethernet
- **2x first generation inference nodes**, each with 8 inference processors (**16 in total**)
- **36x Intel x86 processors compute nodes** for general purpose computing and data processing
- **400 GbE interconnect** using RDMA over Converged Ethernet
- **3 PB Storage** system connected vis 25GbE. Deployed as Ceph, but open to others
- 324 TB HFS; connectivity to compute via 25GbE
- Machine integrated by Supermicro; includes Arista switch



System Component	***Configuration***
INTEL GAUDI TRAINING NODES	
Node count	42
Training processors/node	8
Host x86 processors/node	2
Memory/node	512 GB DDR4
Memory/training processor	32 GB HBM2
Local NVMe	6.4 TB
INTEL GOYA INFERENCE NODES	
Node count	2
Inference processors/node	8
Host x86 processors/node	2
Memory/node	512 GB DDR4
Memory/inference processor	16 GB DDR4
Local NVMe	3.2 TB
STANDARD COMPUTE NODES	
Node count	36
x86 processors/node	2
Memory capacity	384 GB
Local NVMe	3.2 TB
STORAGE SYSTEM	
High performance storage: HDD:NVMe	3 PB:140 TB
High performance filesystems	Ceph, Lustre
Home filesystem storage: HDD:NVMe	324 TB: 12.4 TB
File system	NFS

Gaudi: Architected for performance and efficiency

- *Fully programmable Tensor Processing Cores (TPC) with tools & libraries*
- *Configurable Matrix Math Engine (GEMM)*
- *Multi-stage memory hierarchy with 32GB HBM2 memory*
- *Integrated 10 x 100 Gigabit Ethernet for multi-chip scale-out training*



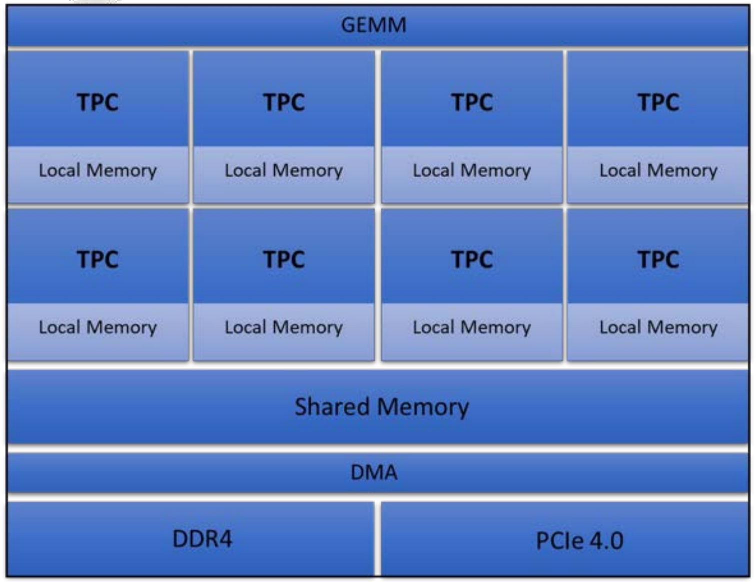
Habana Labs © 2021

Designed for flexible and easy model migration

Ease of use	*Customization*	*Balanced compute & memory*
Integrated with TensorFlow and PyTorch; minimal code changes to get started ➔ *SynapseAI maps model topology onto Gaudi devices*	*SynapseAI TPC SDK facilitates development of custom kernels*	*32GB HBM2 memories similar to GPUs, so existing DL models will fit into Gaudi memory*
*Developers can enjoy the **same abstraction** they are accustomed to today*	*Developers can **customize** models to extract best performance*	*Developers can spend **less effort** to port their models to Gaudi*

Habana First-Generation Inference - architecture

- Specifically Designed for Inference
- 8 Tensor Processing Cores (TPC)
- Configurable Matrix Math Engine (GEMM)
- 16 GB of DDR4 but no HMB2
- Larger Shared Memory than the Gaudi



GEMM

TPC | TPC | TPC | TPC

Local Memory | Local Memory | Local Memory | Local Memory

TPC | TPC | TPC | TPC

Local Memory | Local Memory | Local Memory | Local Memory

Shared Memory

DMA

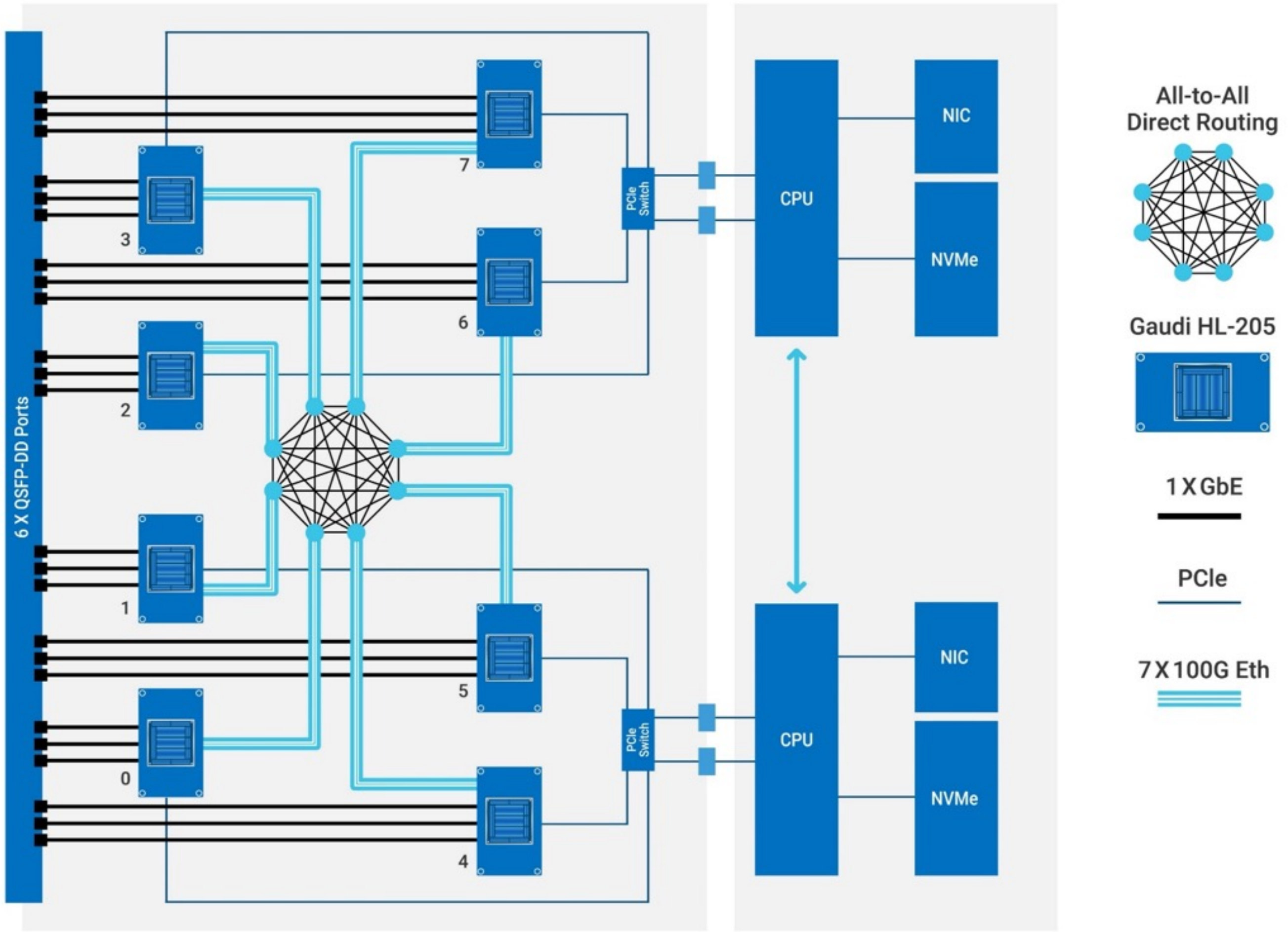
DDR4 | PCIe 4.0

Voyager's three distinct networks support application performance, data movement, and systems management

- 400 GbE for scale-out training
  - Six connections from each Gaudi node to a single 230 Tbps Arista 7808 non-blocking switch
- 25 GbE Bonded Control and data network
  - Every node has a bonded 50GbE (2 X 25GbE) connection.
- 1 GbE out-of-band management network for IPMI and other traffic.

Gaudi servers supports all-to-all connectivity

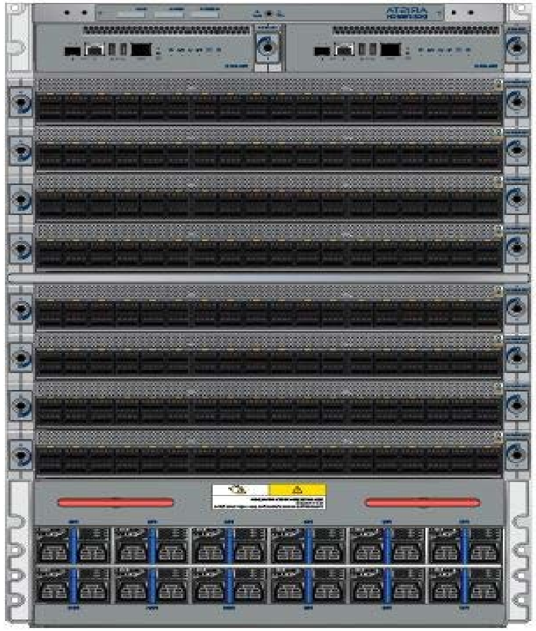
- 8 Gaudi OCP OAM cards
- 24 x 100GbE RDMA RoCE for scale-out
- Non-blocking, all-2-all internal interconnect across Gaudi AI processors
- Separate PCIe ports for external Host CPU traffic



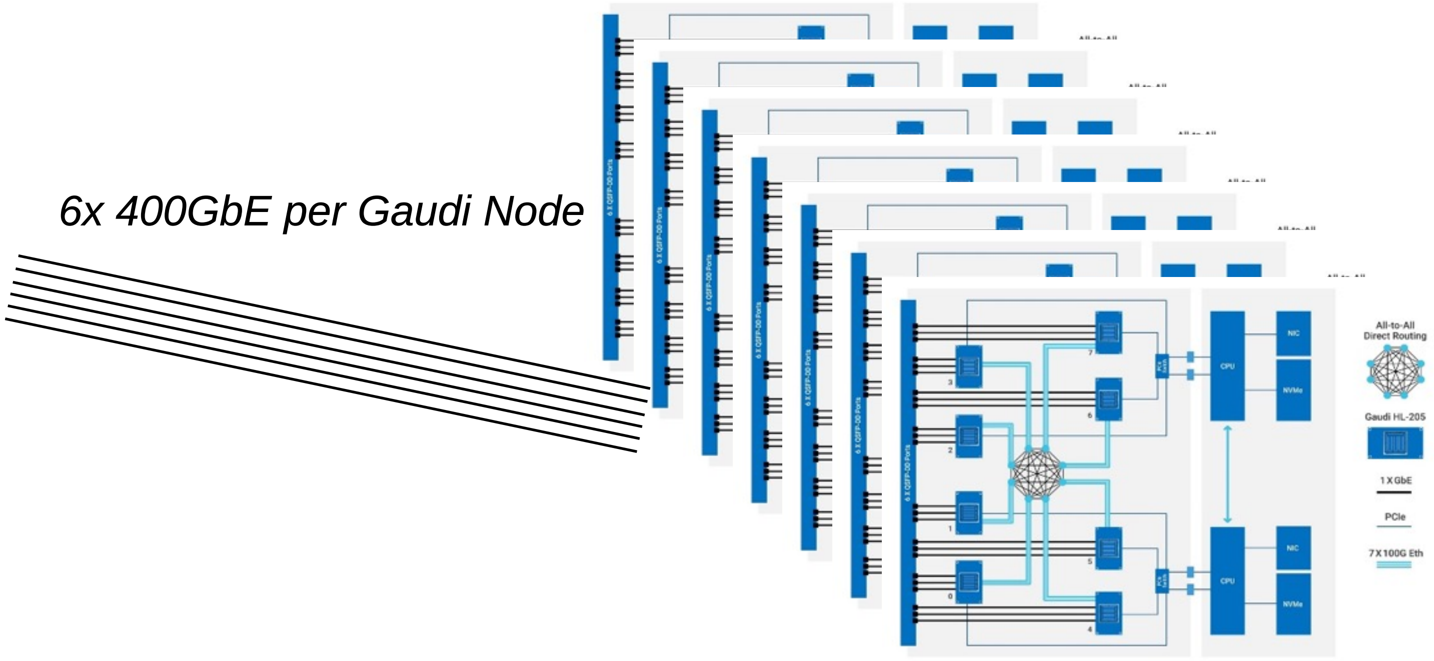
Example of Integrated Server with eight Gaudi AI processors, two Xeon CPU and multiple Ethernet Interfaces

Gaudi design enables highly efficient scaling

- *Natively integrated RoCE on Gaudi processor*
- *6x Quad-100 GbE per node (8x Gaudi)*
- *7808 Arista 400 GbE switch*



7808 Arista 400GbE



6x 400GbE per Gaudi Node

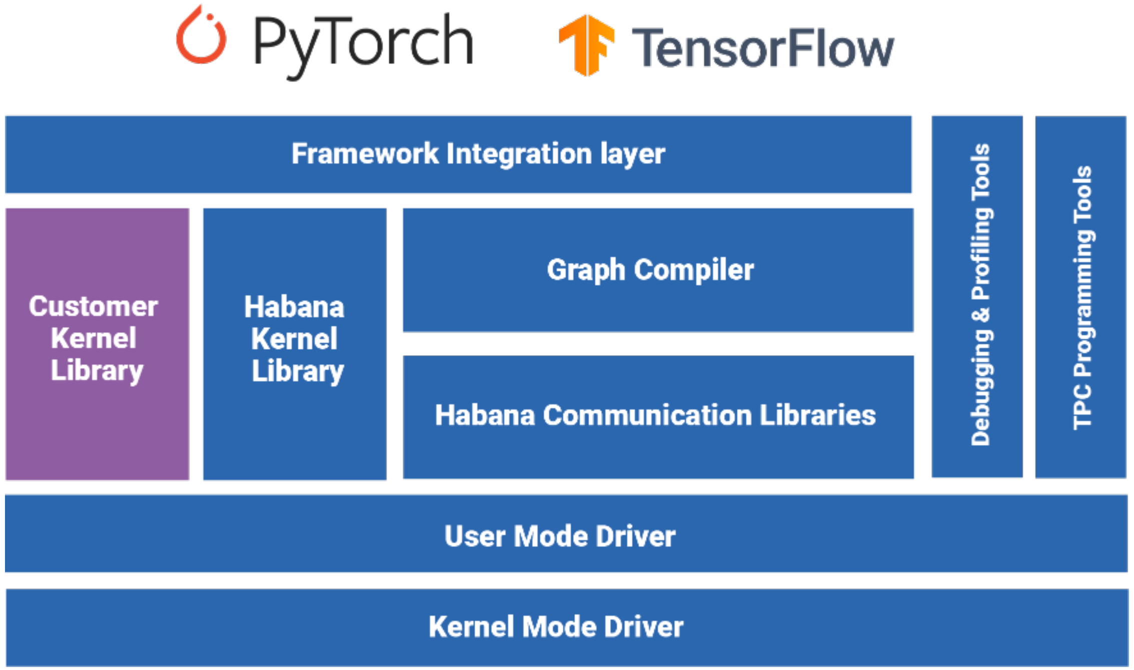
6x Gaudi nodes per rack

Voyager’s storage hierarchy supports AI workflows

	Filesystem	Capacity	Connectivity	Use cases
Node-local NVMe	XFS	3.2 TB 6.4 TB	PCEe 4.0	Large, node-local NVMe drives provide ephemeral storage and excellent performance for workload that don’t need shared data.
Home file system	NFS	324 TB	50 Gb Ethernet	High-Availability Network Files System (NFS) Cluster for user home directory storage
Project storage	Ceph; will investigate other options during Testbed (e.g. VAST being tested)	3 PB	50 Gb Ethernet	Large data, project storage

SynapseAI Software Suite: Designed for Performance and Ease of Use

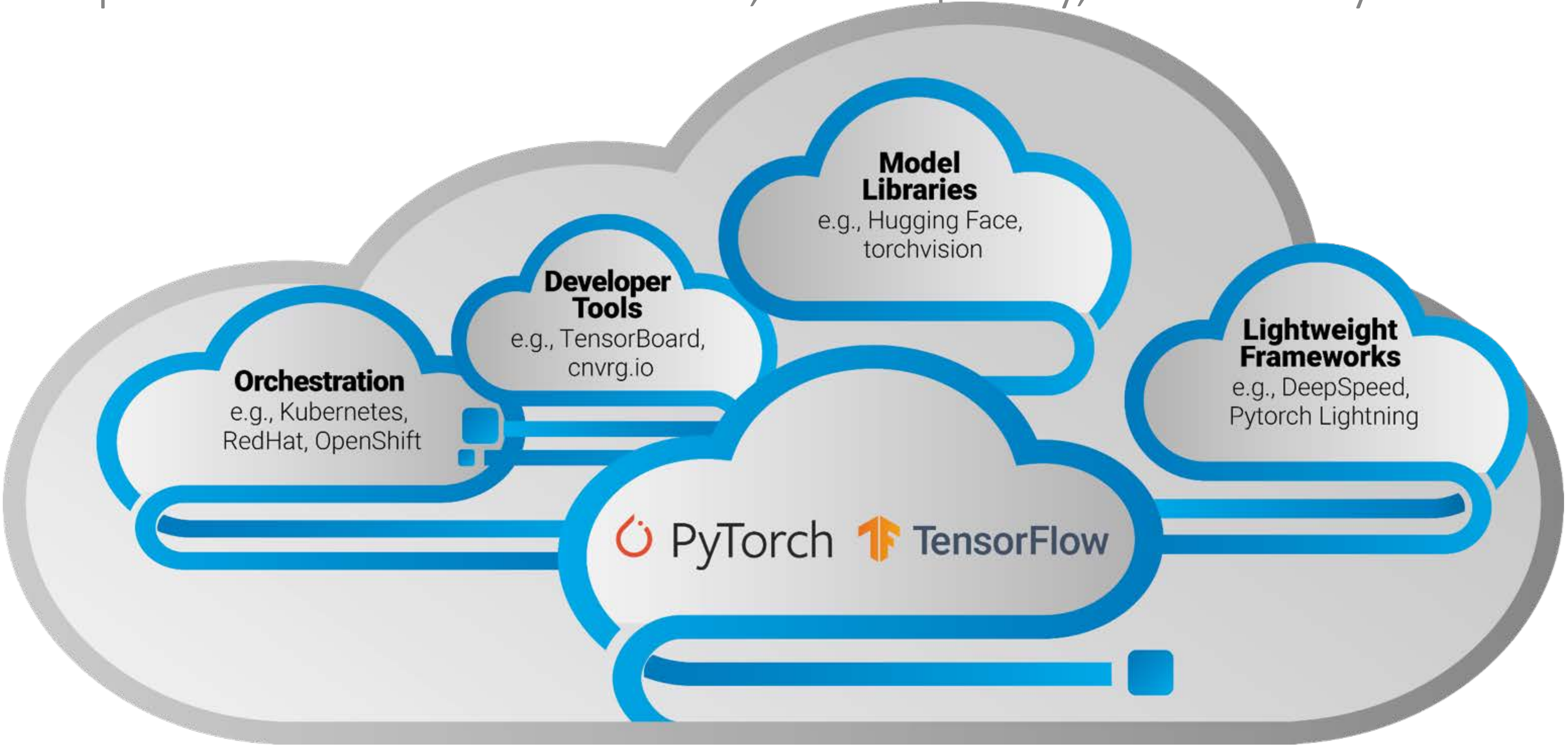
SynapseAI®



- Shared software suite for training and inference
- Start running on Habana 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

Habana Deep Learning Software Ecosystem

– Development on the latest models, more quickly, more easily



Model Libraries
e.g., Hugging Face, torchvision
Developer Tools
e.g., TensorBoard, cnvrg.io
Orchestration
e.g., Kubernetes, RedHat, OpenShift
Lightweight Frameworks
e.g., DeepSpeed, Pytorch Lightning
PyTorch
TensorFlow

SynapseAI®

- Voyager Project
- Voyager Architecture
- **Voyager Applications**

Some of the Applications on *Voyager*

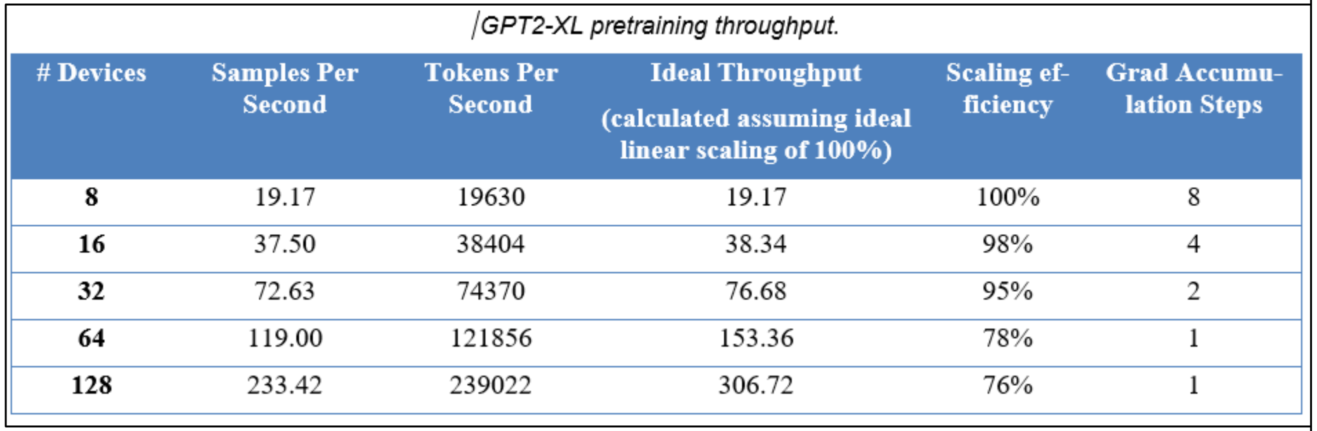
Project	Model
Data Driven Weather Prediction	U-Net
High energy physics	GNN
Cardiac image analysis	U-Net
Biomedical text analytics	BERT DL models
Ultrasound computed tomography	U-Net
Dose prediction in cervical brachytherapy	U-New
Systems biology	Dense neural network
Atmospheric sciences	VAE model
Human microbiome research	Categorical VAE
Astronomy	NN
Cognitive Neuroscience	CNN

Some of the Applications on *Voyager*

Project	Model
Natural language processing	Transformers
Biochemistry – Molecular Dynamics	VAE, AAE, ANCA-AE
Camera trap animal detection	Context R-CNN
Hyperdimensional computing	Graph architecture
Computer vision	VGGnet
E2E ML Pipeline for complex dataset	Hugging Face
Application of DL in Radiology	3D DL models (VGG, ResNet)
MVAPICH MPI implementation on Voyager	Not applicable
Analyzing EEG data with DL	CNN
Research accessibility via visual representation	Diffusion model
Hugging Face GPT2-XL model with 1.5 Billion parameters and GPT3-XL with 1.3 Billion parameters	Large language models
More applications are being ported and tested on Voyager…	

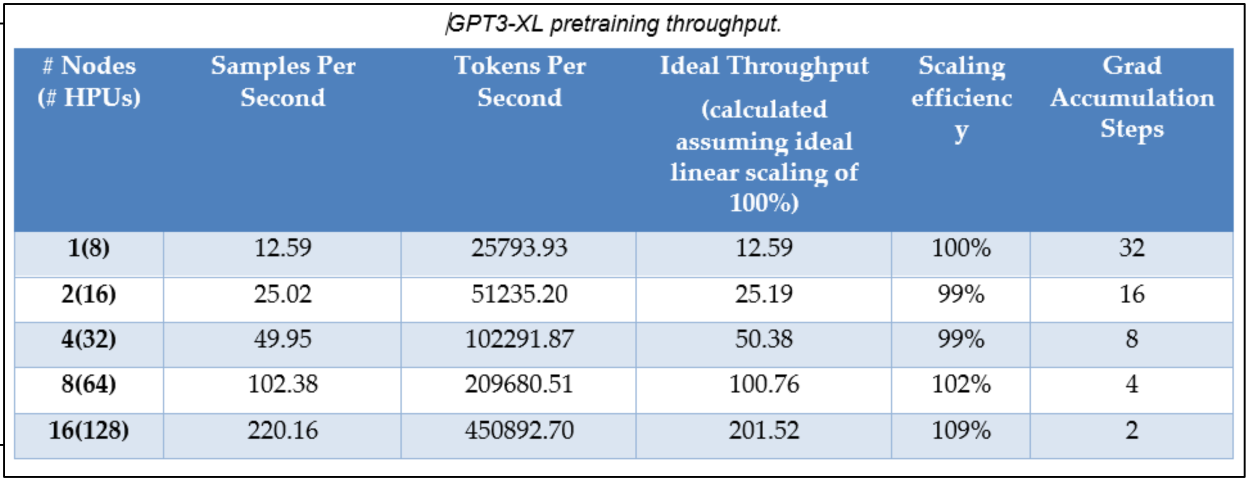
Training of the Hugging Face GPT-2 XL and GPT3-XL model with DeepSpeed ZeRO2 on Voyager

- *Hugging Face GPT2-XL with 1.5 Billion parameters*
- *GPT2-XL numbers are from Synapse version 1.7.0 and with a Global BS of 512*
- *GPT3-XL with 1.3 Billion parameters*
- *GPT3-XL numbers are from Synapse version 1.8.0 and with a Global BS of 2048*
- *DeepSpeed is a popular deep learning software library which includes ZeRO (Zero Redundancy Optimizer), a memory-efficient approach for distributed training*
- *Habana's SynapseAI(R) software currently supports ZeRO-1 and ZeRO-2*



GPT2-XL pretraining throughput.

# Devices	Samples Per Second	Tokens Per Second	Ideal Throughput (calculated assuming ideal linear scaling of 100%)	Scaling efficiency	Grad Accumulation Steps
8	19.17	19630	19.17	100%	8
16	37.50	38404	38.34	98%	4
32	72.63	74370	76.68	95%	2
64	119.00	121856	153.36	78%	1
128	233.42	239022	306.72	76%	1



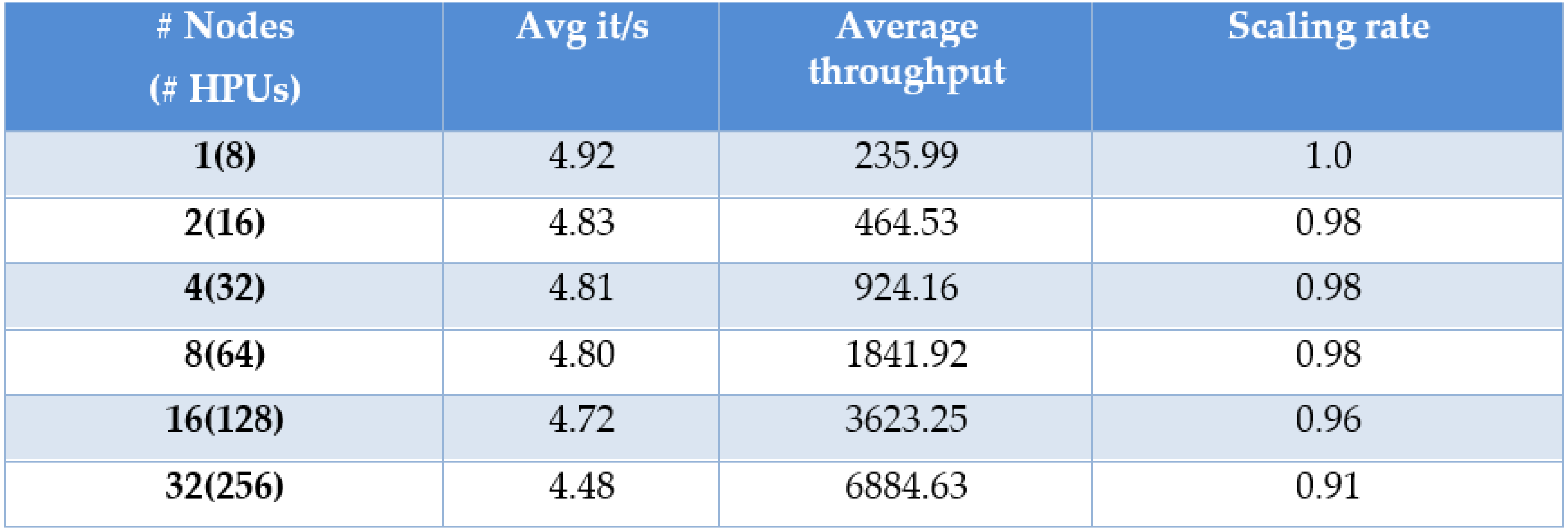
GPT3-XL pretraining throughput.

# Nodes (# HPUs)	Samples Per Second	Tokens Per Second	Ideal Throughput (calculated assuming ideal linear scaling of 100%)	Scaling efficiency	Grad Accumulation Steps
1(8)	12.59	25793.93	12.59	100%	32
2(16)	25.02	51235.20	25.19	99%	16
4(32)	49.95	102291.87	50.38	99%	8
8(64)	102.38	209680.51	100.76	102%	4
16(128)	220.16	450892.70	201.52	109%	2

Training of Stable Diffusion Model on Voyager

- *Stable Diffusion model is based on latent text-to-image diffusion model*
- *Table shows the performance number run with Laion2B-en dataset*
- *Used SynapseAI SW Stack is 1.9.0 pre-release.*
- *The result shows a good scaling rate: with 256 Gaudis, it reached 91% scaling efficiency versus 8 cards*

Stable Diffusion Model Scaling.



# Nodes (# HPUs)	Avg it/s	Average throughput	Scaling rate
1(8)	4.92	235.99	1.0
2(16)	4.83	464.53	0.98
4(32)	4.81	924.16	0.98
8(64)	4.80	1841.92	0.98
16(128)	4.72	3623.25	0.96
32(256)	4.48	6884.63	0.91

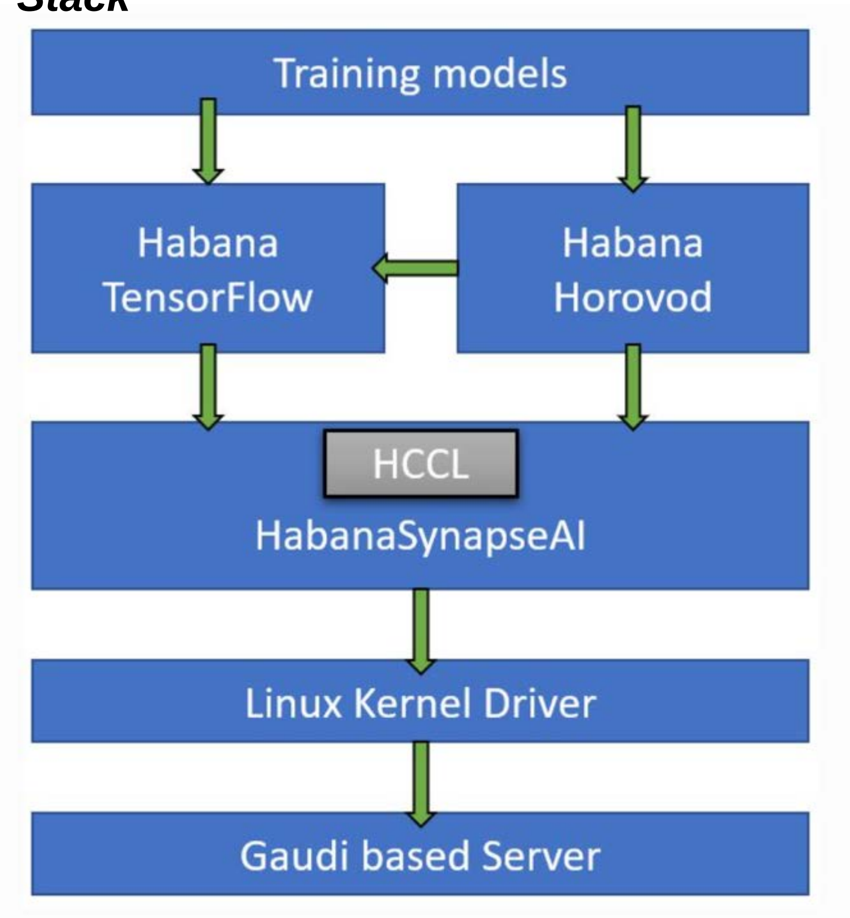
Porting of user applications has been relatively straightforward

- Several applications are now running on Gaudi and the first generation inference card
- So far, our experience is that most codes do not need major changes
- Users can run with familiar TensorFlow and PyTorch frameworks
- Typically, the changes are minor and involved using the Habana module and the Habana integrated versions of PyTorch and TensorFlow. For example:

  ***import** habana_frameworks.torch.core **as** htcore*
- Rich suite of models supported and documented on Model Reference page (https://github.com/HabanaAI/Model-References) and documented on developer site (https://developer.habana.ai/)
- SDSC and researchers assisted users in porting with major contributions from Habana staff.
- Typically do an onboarding call for new projects followed by additional meetings/tickets as needed.

Example: Habana Horovod usage with TensorFlow

Gaudi Distributed Training Software Stack



Code snippets illustrating changes for distributed training using Horovod

```
import tensorflow as tf
from habana_frameworks.tensorflow import load_habana_module
load_habana_module()
import horovod.tensorflow.keras as hvd
#Initialization of Horovod
hvd.init()
```

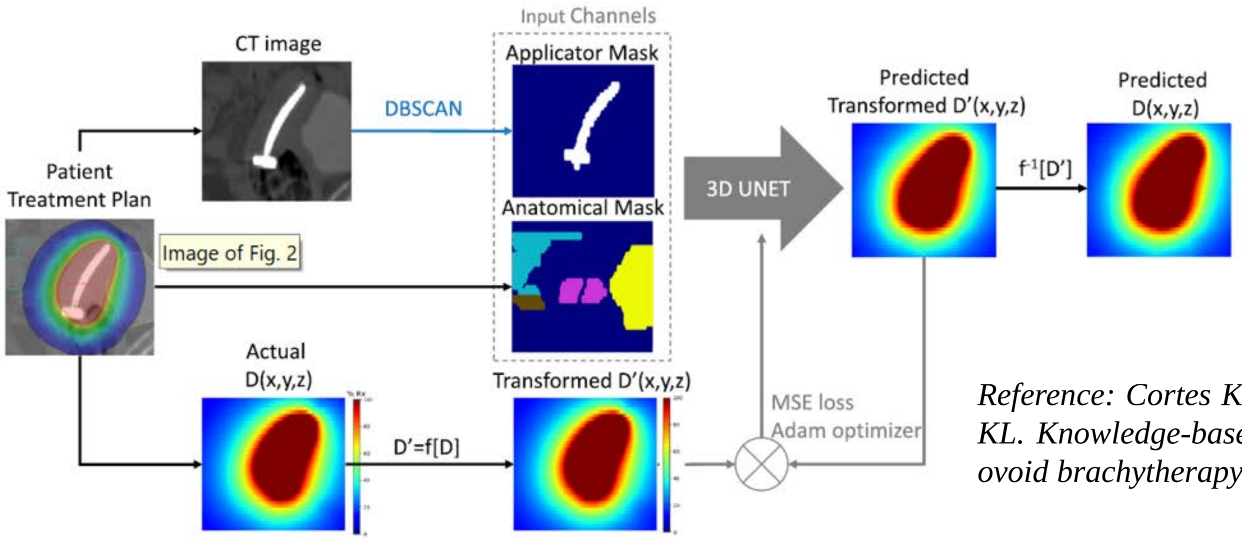
```
# Ensure only 1 process downloads the data on each node
if hvd.local_rank() == 0:
        (x_train, y_train), (x_test, y_test) =
        tf.keras.datasets.mnist.load_data() hvd.broadcast(0, 0)
else:
        hvd.broadcast(0, 0) (x_train, y_train), (x_test, y_test) =
        tf.keras.datasets.mnist.load_data()
```

```
horovod.optimizer = tf.keras.optimizers.SGD(learning_rate=0.01*hvd.size())
optimizer =hvd.DistributedOptimizer(optimizer)
callbacks = [hvd.callbacks.BroadcastGlobalVariablesCallback(0), ]
```

3D-UNET Model for Dose Prediction in Cervical Brachytherapy

Sandra Meyers PI, and Lance Moore,
Department of Radiation Medicine & Applied Sciences, UC San Diego Health

- Cervical cancer is treated with radiation, but it is very difficult to predict in 3D how dosage from applicators will impact other tissue.
- PIs developed a workflow with inputs of possible dosage plan, segmented tissue and applicator mask



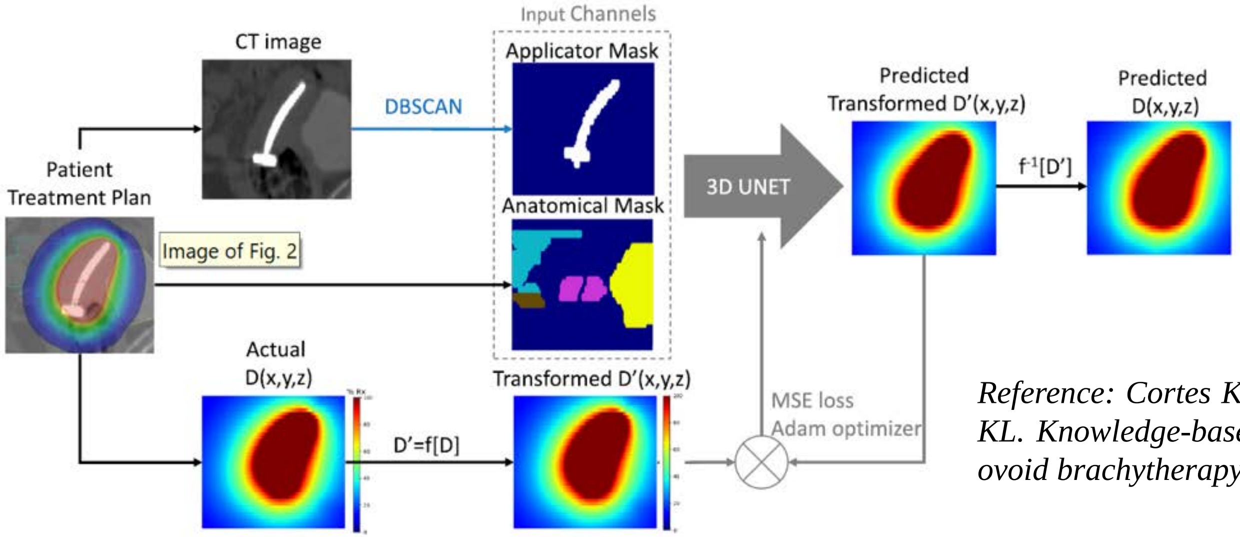
Reference: Cortes KG, Kallis K, Simon A, Mayadev J, Meyers SM, Moore KL. Knowledge-based three-dimensional dose prediction for tandem-and-ovoid brachytherapy. Brachytherapy. 2022 Jul-Aug;21(4):532-542.

The workflow of data transformations and input preparation for the UNET model

3D-UNET Model for Dose Prediction in Cervical Brachytherapy

Goal: Build a deep learning model that can accurately predict the 3D radiation dose from brachytherapy treatment plans for cervical cancer.

Challenges: there are numerous applicator options, and scarcity of data. This leads to questions regarding how best to pool data across applicator types with training of deep learning.



Reference: Cortes KG, Kallis K, Simon A, Mayadev J, Meyers SM, Moore KL. Knowledge-based three-dimensional dose prediction for tandem-and-ovoid brachytherapy. Brachytherapy. 2022 Jul-Aug;21(4):532-542.

The workflow of data transformations and input preparation for the UNET model

3D-UNET Model for Dose Prediction - porting

- PIs developed model for a single GPU device execution in PyTorch
  with some custom data modules fo loading/processing input and loss functions

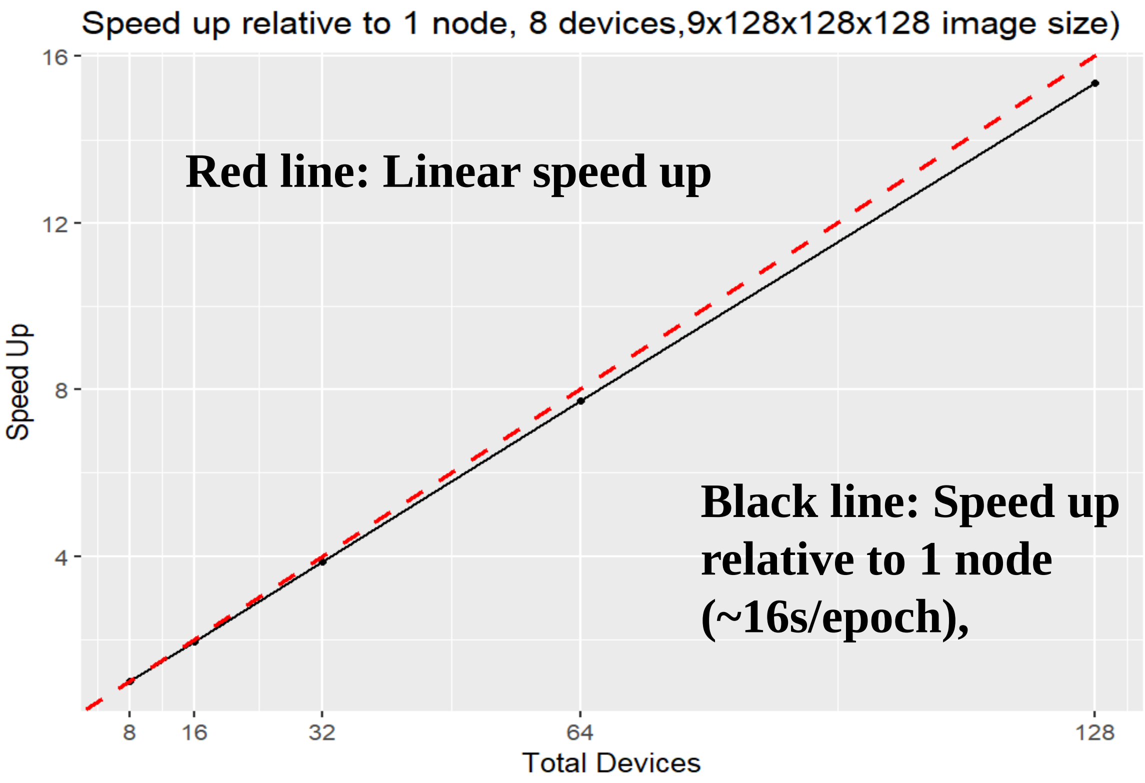
- Each input is nine 3D images for size 9x128x128x128

- We ported the model (32M parameters) by using the Habana guidelines

- We additionally had to port the code from single device to multiple device execution. There were a couple of issues that arose related to the custom loss function and data loading methods. These were due to device placement and properly declaring PyTorch variables with gradient options. But sometimes the errors appeared to be from Habana communication modules, but were in the end simple fixes using correct PyTorch declarations.

3D-UNET Model for Dose Prediction – scaling results on Voyager

We ran through a series of scaling tests with a few epochs, for 1,2,4,8 and 16 nodes, each with 8 Habana HPU devices and tracked the seconds per epoch.



Red line: Linear speed up

Black line: Speed up relative to 1 node (~16s/epoch),

Ultrasound computed tomography (USCT)

PI: Dr. Jiaze (Jason) He, University of Alabama, Department of Aerospace Engineering and Mechanics

- USCT from wave propagation is widely used in medical imaging, nondestructive evaluation, and structural health monitoring
- USCT is the construction of images from wave propagation (Fig 1)
- State of the art methods for reconstruction based on full wave inversion theory (FWI) are computationally expensive (Fig 2)

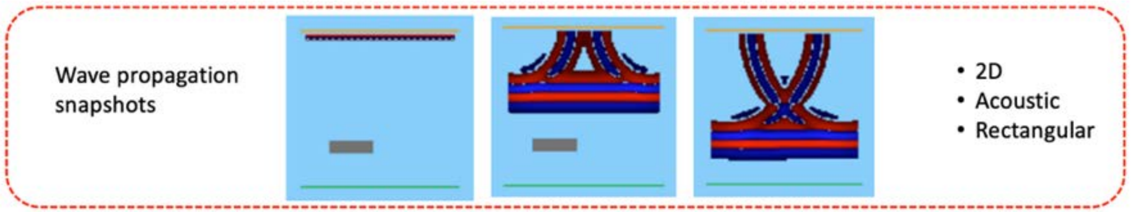


Fig 1.

Aerospace Engineering and Mechanics

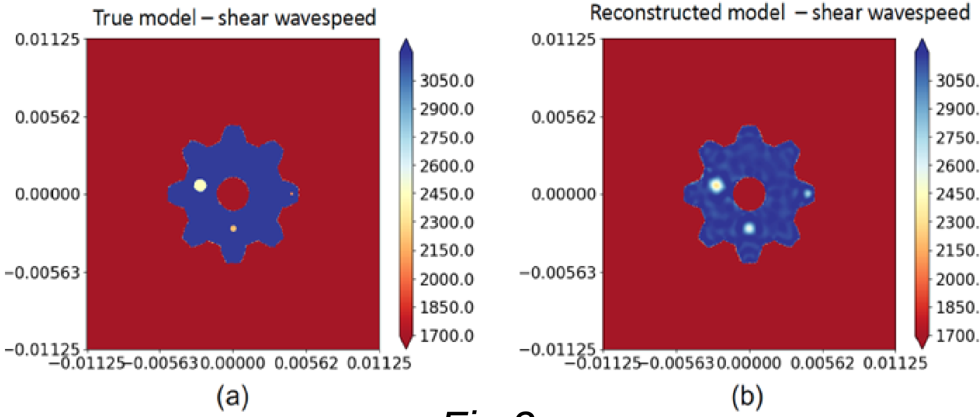


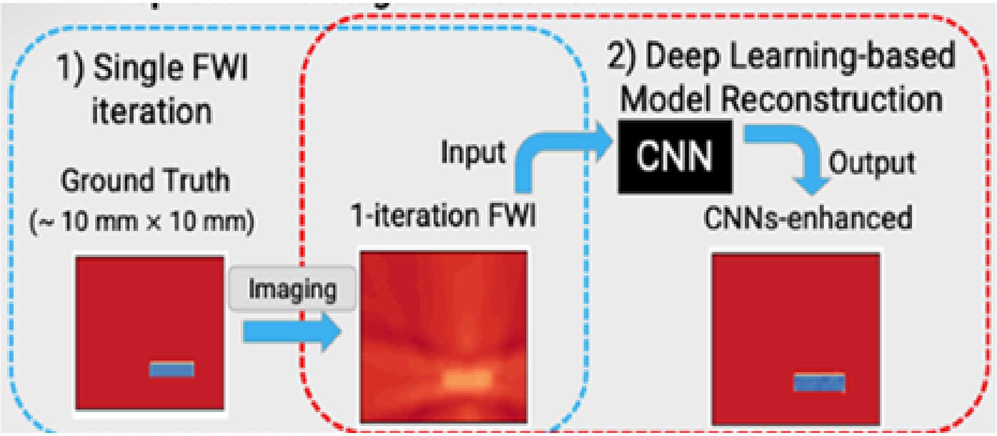
Fig 2.

Ultrasound computed tomography (USCT)

- **Goal:** create tomography-informed neural network framework to significantly improve the reconstruction speed and quality of ultrasonic FWI

- **Challenge:** data is generated from large number of FWI simulations of materials with possible objects/defects in them.

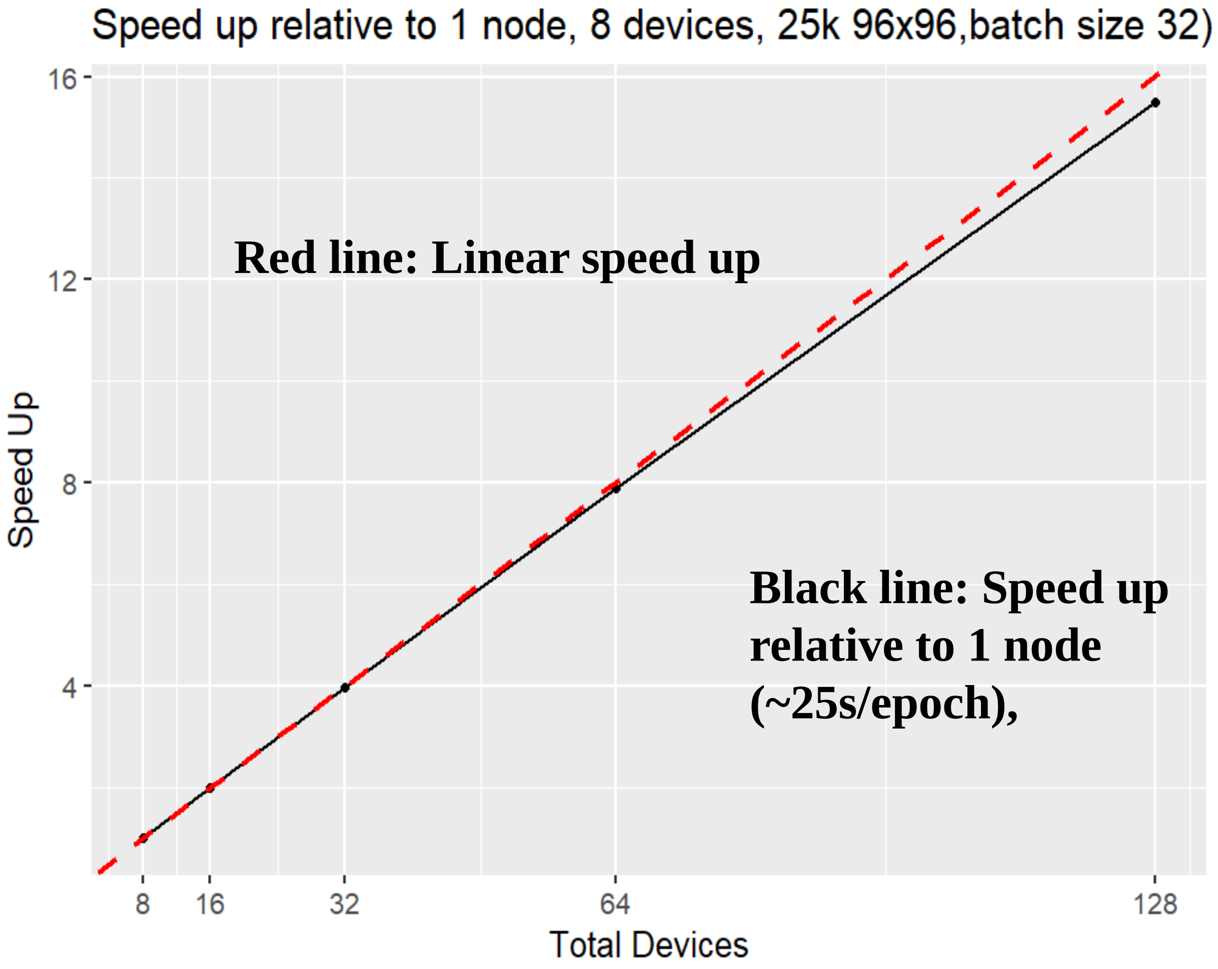
- A training data set of 25000 96x96 images were generated. The PI and colleagues developed a UNET deep learning model (44M parameters) that was written for a single GPU device execution in PyTorch



(Md Aktharuzzaman, Shoaib Anwar , Dmitry Borisov, Jing Rao, Jiaze He, 2D Numerical Ultrasound Computed Tomography For Elastic Material Properties In Metals, in Proceedings of the ASME 2022 International Mechanical Engineering Congress and Exposition IMECE2022 October 30-November 3, 2022, Columbus, Ohio).

Ultrasound computed tomography (USCT)

- Ported the code from single GPU to multiple-device/multiple-node execution, then port the model to Voyager by using the Habana guidelines almost directly
- We ran through a series of scaling tests with 25000 images per epoch, and tracked the seconds per epoch.



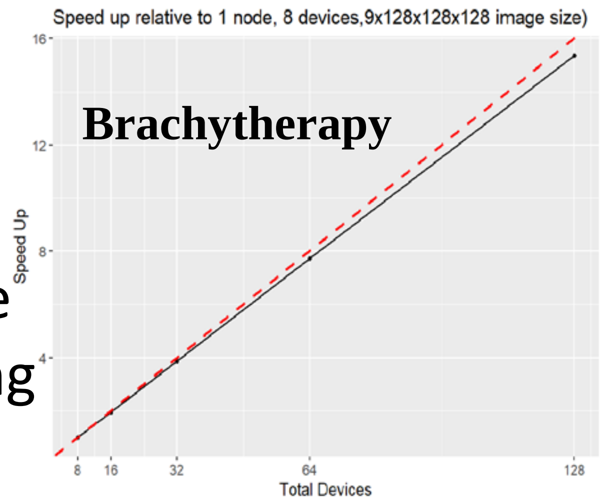
Red line: Linear speed up

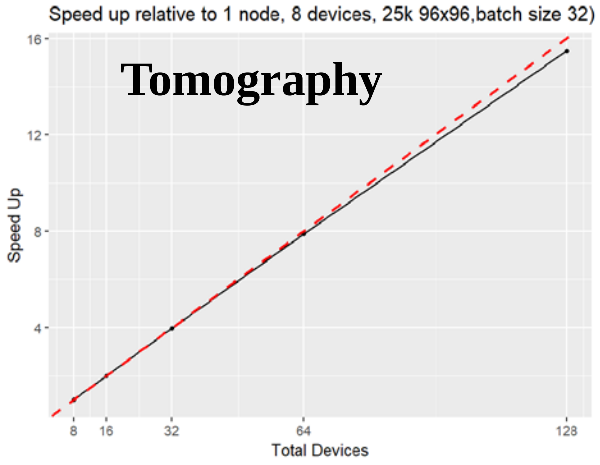
Black line: Speed up relative to 1 node (~25s/epoch),

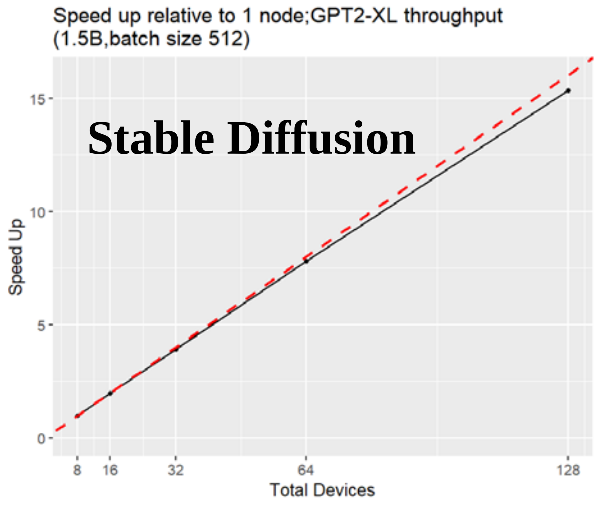
Quick Overview of Performance

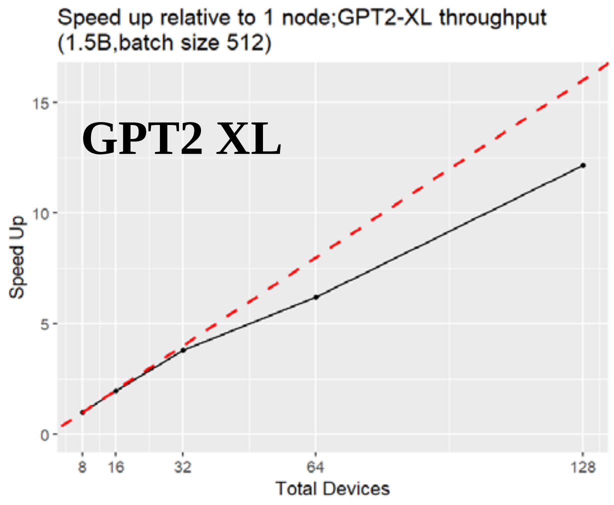
- The two UNET applications (<50M parameters) have similar scaling to Habana's test with ~1B parameter Stable Diffusion – between 8 to 16 nodes scaling drop off relative to linear

- The ~1.5B GPT2 XL tests show drop off after 4 to 8 nodes



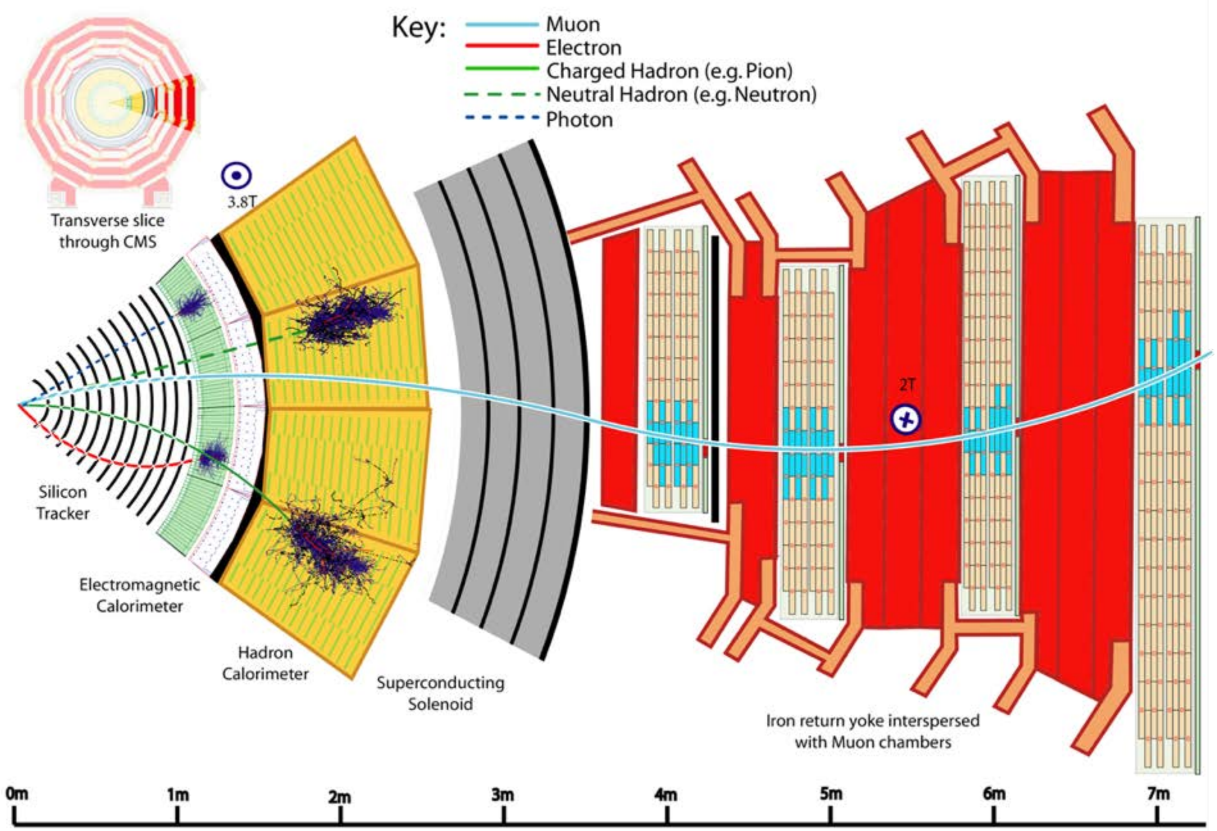






High energy physics – particle physics motivation

- **Voyager Co-PI Javier Duarte (UCSD)**
- Precise measurements of subatomic particles like the Higgs boson require enormous colliders and detectors, massive PB-scale datasets, and computationally intensive algorithms to reconstruct each particle collision
- AI algorithms and specialized AI processors can dramatically reduce the computational footprint of processing these datasets and improve the physics sensitivity
- Experiments at the CERN LHC like CMS are comprised of different subdetectors, each dedicated to measuring different types of particles.
- These complementary subdetectors are shown in Figure

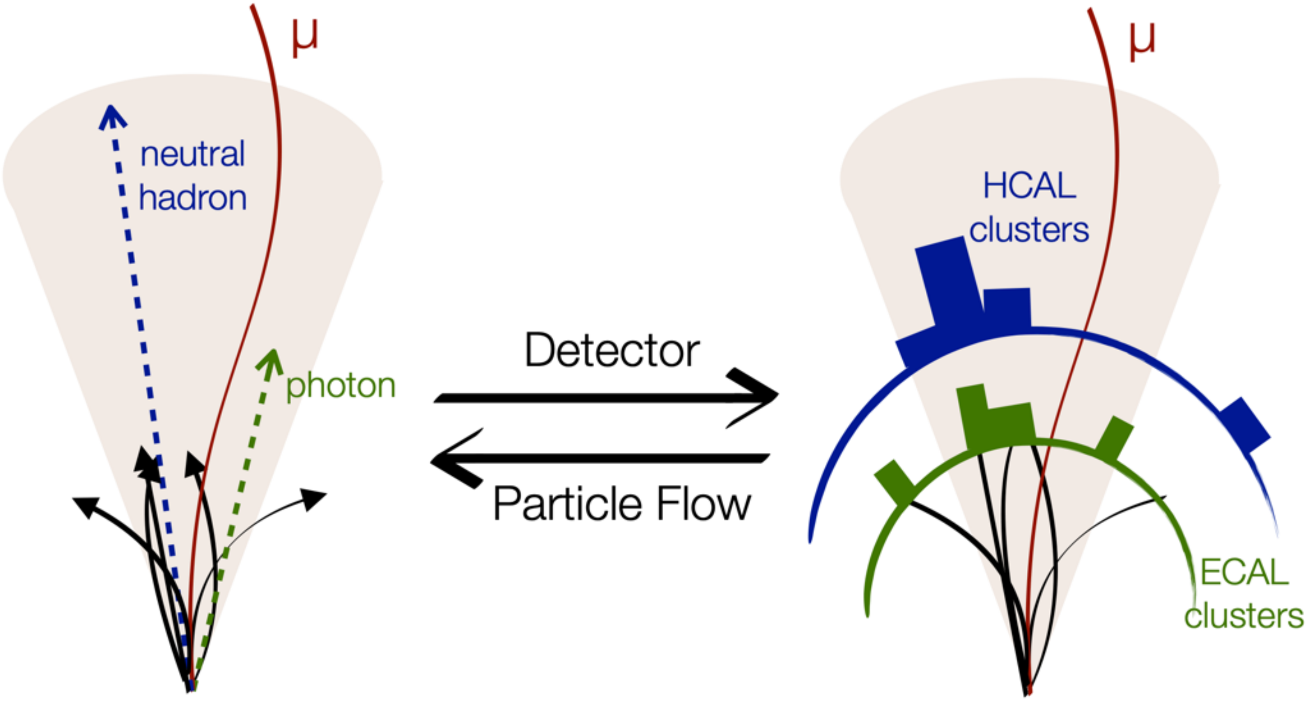


Key:
Muon
Electron
Charged Hadron (e.g. Pion)
Neutral Hadron (e.g. Neutron)
Photon
Transverse slice through CMS
3.8T
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with Muon chambers
0m
1m
2m
3m
4m
5m
6m
7m

A sketch of the specific particle interactions in a transverse slice of the CMS detector, from the beam interaction region to the muon detector. The muon and the charged pion are positively charged, and the electron is negatively charged.

High energy physics – particle-flow reconstruction

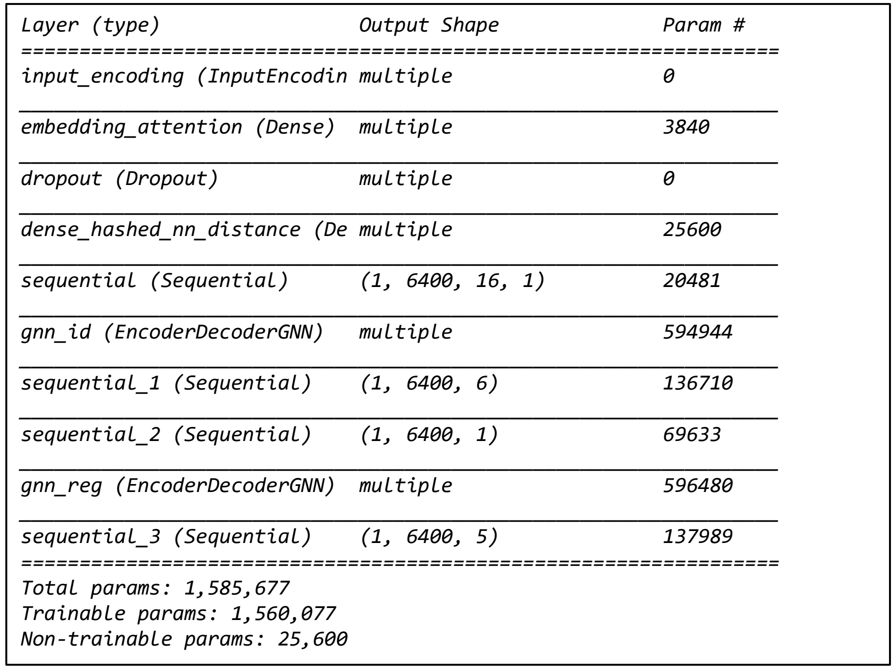
- During each proton-proton collision, outgoing particles interact with detector, leaving energy deposits in the calorimeters and tracks in the silicon tracker.
- An important HEP event reconstruction task is particle-flow (PF) reconstruction: the combination of information from complementary detector subsystems to produce a holistic, particle-level interpretation of the collision event.



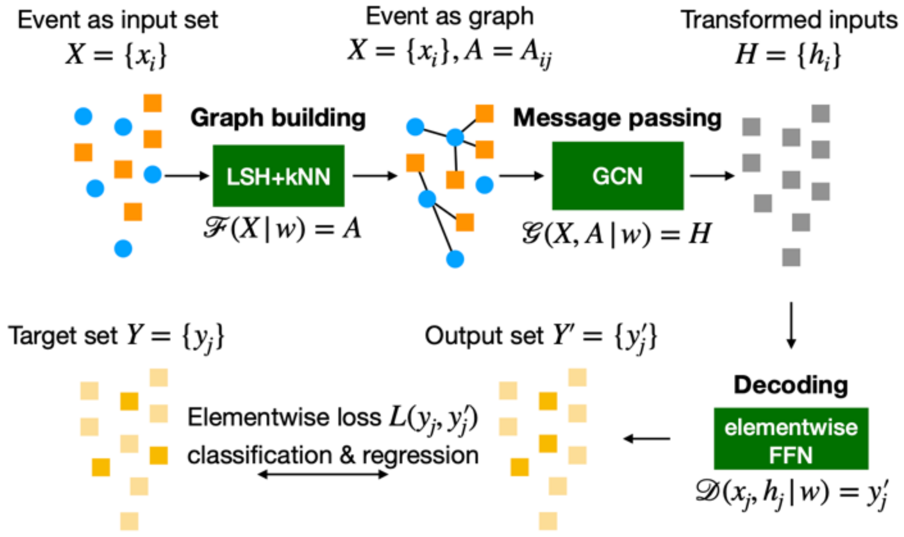
Particles leave tracks and clusters as they traverse the detector. These tracks and clusters can be used to infer the particles that created them through particle-flow reconstruction.

High energy physics – Machine-Learned Particle-Flow (MLPF)

- Developed Machine-Learned Particle-Flow (MLPF) algorithm based on Graph neural network (GNN) capable of inferring particles based on detector measurements
- GNN model implemented in TensorFlow v2.9.1
- This model can match or improve on the performance of standard PF reconstruction and is natively parallelizable and scales approximately linearly with the number of input detector elements.



```
Layer (type)                 Output Shape              Param #
=================================================================
input_encoding (InputEncodin multiple                  0
_________________________________________________________________
embedding_attention (Dense)  multiple                  3840
_________________________________________________________________
dropout (Dropout)            multiple                  0
_________________________________________________________________
dense_hashed_nn_distance (De multiple                  25600
_________________________________________________________________
sequential (Sequential)      (1, 6400, 16, 1)          20481
_________________________________________________________________
gnn_id (EncoderDecoderGNN)   multiple                  594944
_________________________________________________________________
sequential_1 (Sequential)    (1, 6400, 6)              136710
_________________________________________________________________
sequential_2 (Sequential)    (1, 6400, 1)              69633
_________________________________________________________________
gnn_reg (EncoderDecoderGNN)  multiple                  596480
_________________________________________________________________
sequential_3 (Sequential)    (1, 6400, 5)              137989
=================================================================
Total params: 1,585,677
Trainable params: 1,560,077
Non-trainable params: 25,600
```



Event as input set $X = \{x_i\}$

Event as graph $X = \{x_i\}, A = A_{ij}$

Transformed inputs $H = \{h_i\}$

Graph building

LSH+kNN

$\mathcal{F}(X \mid w) = A$

Message passing

GCN

$\mathcal{G}(X, A \mid w) = H$

Target set $Y = \{y_j\}$

Output set $Y' = \{y'_j\}$

Decoding

elementwise FFN

$\mathcal{D}(x_j, h_j \mid w) = y'_j$

Elementwise loss $L(y_j, y'_j)$

classification & regression

$x_i = [\text{type}, p_{\text{T}}, E_{\text{ECAL}}, E_{\text{HCAL}}, \eta, \phi, \eta_{\text{outer}}, \phi_{\text{outer}}, q, \ldots], \quad \text{type} \in \{\text{track}, \text{cluster}\}$

$y_j = [\text{PID}, p_{\text{T}}, E, \eta, \phi, q, \ldots], \quad \text{PID} \in \{\text{none}, \text{charged hadron}, \text{neutral hadron}, \gamma, e^{\pm}, \mu^{\pm}\}$

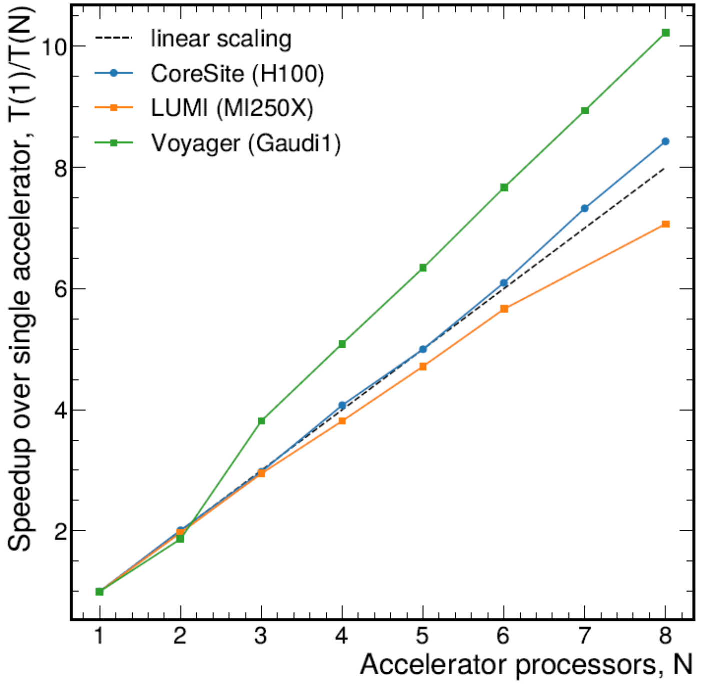
$h_i \in \mathbb{R}^{256}$

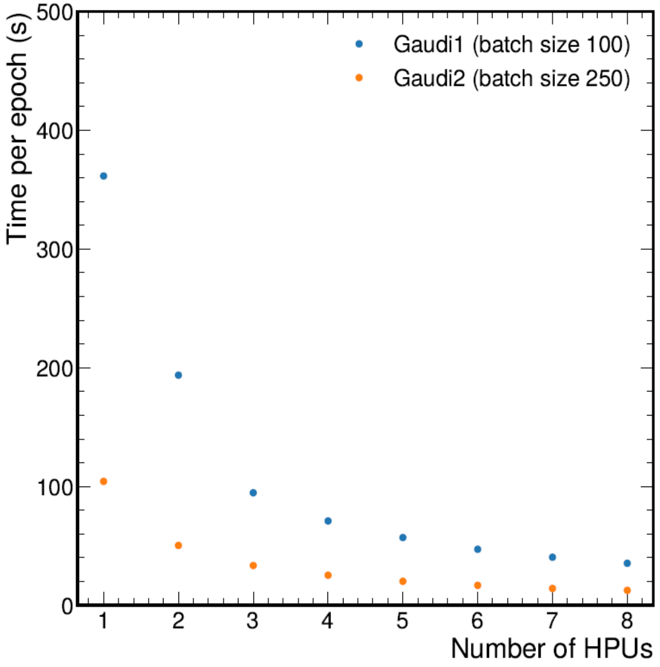
Trainable neural networks: $\mathcal{F}, \mathcal{G}, \mathcal{D}$

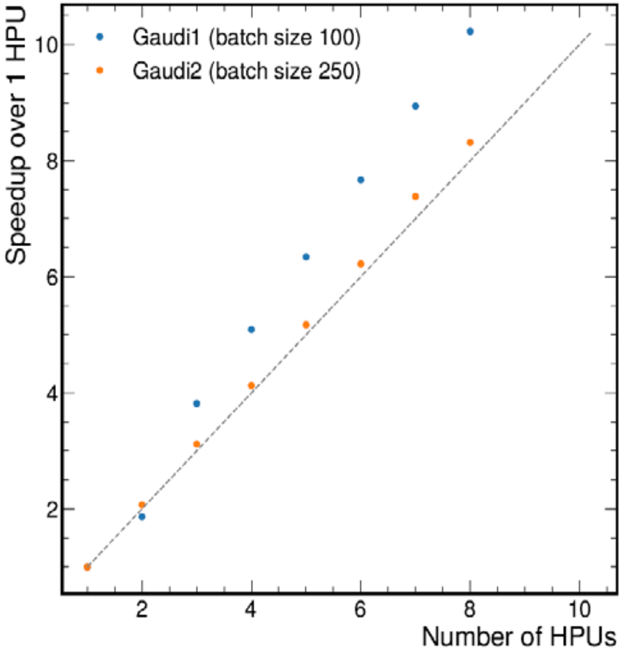
● - track, 🟧 - calorimeter cluster, ■ - encoded element

🟨 - target (predicted) particle, □ - no target (predicted) particle

MLPF performance results





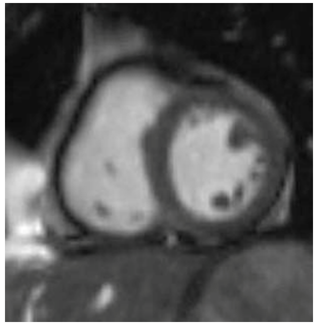


NVIDIA H100 GPUs from Flatiron Institute's CoreSite cluster , AMD MI250 GPUs from the LUMI supercomputer, and Intel Habana Gaudi HPUs from the Voyager supercomputer

Cardiac image analysis

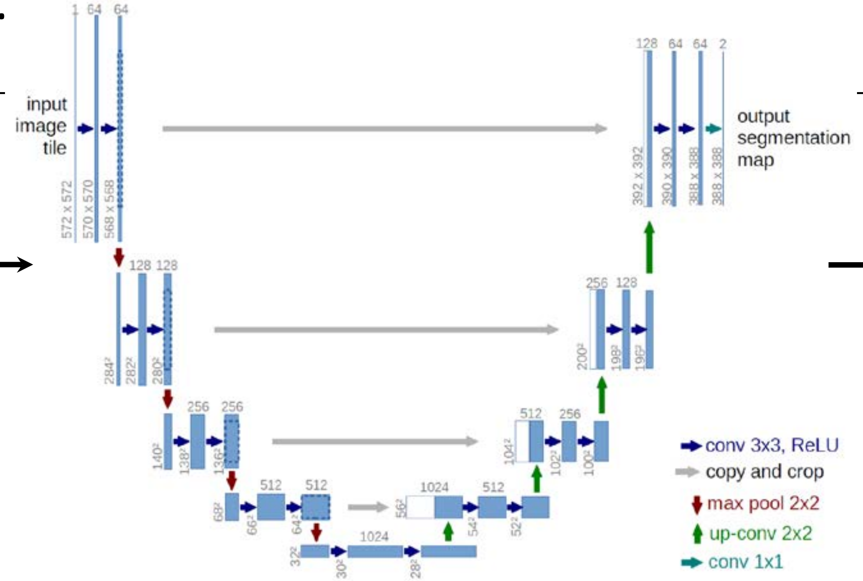
Mai Nguyen (SDSC, Voyager Co-PI) Garrison W. Cottrell (UCSD): Using *Voyager* to analyze cardiac image data. When done manually, this is a time-consuming and labor intensive process that can produce inconsistent results. The goal of Mai's work is to develop a deep learning approach that provides automated, efficient and consistent segmentation of cardiac structures from MRI, thereby aiding the non-invasive detection of cardiac disease. Porting to Voyager:

- The training code for this application has been successfully ported to Voyager Gaudi training nodes.
- Habana First Generation Inference Processors: model conversion from TensorFlow to ONNX and model compilation successfully completed. Plan to compare execution times and accuracy for training and inference to GPU.



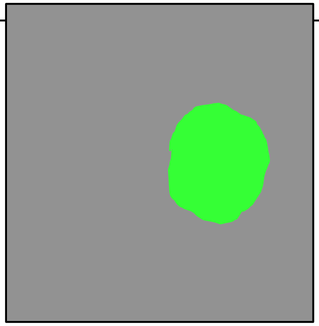
Cardiac MRI Frame

→



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

→



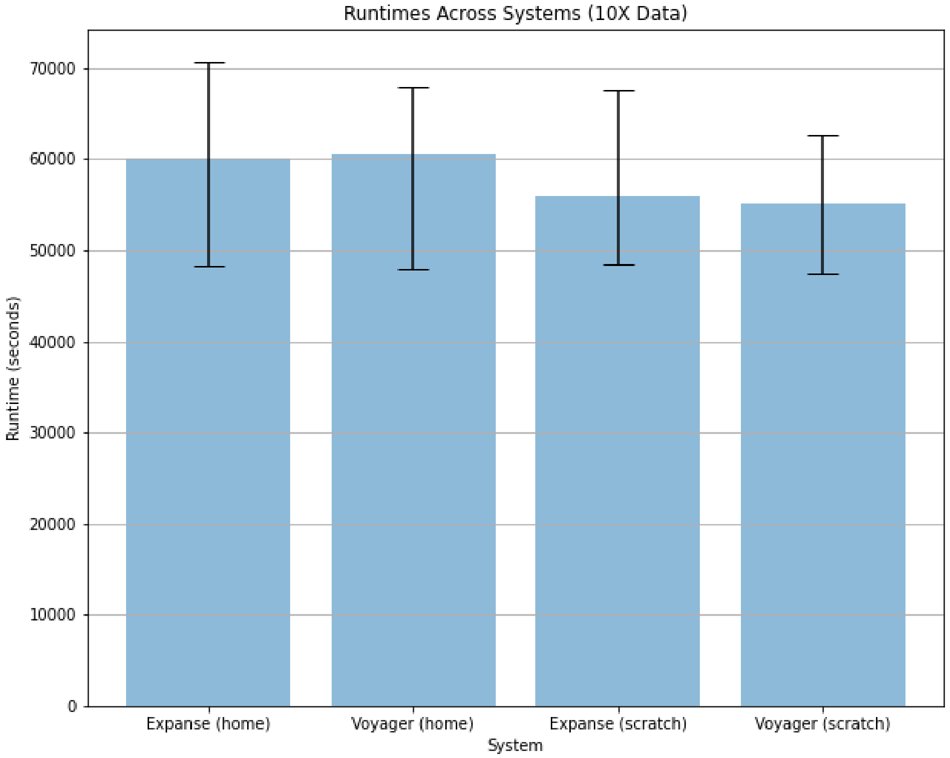
Segmented Left Ventricle Endocardium

Model uses a U-Net architecture implemented in TensorFlow to segment the left ventricle from a cardiac MRI.

M. H. Nguyen, M. Bobar, D. Uys, G. W. Cottrell, I. Altintas, IEEE 14th International Conference on e-Science (2018) arXiv:1909.08028

Voyager Gaudi and V100 results

- Researchers have performed a comparison study to evaluate training performance of *Voyager*’s Gaudi cards vs. general-purpose GPUs
- The dataset for this application consists of CMR images for multiple patients and is approximately 1 GB in size
- This dataset was duplicated ten times to create a 10 GB dataset, which was used for the analysis
- Figure shows the execution times for Gaudi vs. V100 GPU for different file systems



Summary

- *Voyager's* AI-focused architecture is an excellent resource for exploring AI for science and engineering. In testbed phase of operations since May 2022.
- Several training and inference applications running on *Voyager.* So far, our experience that code changes are minimal in most cases
- The system was designed to support exploration in multiple dimensions (Gaudi and First-Gen Inference processors, RoCE interconnect, 400 GbE switch, Kubernetes, Ceph et al)
- Considering the amount of innovation, the acquisition, deployment and installation was remarkably smooth with issues resolved jointly with Supermicro and Habana experts
- Excellent partnership between SDSC, Habana/Intel, Supermicro and the research community to gain a deep understanding of how AI/ML applications perform and what work is needed to optimize them for these kind of specialized architectures
- We are gaining important knowledge and lessons in areas such as Kubernetes, AI model porting and will be followed by performance, and scaling
- Engaging the broader community (academia, national labs, industry) via training and workshops

Voyager would not be possible without a dedicated team of professionals and experts

Rommie Amaro
Haisong Cai*
Trevor Cooper
Chris Cox*
Javier Duarte
Tom Hutton*
Christopher Irving*
Marty Kandes
Amit Majumdar
Tim McNew*
Dmitry Mishin
Mai Nguyen

Susan Rathbun
Paul Rodriguez
Scott Sakai
Manu Shantharam
Robert Sinkovits
Fernando Silva*
Shawn Strande
Tom Tate*
Mahidhar Tatineni
Mary Thomas
Cindy Wong
Nicole Wolter

Supermicro Team
Habana/Intel Gaudi Team
Arista

NSF Award 2005369

*A special thanks to the HPC Systems Group and the Data Center staff who have been performing onsite work under COVID restrictions