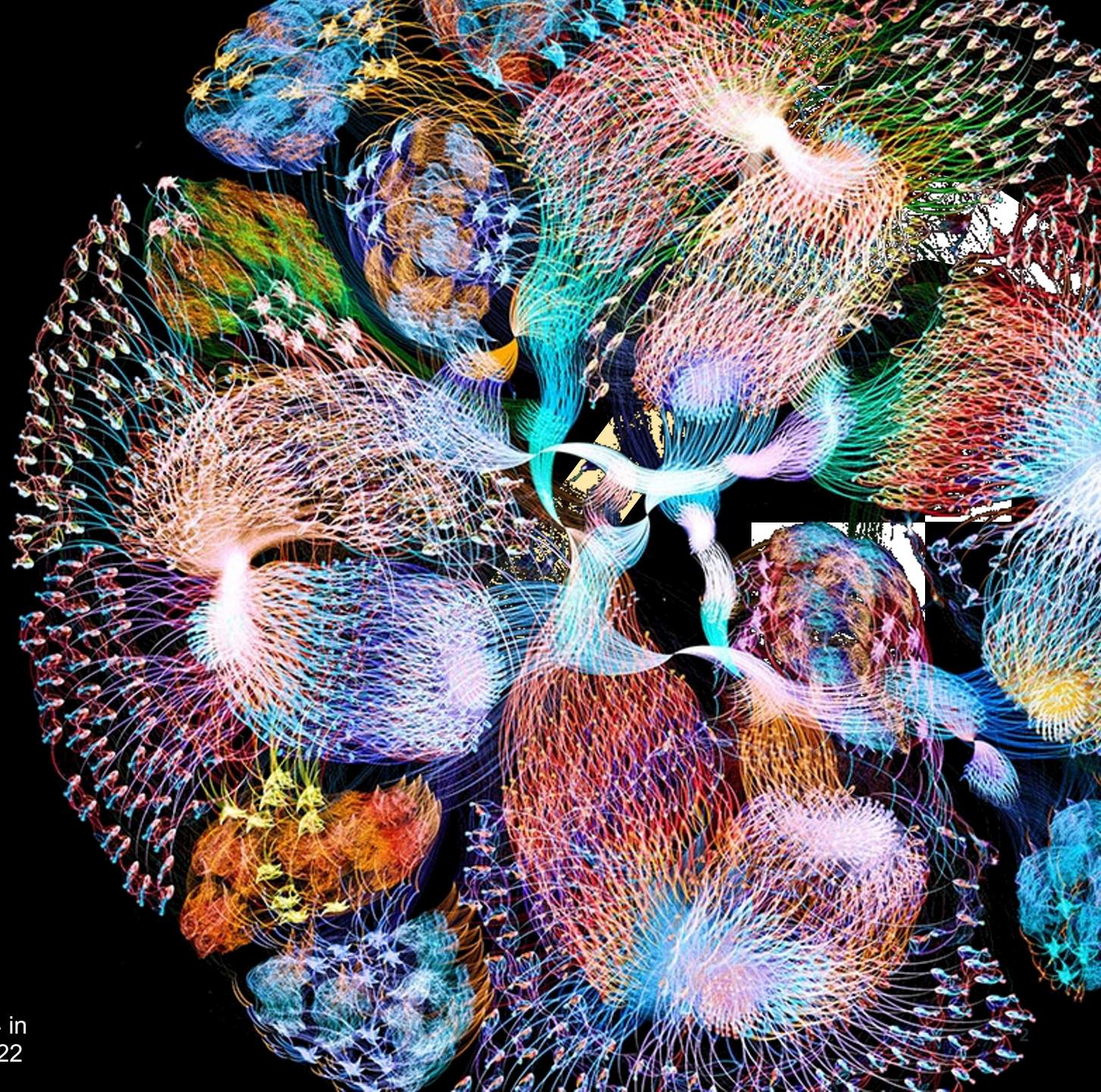
Graphcore IPUs: Accelerating Argonne's AI/ML Applications

GRAPHCORE

TOPICS

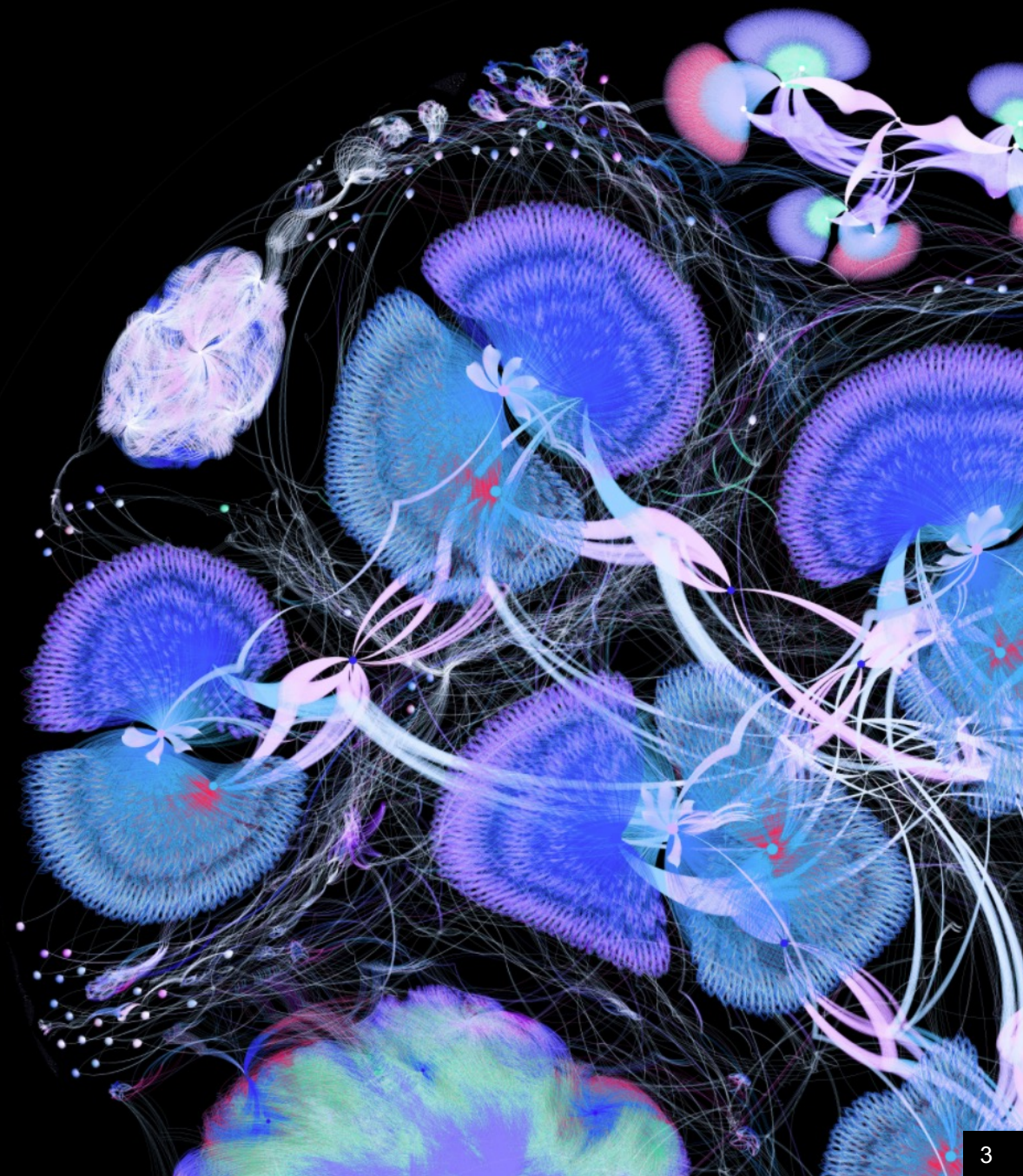
- Graphcore Introduction
  - Company
  - Processor Architecture (IPU)
  - System Architecture
  - Software & Enablement
- General Benchmarks
- Customer Case Studies
- Argonne Applications
  - UNO
  - BraggNN
- “Good” Computer



GRAPHCORE

Note: Upgrade from MK1 to Bow POD64 in ALCF AI Testbed in October/November ‘22

GRAPHCORE



GRAPHCORE ENABLING MACHINE INTELLIGENCE

- Founded in 2016
- Technology: Intelligence Processor Unit (IPU)
- Team: 650+ globally
- Offices: UK, US, China, Norway, Poland
- Raised >$730M

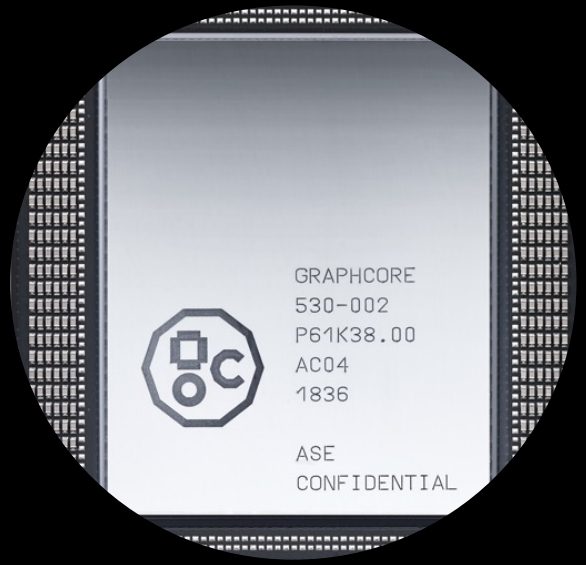
SEQUOIA · ATOMICO · SOFINA · Microsoft

BMW i VENTURES · DELL · BOSCH · SAMSUNG · Merian GLOBAL INVESTORS

Amadeus Capital Partners · pitango VENTURE CAPITAL · draperesprit · Foundation CAPITAL

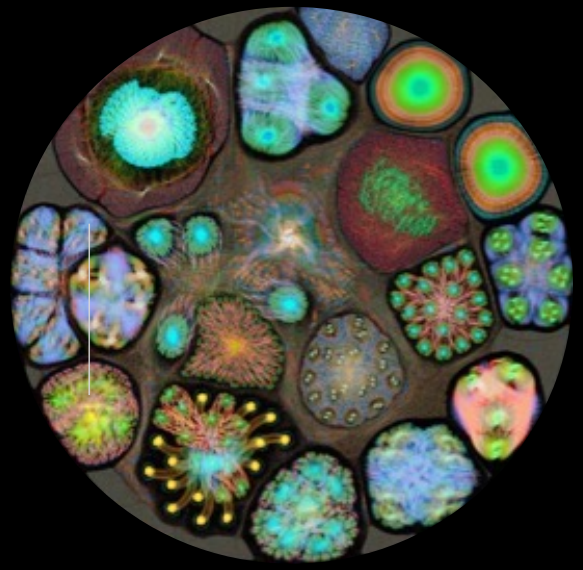
ABOUT US...

Hardware



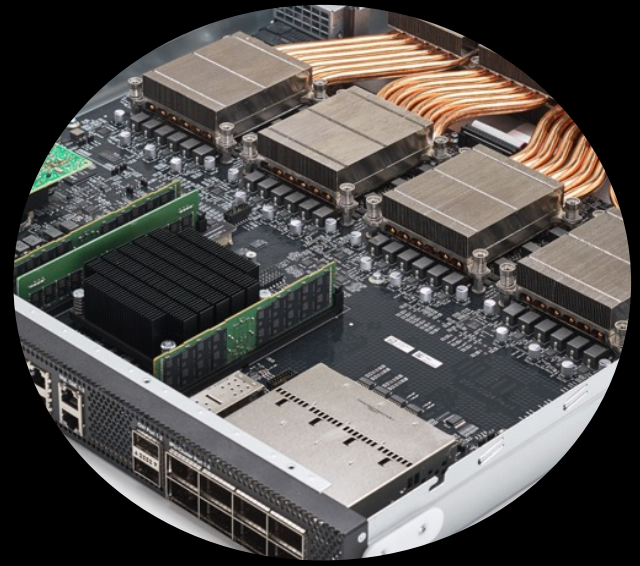
IPU processors designed for AI

Software



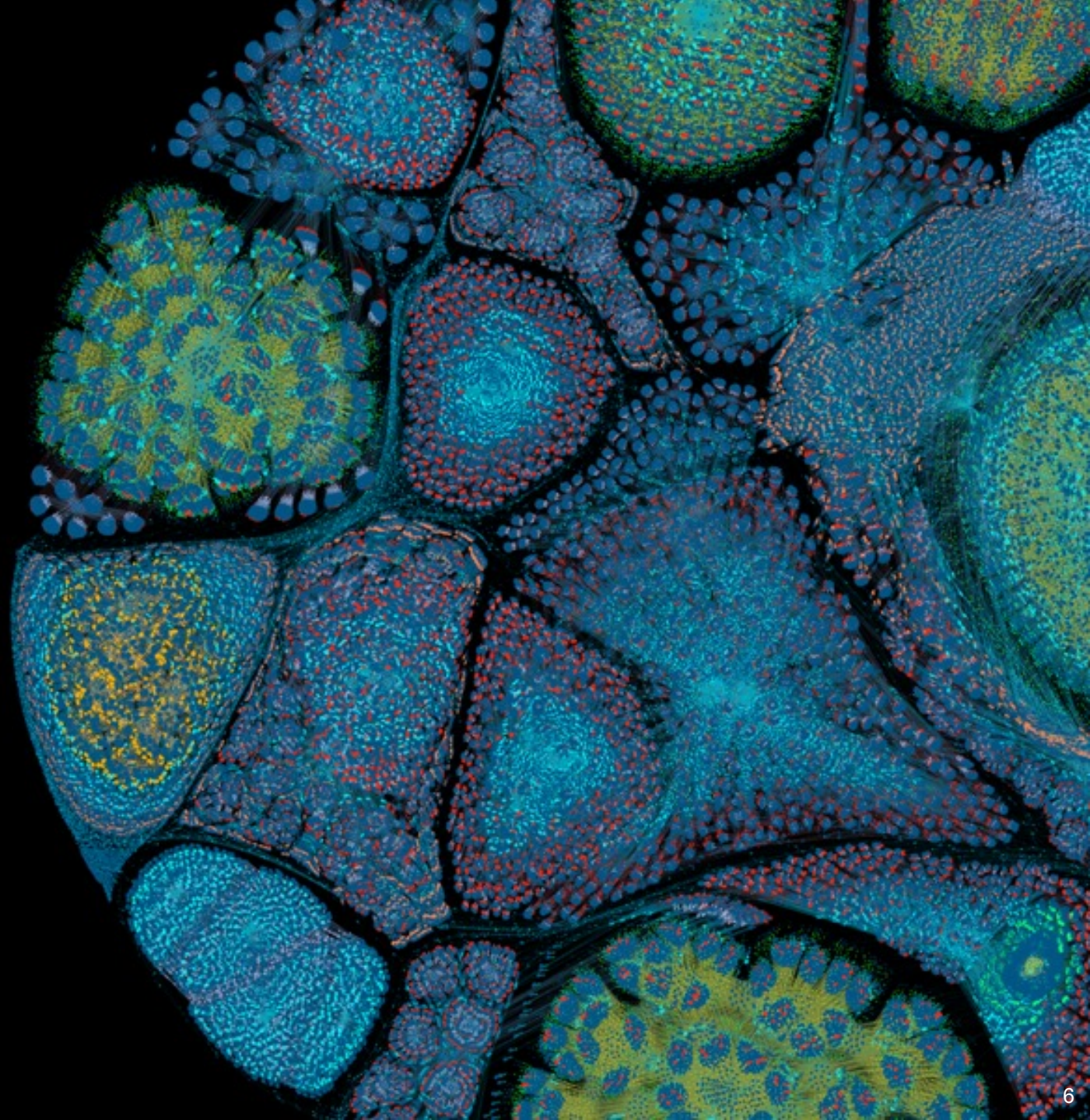
Poplar® software stack & development tools

Systems



BOW-2000 and Server IPU-POD$_{64}$ scale-out

PROCESSOR ARCHITECTURE



INTRODUCING THE BOW IPU
WORLD'S FIRST 3D WAFER-ON-WAFER PROCESSOR

3D silicon wafer stacked processor

350 TeraFLOPS AI compute

Optimized silicon power delivery

0.9 GigaByte In-Processor-Memory @ 65TB/s

1,472 independent processor cores

8,832 independent parallel programs

10x IPU-Links™ delivering 320GB/s

GRAPHCORE

BOW IPU PROCESSOR

Deep Trench Capacitor

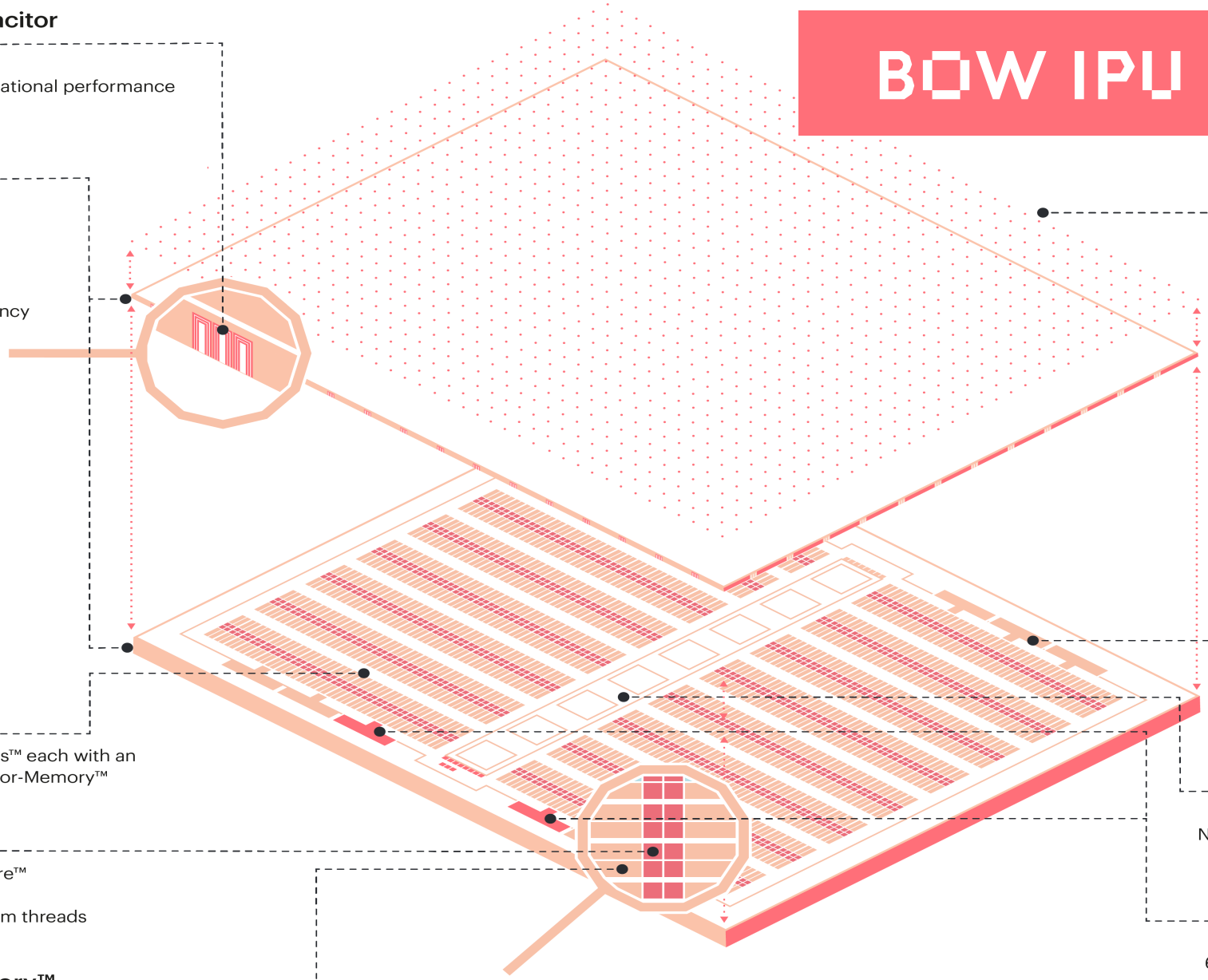
Efficient power delivery
Enables increase in operational performance

Wafer-On-Wafer

Advanced silicon 3D stacking technology

Closely coupled power delivery die

Higher operating frequency and enhanced overall performance



Solder Bumps

IPU-Links™

10x IPU-Links,
320GB/s chip to chip bandwidth

IPU-Exchange™

11 TB/s all to all IPU-Exchange™
Non-blocking, any communication pattern

PCIe

PCI Gen4 x16
64 GB/s bidirectional bandwidth to host

IPU-Tiles™

1472 independent IPU-Tiles™ each with an IPU-Core™ and In-Processor-Memory™

IPU-Core™

1472 independent IPU-Core™

8832 independent program threads executing in parallel

In-Processor-Memory™

900MB In-Processor-Memory™ per IPU

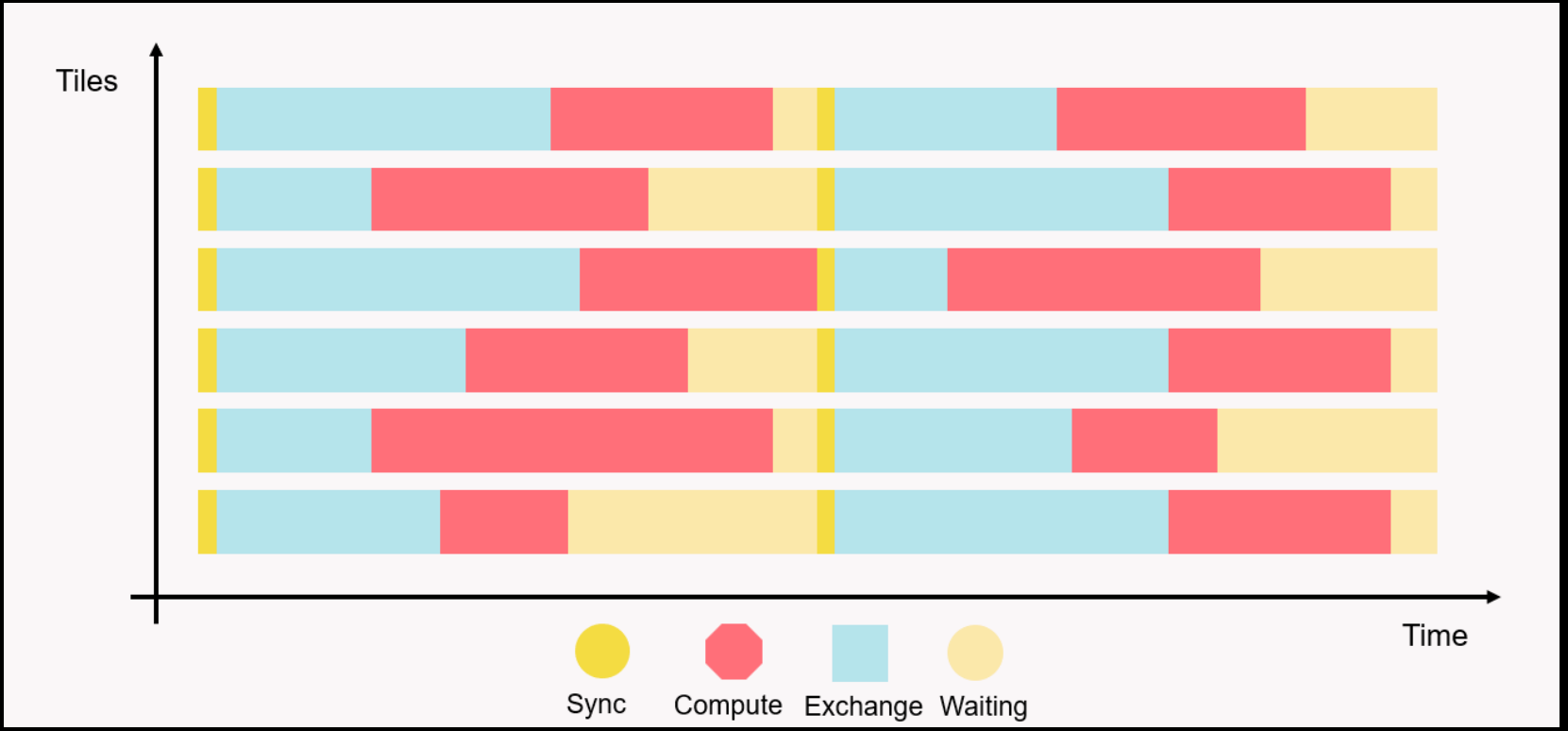
65.4TB/s memory bandwidth per IPU

BULK SYNCHRONOUS PARALLEL (BSP)

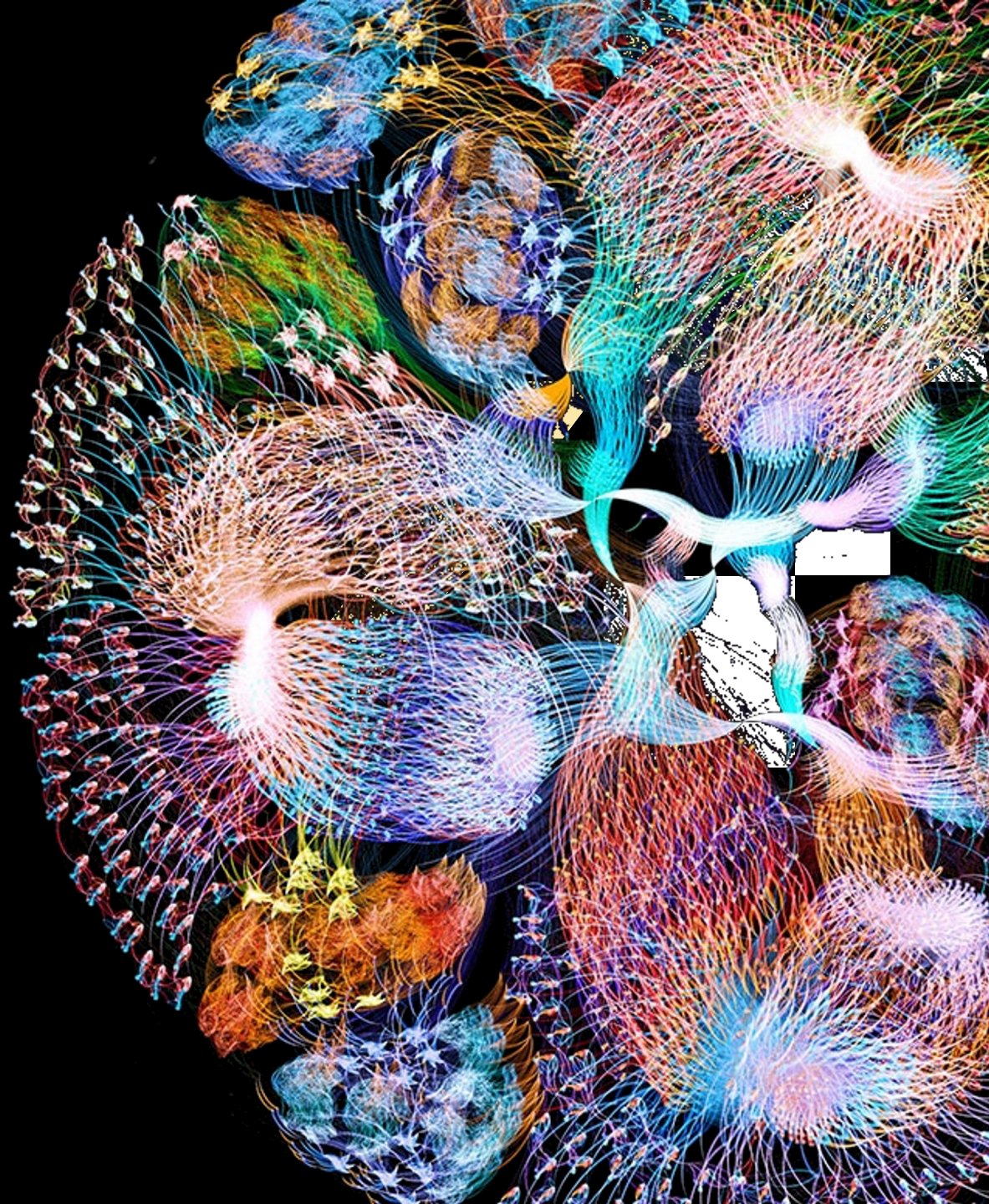
"Bridging Model" for parallel computation

Repeat Supercycles composed of 3 phases
- Compute – execution from local memory
- Sync – global barrier synchronization
- Exchange – global communication

Graphcore: Sync, Exchange, Compute
- Globally - Exchange immediately follows Sync
- Each Tile - Compute immediately follows Exchange



SYSTEM ARCHITECTURE



BOW-2000 IPU MACHINE

4 x Bow 3D Wafer-on-Wafer IPUs

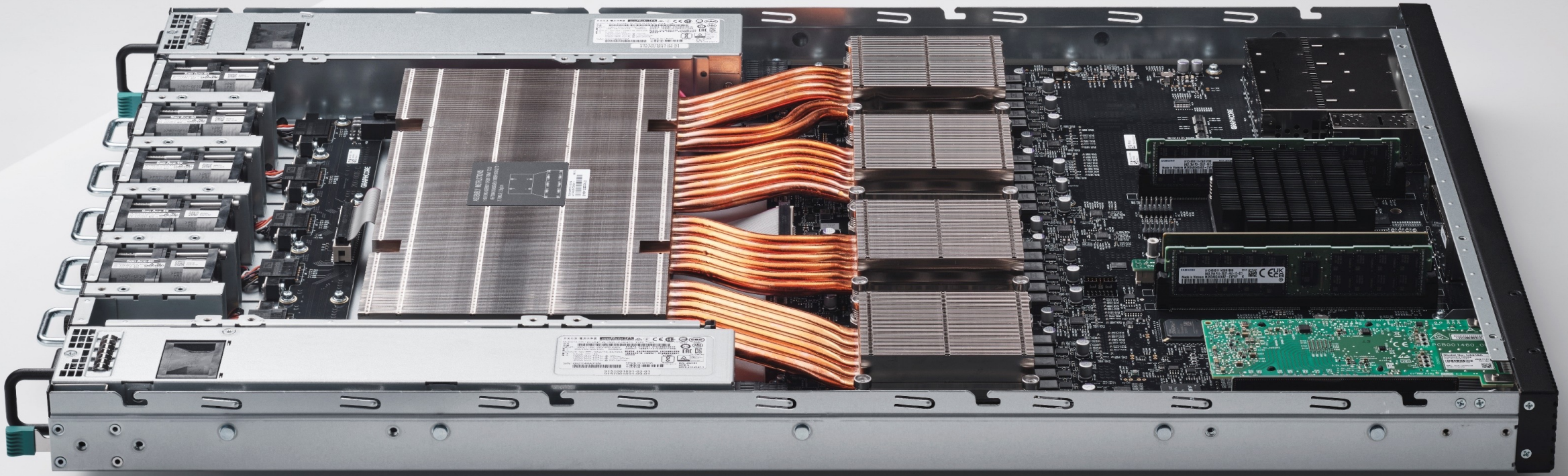
1.4 PetaFLOPS AI Compute

3.6 GB In-Processor-Memory @ 260TB/s

Up to 256 GB IPU Streaming Memory

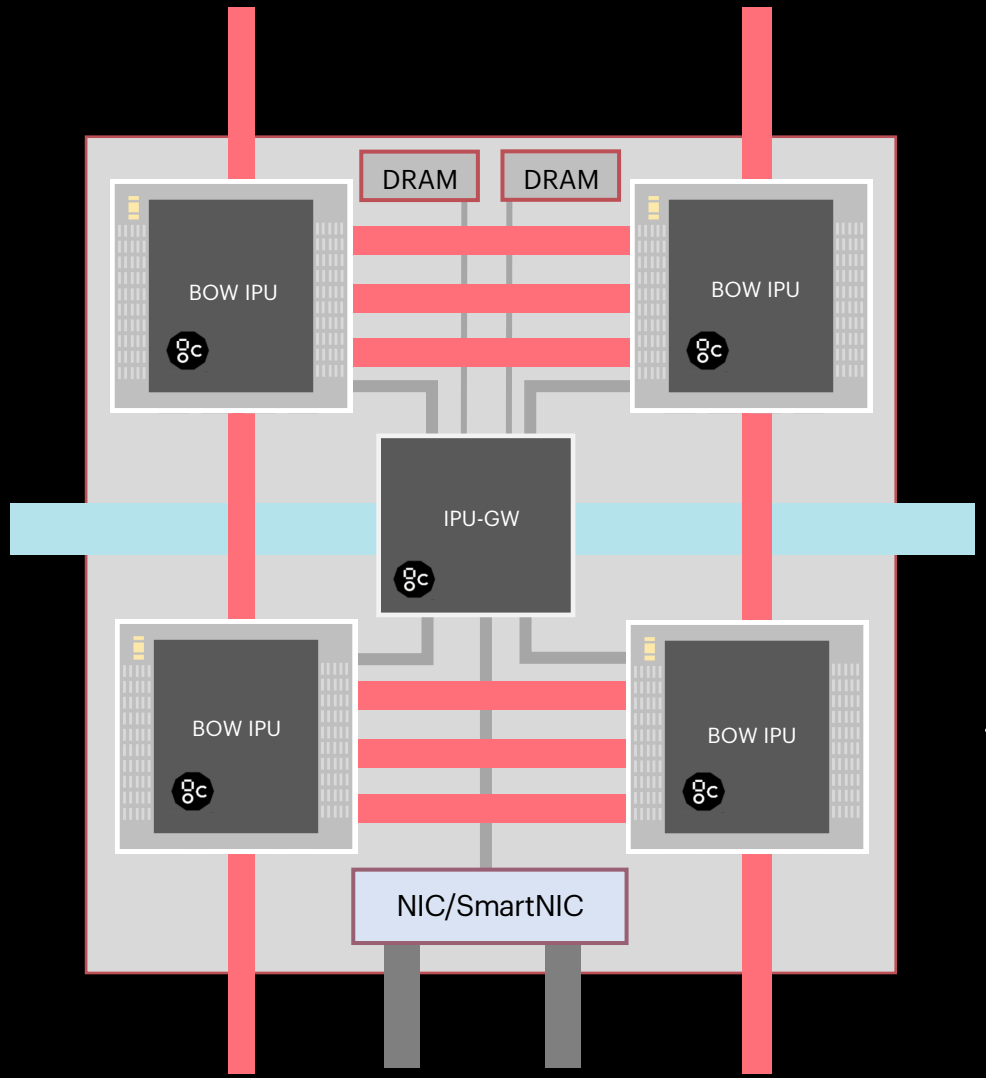
2.8 Tbps IPU-Fabric™

1U blade form factor



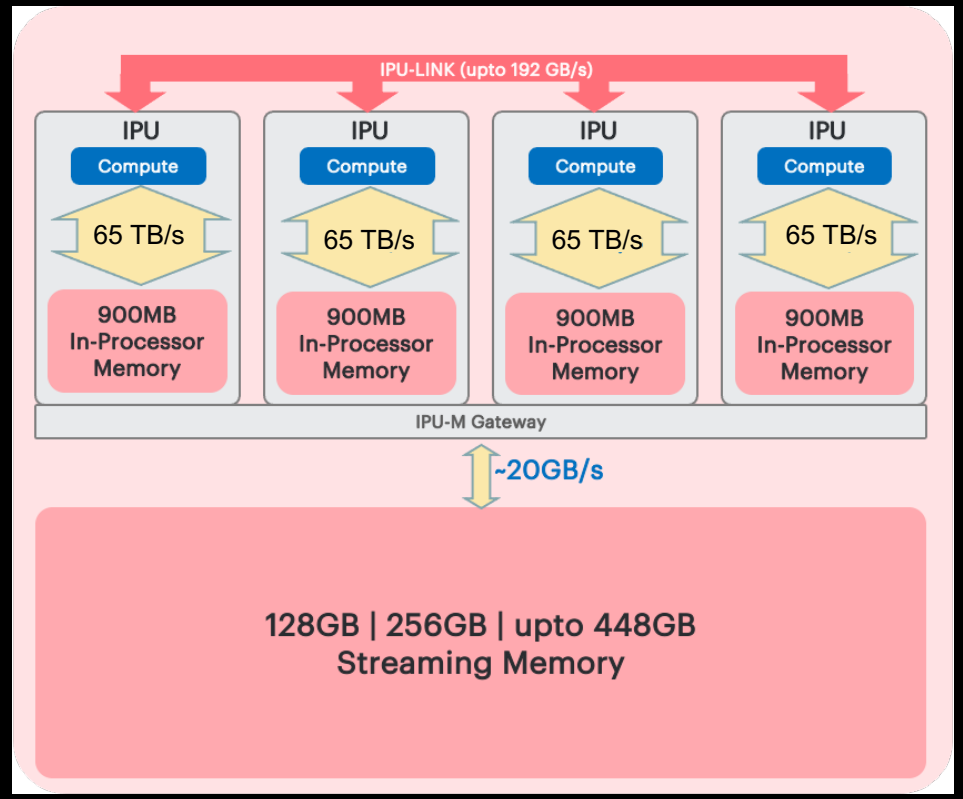
GRAPHCORE

BOW-2000 IPU MACHINE



3.6GB ultra-fast memory for models/weights/ activations

...supplemented by Streaming Memory for more flexibility in model sizes, optimiser configurations, & parallelisation schemes



x16 IPU-Link 64GB/s

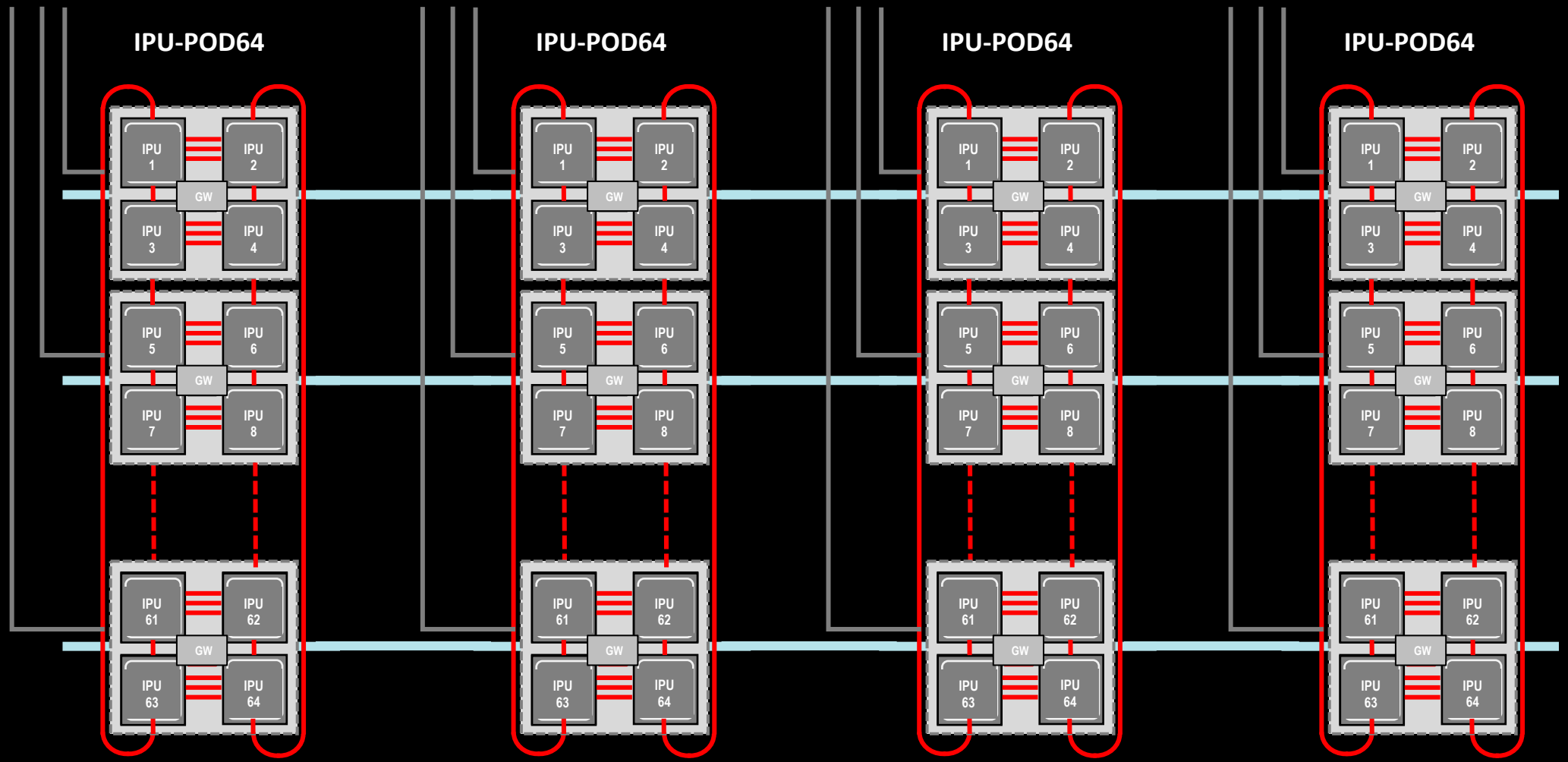
100GE Host-Link Network I/F

100Gbps IPU-GW Link

x8 PCIe G4 32GB/s

GRAPHCORE

IPU-FABRIC FOR SCALE-OUT



GRAPHCORE

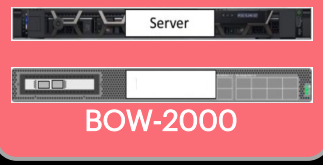
BOW-2000/IPU-POD RECONFIGURABILITY

Reusable, simple migration between platforms

Disaggregated Server enables configurable Host Server /IPU ratio depending on workload for optimized TCO

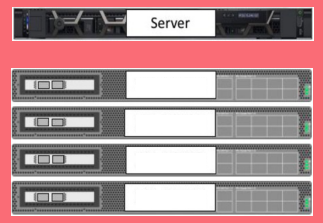
BOW-POD4 DA
x1 BOW-2000

Host Server



+ BOW-2000

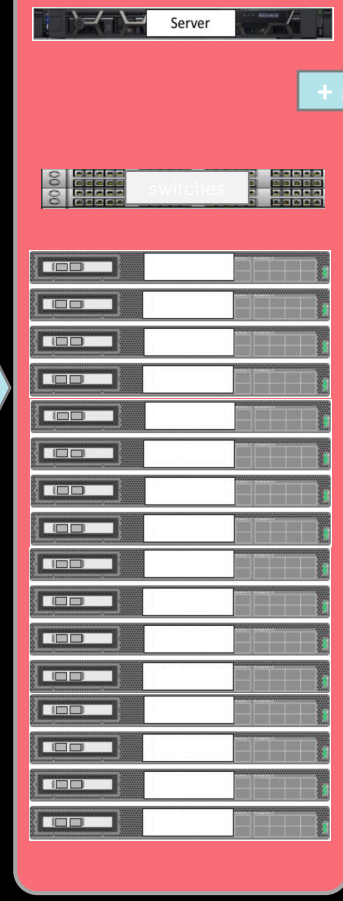
BOW-POD16 DA
x4 BOW-2000
(direct attach)



Switched version adds TOR switch & Mgmt switch

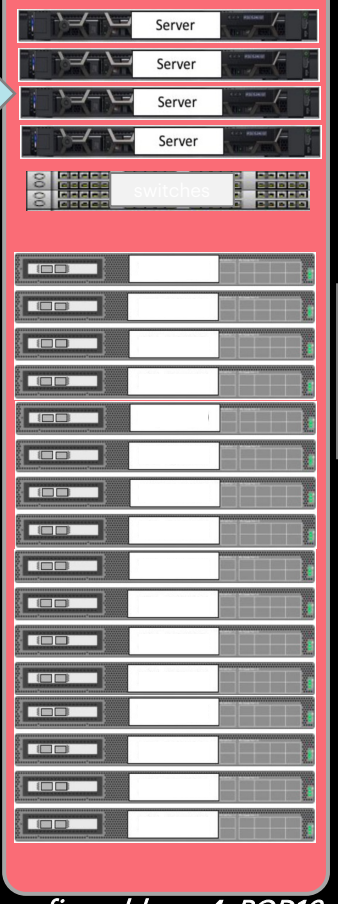
+ servers
+ switches
+ BOW-2000
re-cable

BOW-POD64
1 host server



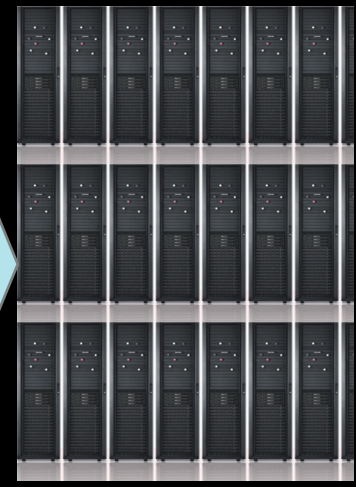
+ servers

BOW-POD64
4 host server



reconfigurable as 4xPOD16

scaleout



GRAPHCORE

1x BOW-2000 | 4x BOW-2000 | 16x BOW-2000 | Up to 16Kx BOW-2000

BOW: 3RD GENERATION IPU SYSTEMS

SHIPPING TO CUSTOMERS TODAY

BOW POD16

4x Bow-2000
5.6 PetaFLOPS
1 CPU server

BOW POD32

8x Bow-2000
11.2 PetaFLOPS
1 CPU server

BOW POD64

16x Bow-2000
22.4 PetaFLOPS
1-4 CPU server(s)

BOW POD256

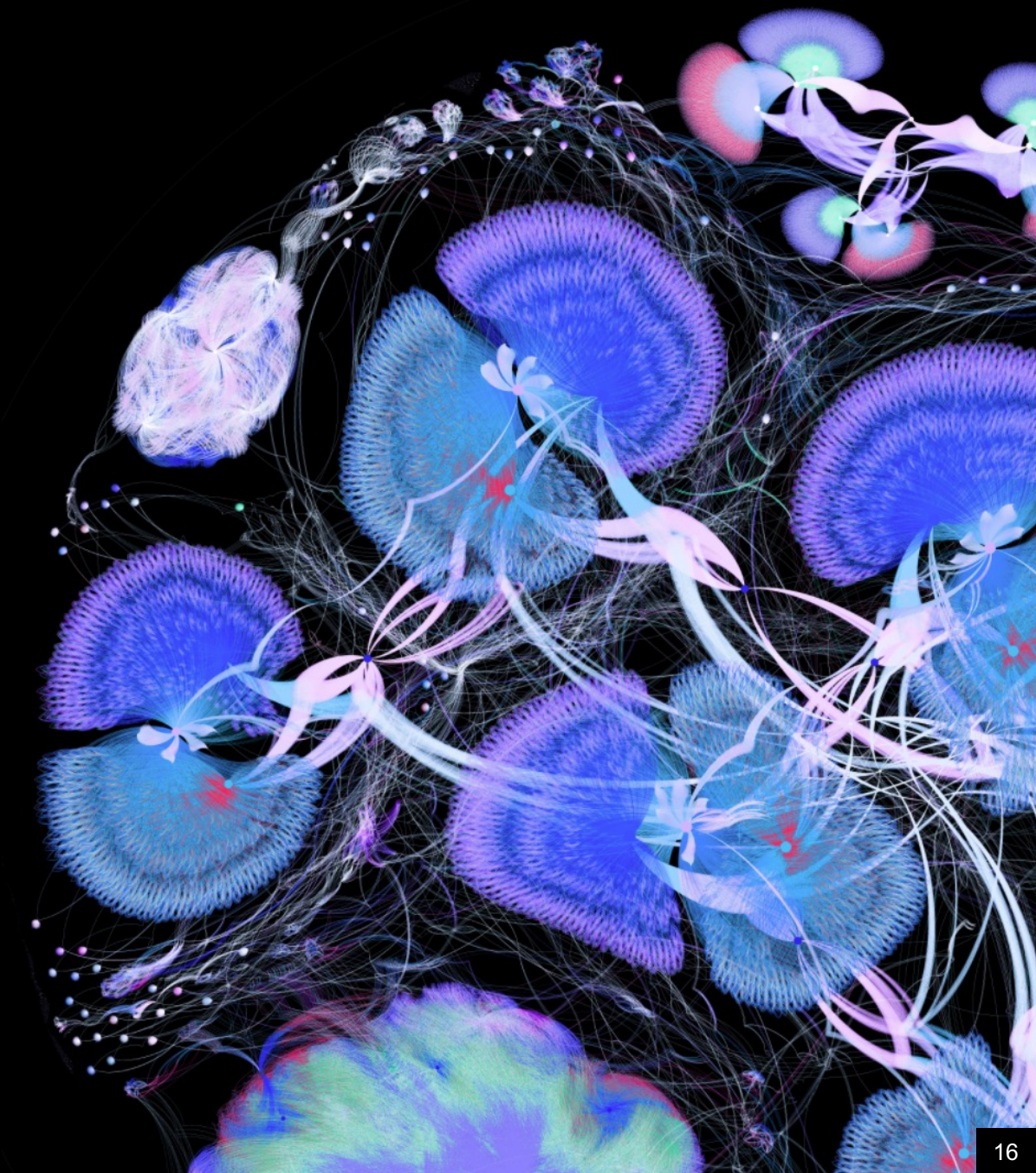
64x Bow-2000
89.6 PetaFLOPS
4-16 CPU server(s)

BOW POD1024

256x Bow-2000
358.4 PetaFLOPS
16 - 64 CPU server(s)
Early access

GRAPHCORE

SOFTWARE & ENABLEMENT



GRAPHCORE SOFTWARE MATURITY

ML APPLICATIONS

- NLP/TRANSFORMERS
- IMAGE CLASSIFICATION/CNNS
- OBJECT DETECTION
- LARGE MODELS
- MLPERF
- CONDITIONAL SPARSITY
- GNNS

DEVELOPER ECOSYSTEM

- TUTORIALS
- CODE EXAMPLES
- DOCUMENTATION
- VIDEOS
- NATIVE IPU CODERS PROGRAM
- APPS PORTFOLIO

POPLAR® SDK

FRONTENDS

- JUPYTER NOTEBOOKS
- INFERENCE DEPLOYMENT TOOLKIT

FRAMEWORKS

- ONNX
- HALO
- Keras
- PaddlePaddle

FW BACKENDS

- XLA
- POPART+
- POPDIST

POPLAR®

- PARTITIONER
- POPIR
- POPIT
- POPLIBS
- GCL
- POPLAR
- GRAPH ENGINE
- GRAPH COMPILER

DRIVERS

- GC DEVICE ACCESS LAYER
- IPUOF DRIVER
- PCIe DRIVER

POPVISION TOOLS

- GRAPH ANALYZER
- SYSTEM ANALYZER
- DEBUGGER
- DEVELOPMENT ENVIRONMENT

SYSTEM SOFTWARE

- V-IPU

SYSTEM MONITORING

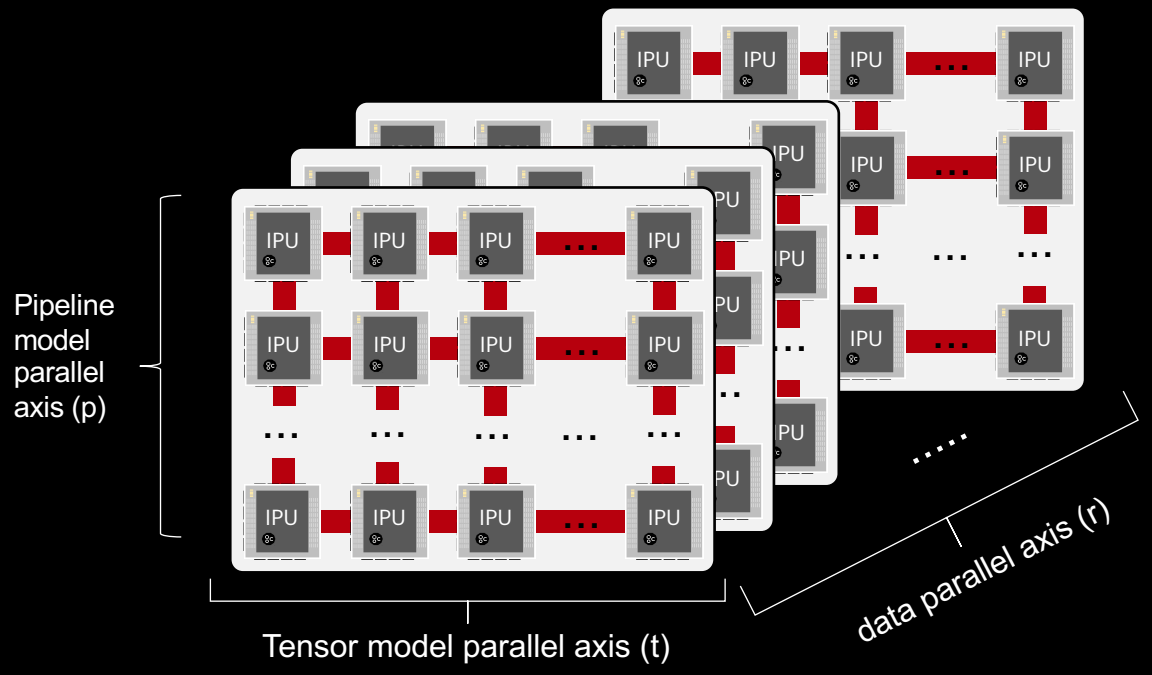
- PROMETHEUS
- GRAFANA

JOB DEPLOYMENT

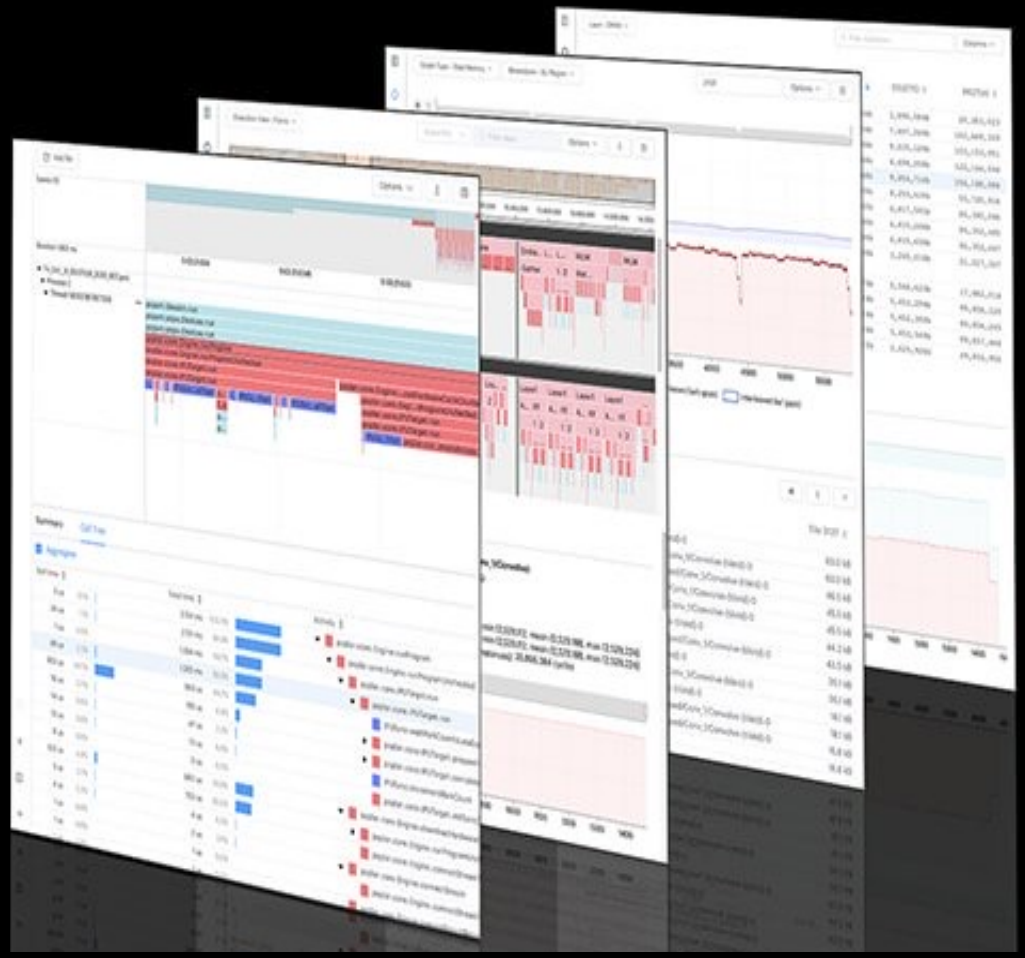
- K8S
- SLURM

GRAPHCORE

USEFUL SW CAPABILITIES



Multi-IPU Capabilities



Debug Tools


gpu_cnn_keras.py ↔ ipu_cnn_keras.py tf_keras
Keras
GPU
IPU
as tf
ras.layers import *
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()
x_train = x_train.astype('float32') / 255.0
y_train = tf.keras.utils.to_categorical(y_train, 10)
ds_train = tf.data.Dataset.from_tensor_slices((x_train, y_train)).batch(64, drop_remainde
model = tf.keras.Sequential([
Conv2D(32, (3, 3), padding='same', input_shape=x_train.shape[1:]),
Activation('relu'),
Conv2D(32, (3, 3)),
Activation('relu'),
MaxPooling2D(pool_size=(2, 2)),
Dropout(0.25),
Conv2D(64, (3, 3), padding='same'),
Activation('relu'),
Conv2D(32, (3, 3)),
Activation('relu'),
MaxPooling2D(pool_size=(2, 2)),
Dropout(0.25),
Flatten(),
Dense(512),
Activation('relu'),
Dropout(0.5),
Dense(10),
Activation('softmax')
])
model.compile(loss='categorical_crossentropy',
optimizer=tf.optimizers.SGD(learning_rate=0.016),
metrics=['accuracy'])
model.fit(ds_train, epochs=40)
import tensorflow as tf
from tensorflow.keras.layers import *
from tensorflow.python import ipu
cfg = ipu.config.IPUConfig()
cfg.auto_select_ipus = 1
cfg.configure_ipu_system()
with ipu.ipu_strategy.IPUStrategy().scope():
(x_train, y_train), (x_test, y_test) = tf.keras.datasets.cifar10.load_data()
x_train = x_train.astype('float32') / 255.0
y_train = tf.keras.utils.to_categorical(y_train, 10)
ds_train = tf.data.Dataset.from_tensor_slices((x_train, y_train)).batch(64, drop_rema
model = tf.keras.Sequential([
Conv2D(32, (3, 3), padding='same', input_shape=x_train.shape[1:]),
Activation('relu'),
Conv2D(32, (3, 3)),
Activation('relu'),
MaxPooling2D(pool_size=(2, 2)),
Dropout(0.25),
Conv2D(64, (3, 3), padding='same'),
Activation('relu'),
Conv2D(32, (3, 3)),
Activation('relu'),
MaxPooling2D(pool_size=(2, 2)),
Dropout(0.25),
Flatten(),
Dense(512),
Activation('relu'),
Dropout(0.5),
Dense(10),
Activation('softmax')
])
model.compile(loss='categorical_crossentropy',
optimizer=tf.optimizers.SGD(learning_rate=0.016),
metrics=['accuracy'])
model.fit(ds_train, epochs=40)

MODEL GARDEN ML DOMAIN COVERAGE

COMPUTER VISION

IMAGE CLASSIFICATION
- ResNet50 v1.5
- EfficientNet-B0
- EfficientNet-B4
- ResNeXt-101
- MobileNet v2
- MobileNet v3
- ViT
- DINO
- SWIN

OBJECT DETECTION
- YOLO v3
- YOLO v4
- Faster RCNN
- EfficientDet

OBJECT SEGMENTATION
- Unet (Industrial)
- Unet (Medical)

GNN
- TGN
- MPNN
- Cluster-GCN NEW

AI FOR SIMULATION
- DeeMPD NEW
- DeepDriveMD NEW
- ETO
- CosmoFlow NEW
- ABC Covid-19 NEW

REINFORCEMENT
- RL

Reinforcement Learning

PROBABILISTIC
- MCMC
- VAE

NLP
- BERT-Base
- BERT-Large
- GroupBERT
- PackedBERT
- GPT2
- RoBERTa
- Deberta
- BART
- T5
- Hubert

SPEECH

STT (ASR)
- RNN-T
- Conformer

TTS
- DeepVoice3
- FastSpeech2
- FastPitch

RECOMMENDER
- Autoencoder
- DIN
- DIEN

MULTIMODAL
- LXMERT
- CLIP
- Mini DALL-E

OTHER
- Sales Forecast
- Neural Image Fields

TensorFlow

PyTorch

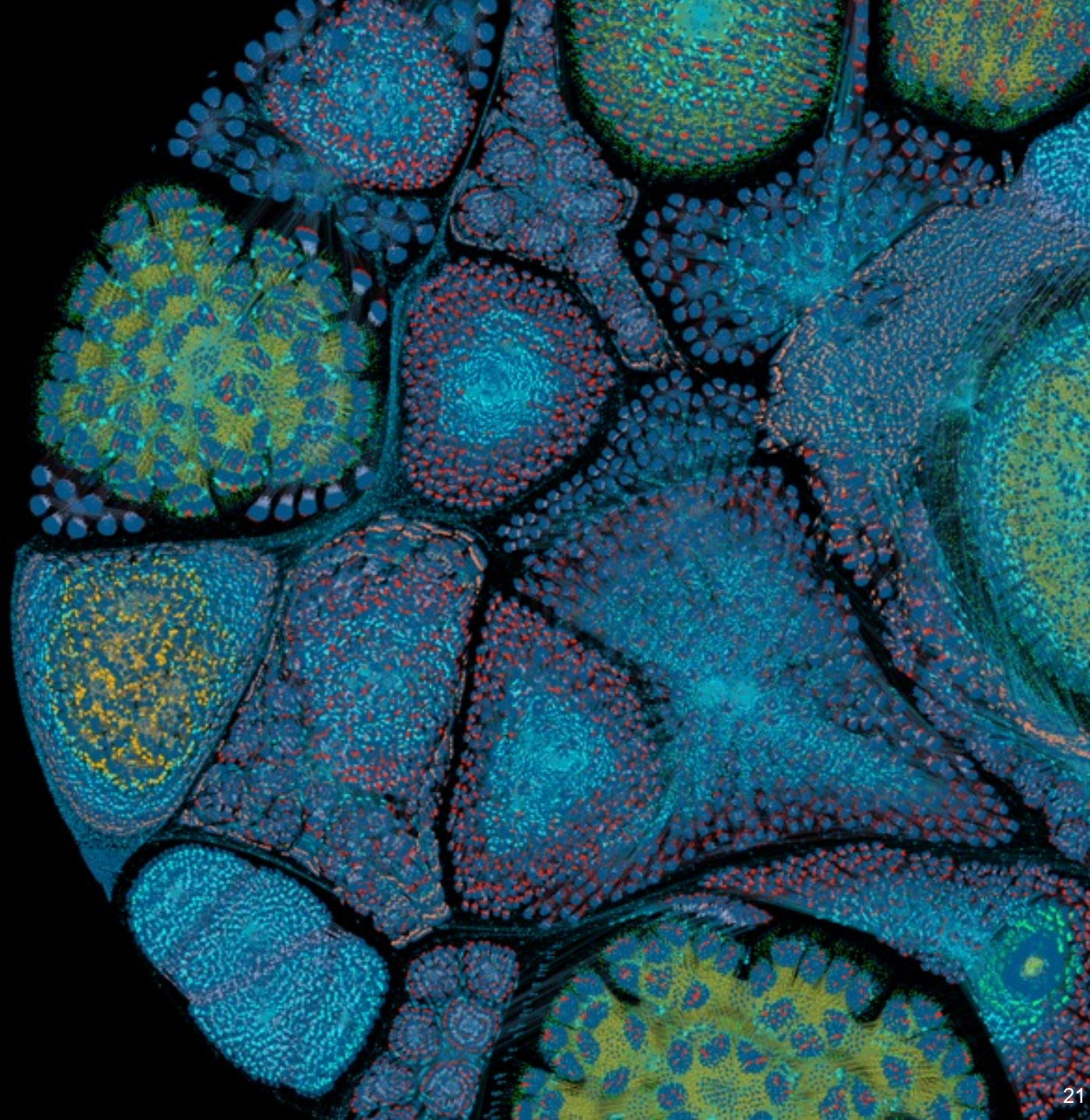
Hugging Face

Keras

PaddlePaddle

POPART

GENERAL BENCHMARKS

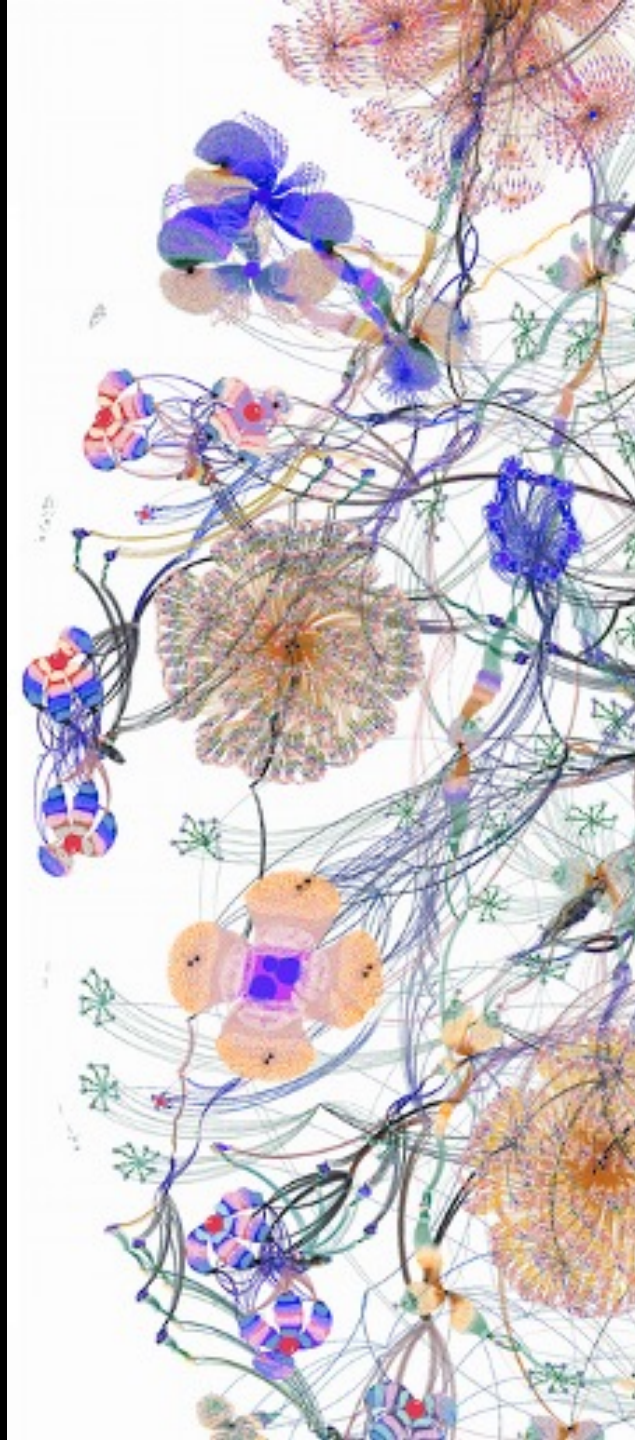


PERFORMANCE RESULTS
MLPERF & BENCHMARKS

We publish our MLPerf & benchmark performance on multiple models, platforms, & frameworks

- Published performance results with config/parameters
- Allows customers to do their own analysis/comparison

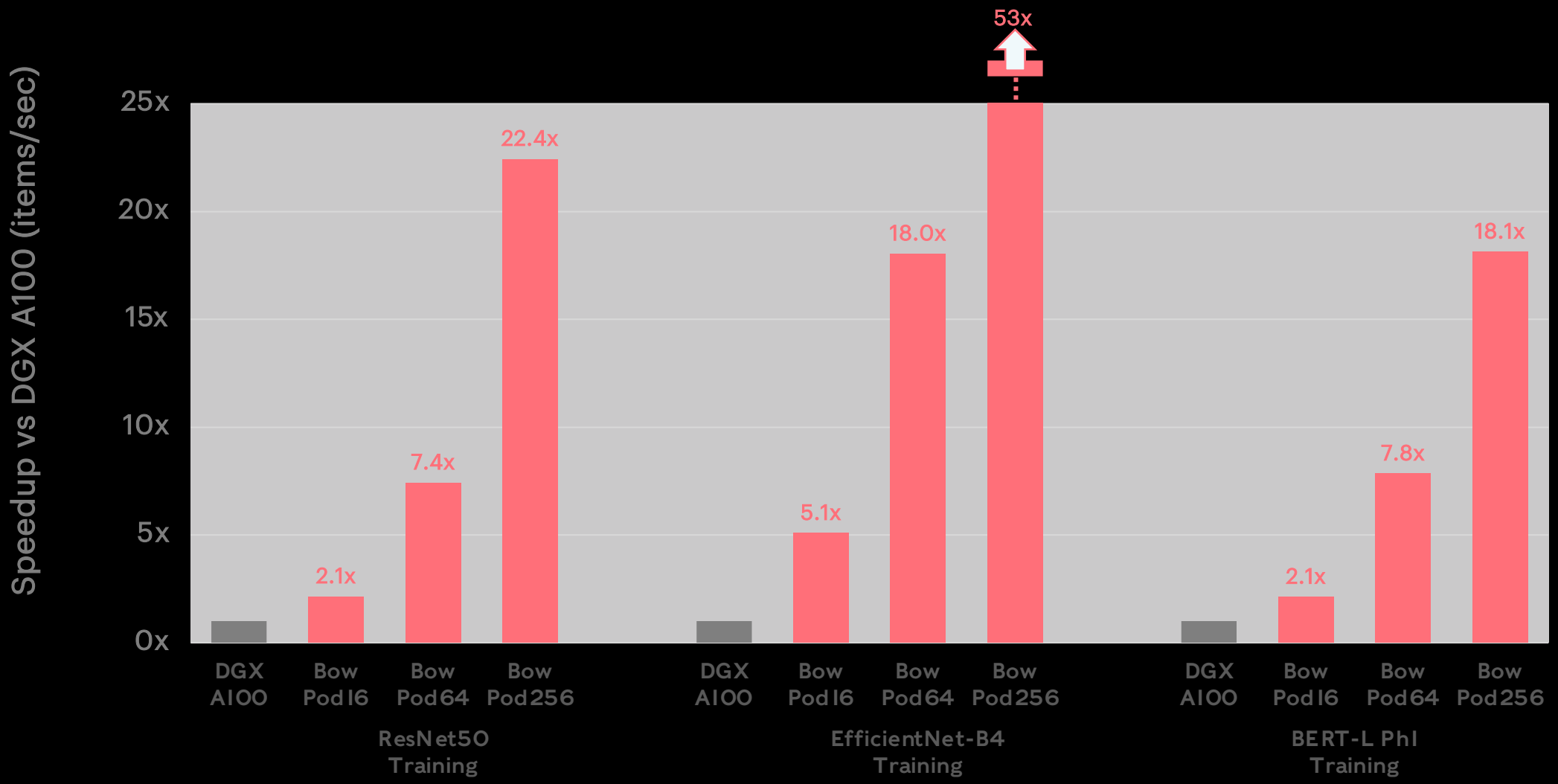
Model	Variant	Platform	SDK Version	Framework	Dataset	Batch Size	Precision	Throughput (items/sec)
ResNet-50 v1.5		Bow Pod16	SDK2.6	TensorFlow1	ImageNet2012	3,520	16.16	44,047
ResNet-50 v1.5		Bow Pod16	SDK2.6	PyTorch	ImageNet2012	16,384	16.16	37,219
ResNet-50 v1.5		Bow Pod64	SDK2.6	TensorFlow1	ImageNet2012	5,120	16.16	152,522
ResNet-50 v1.5		Bow Pod128	SDK2.6	TensorFlow1	ImageNet2012	5,120	16.16	0
ResNet-50 v1.5		Bow Pod256	SDK2.6	TensorFlow1	ImageNet2012	10,240	16.16	450,701
EfficientNet-B4	G16-EfficientNet	Bow Pod16	SDK2.6	TensorFlow1	ImageNet2012	6,144	16.16	8,992
EfficientNet-B4	G16-EfficientNet	Bow Pod16	SDK2.6	PyTorch	ImageNet2012	1,024	16.32	6,111
EfficientNet-B4	G16-EfficientNet	Bow Pod64	SDK2.6	TensorFlow1	ImageNet2012	6,144	16.16	35,247
EfficientNet-B4	G16-EfficientNet	Bow Pod256	SDK2.6	TensorFlow1	ImageNet2012	6,144	16.16	117,240
ResNeXt101		Bow Pod16	SDK2.6	TensorFlow1	ImageNet2012	768	16.16	12,233
ViT	Pre-Training	Bow Pod16	SDK2.6	PyTorch	ImageNet1k	65,536	16.16	7,512
ViT	Pre-Training	Bow Pod64	SDK2.6	PyTorch	ImageNet1k	65,536	16.16	27,981
ViT	Fine-Tuning	Bow Pod16	SDK2.6	PyTorch	ImageNet1k	2,040	16.16	7,977



https://www.graphcore.ai/performance-results

BOW POD ADVANTAGE

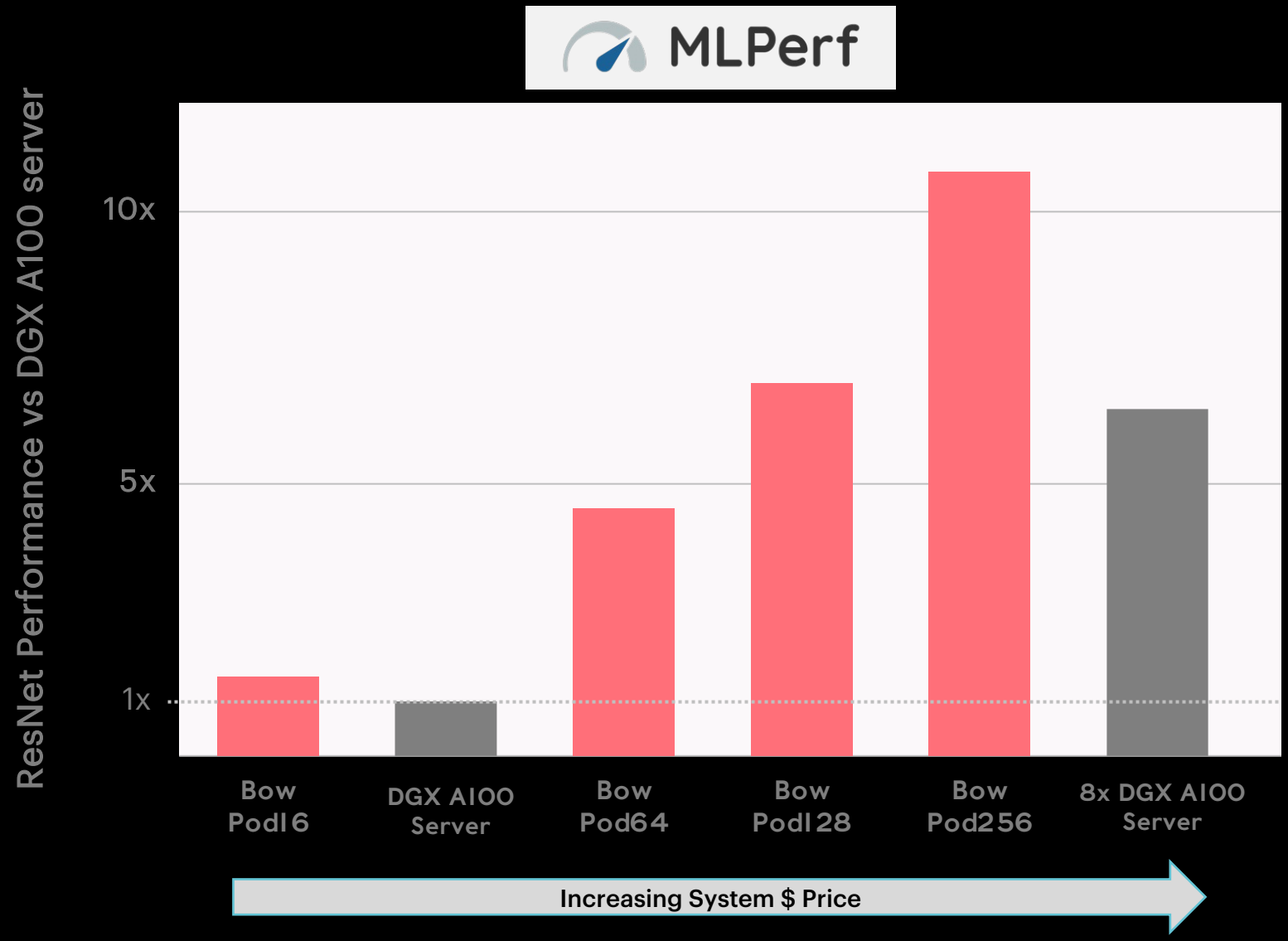
SPEED-UP OVER DGX-A100 ACROSS RANGE OF MODELS | SCALING TO LARGER POD SYSTEMS



Bow Pod Platforms | SDK2.6 | ResNet-50 v1.5 Training Throughput TensorFlow | G16-EfficientNet-B4 TTT Tensorflow | BERT Large Ph1 Pre-Training (SL128) PopART Packed DGX A100 (A100-SXM4-80GB) | TensorFlow | Mixed Precision | https://developer.nvidia.com/deep-learning-performance-training-inference

MLPERF RESNET: BOW POD PERFORMANCE

BOW POD256 DELIVERS SIGNIFICANTLY BETTER PERFORMANCE THAN 8X DGX A100



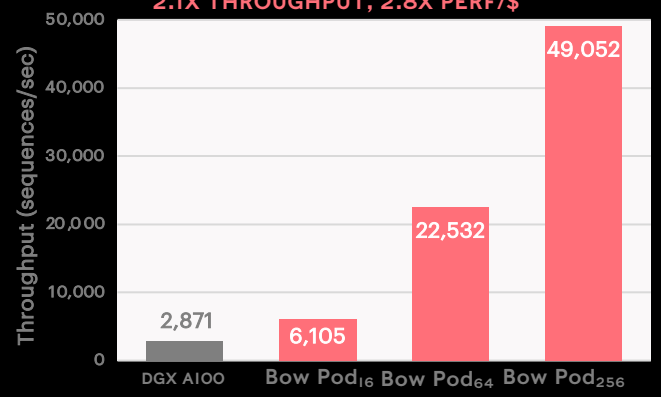
GRAPHCORE

MLPerf v2.0 Training Results | MLPerf ID: 2.0-2047, 2.0-2050, 2.0-2052, 2.0-2054, 2.0-2090, 2.0-2094, 2.0-2073
The MLPerf name and logo are trademarks. See www.mlperf.org for more information,

TRAINING BENCHMARKS

BERT-Large Pre-Training (SL128) Throughput

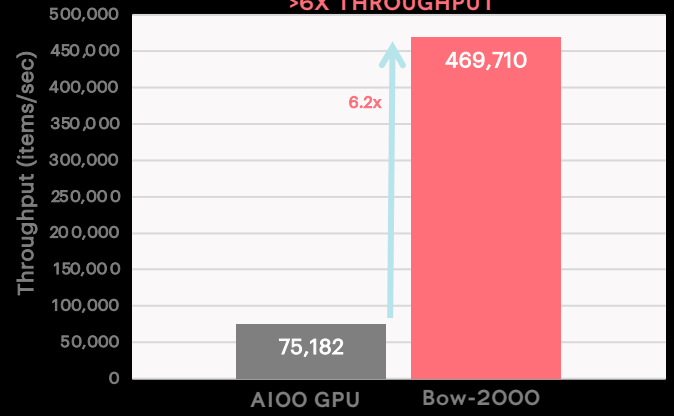
2.IX THROUGHPUT, 2.8X PERF/$



BERT-Large Phase 1 Pre-Training Throughput (SL128)
Bow Pod16 | Bow Pod64 | Bow Pod256 | PopART with Packing | SDK 2.6 | https://www.graphcore.ai/performance-results
DGX A100 320GB (A100-SXM4-40GB) | Mixed Precision | published BERT results
DGX A100 TensorFlow - https://ngc.nvidia.com/catalog/resources/nvidia:bert_for_tensorflow/performance

MPNN-GIN Training Throughput

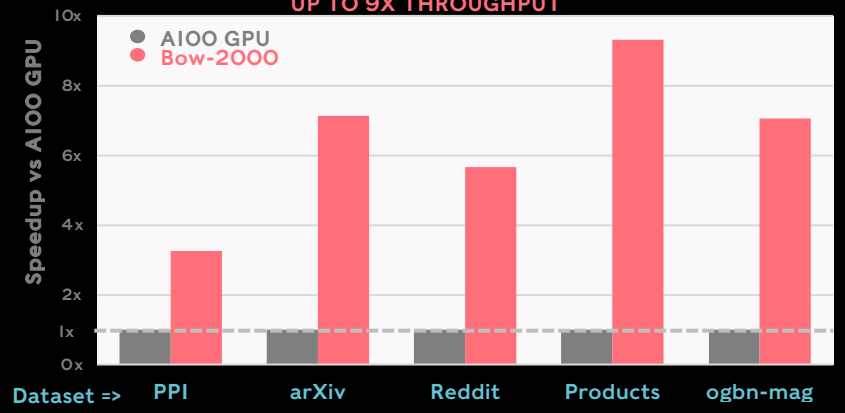
>6X THROUGHPUT



MPNN-GIN Graph Isomorphism Network Training | mol-hiv dataset (host-generated) | TensorFlow 2
41K graphs, 25.5 nodes per graph, 9-dimensional input node features | 1.7M model parameters
Bow-2000 | FP 16.16 | SDK 2.6 | https://www.graphcore.ai/performance-results | 1x A100 (A100-SXM4-40GB)
| FP16 | Measured results

Cluster-GCN Training

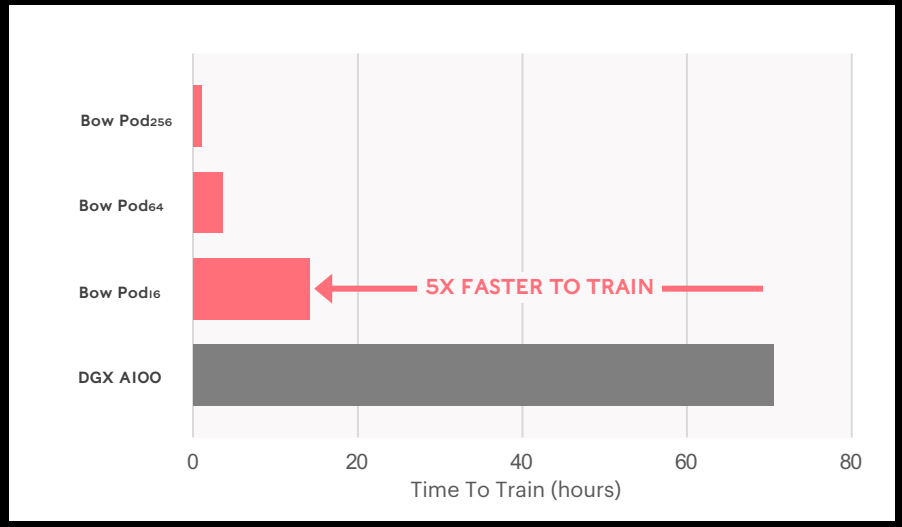
UP TO 9X THROUGHPUT



Cluster-Graph Convolutional Networks Training | Multiple Datasets | TensorFlow 2 | FP16
Bow-2000 | SDK 2.6 | https://www.graphcore.ai/performance-results | (A100-SXM4-40GB) measured results |
Throughput values are measured using the best combination of data structure and matmul implementation for each device and each dataset

EfficienNet-B4 Time-To-Train

5X SHORTER TTT, ~IOX PERF/$



EfficientNet-B4 Training Time To Train Performance
Bow Pod Platforms | SDK2.6 Results | G16-EfficientNet-B4 Training
DGX A100 (A100-SXM4-80GB) | TensorFlow | Mixed Precision |
https://developer.nvidia.com/deep-learning-performance-training-inference

GRAPHCORE

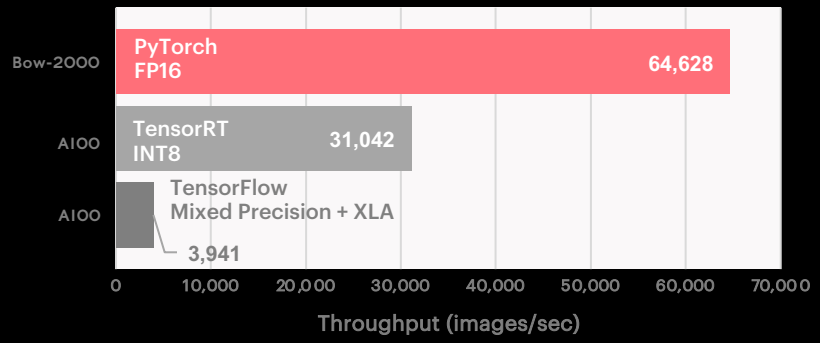
GRAPHCORE BOW POD16
$149,995 MSRP

NVIDIA DGX-A100 640GB SERVER
$299,000 MSRP

INFERENCE BENCHMARKS

ResNet50 Inference

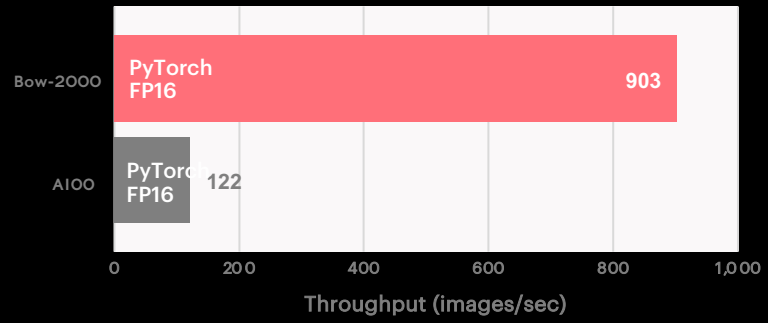
3.8X THROUGHPUT ON TF2/FP16



ResNet-50 v1.5 Inference | Highest throughput comparison
1x Bow-M2000 | FP 16.16 | SDK2.6 | Synthetic Data (host-generated) | https://www.graphcore.ai/performance-results
1x A100 (A100-SXM4-80GB) results using TensorRT 8.0 | INT8 | Synthetic Data (host-generated)
1x A100 (A100-SXM4-40GB) results using TensorFlow | Mixed Precision + XLA | Synthetic Data (host-generated)
Results published by NVIDIA | A100 TensorRT: (https://developer.nvidia.com/deep-learning-performance-training-inference) on 22 Mar 2022
A100 TensorFlow: https://github.com/NVIDIA/DeepLearningExamples/tree/master/TensorFlow/Classification/ConvNets/resnet50v1.5#inference-performance-nvidia-dgx-a100-1x-a100-40gb

YOLO V4 Inference

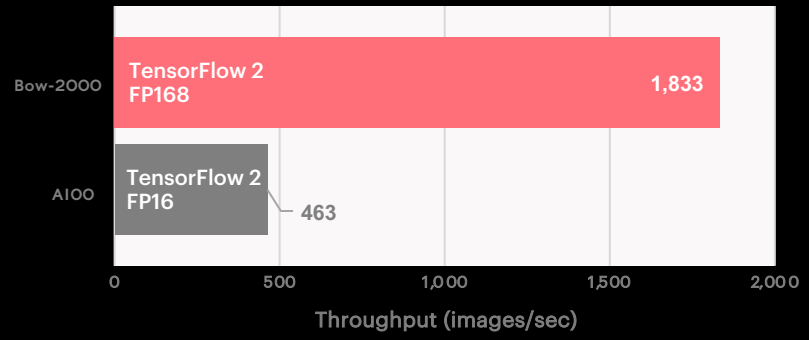
>7X THROUGHPUT



YOLO v4 Inference| Highest throughput comparison | Image Size 896
1x Bow-2000 | FP 16.16 | SDK 2.6 | PyTorch | Synthetic Data (host-generated) | https://www.graphcore.ai/performance-results,
1x A100 (A100-SXM4-40GB) results using PyTorch | FP16 | Synthetic Data(host-generated)

UNET (Medical) Inference

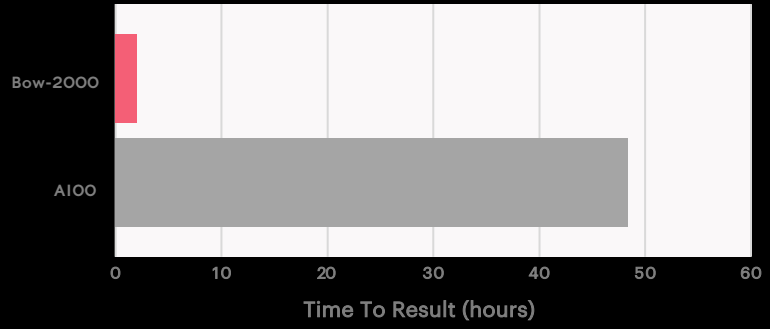
3.8X THROUGHPUT



Unet (medical) Inference| Highest throughput comparison | 572x572x1
1x Bow-M2000 | FP 16.16 | SDK 2.6 | Synthetic Data (host-generated) | https://www.graphcore.ai/performance-results,
1x A100 (A100-SXM4-80GB) results using TensorFlow2 | FP16 | Synthetic Data (host-generated)

MCMC Alpha Estimation

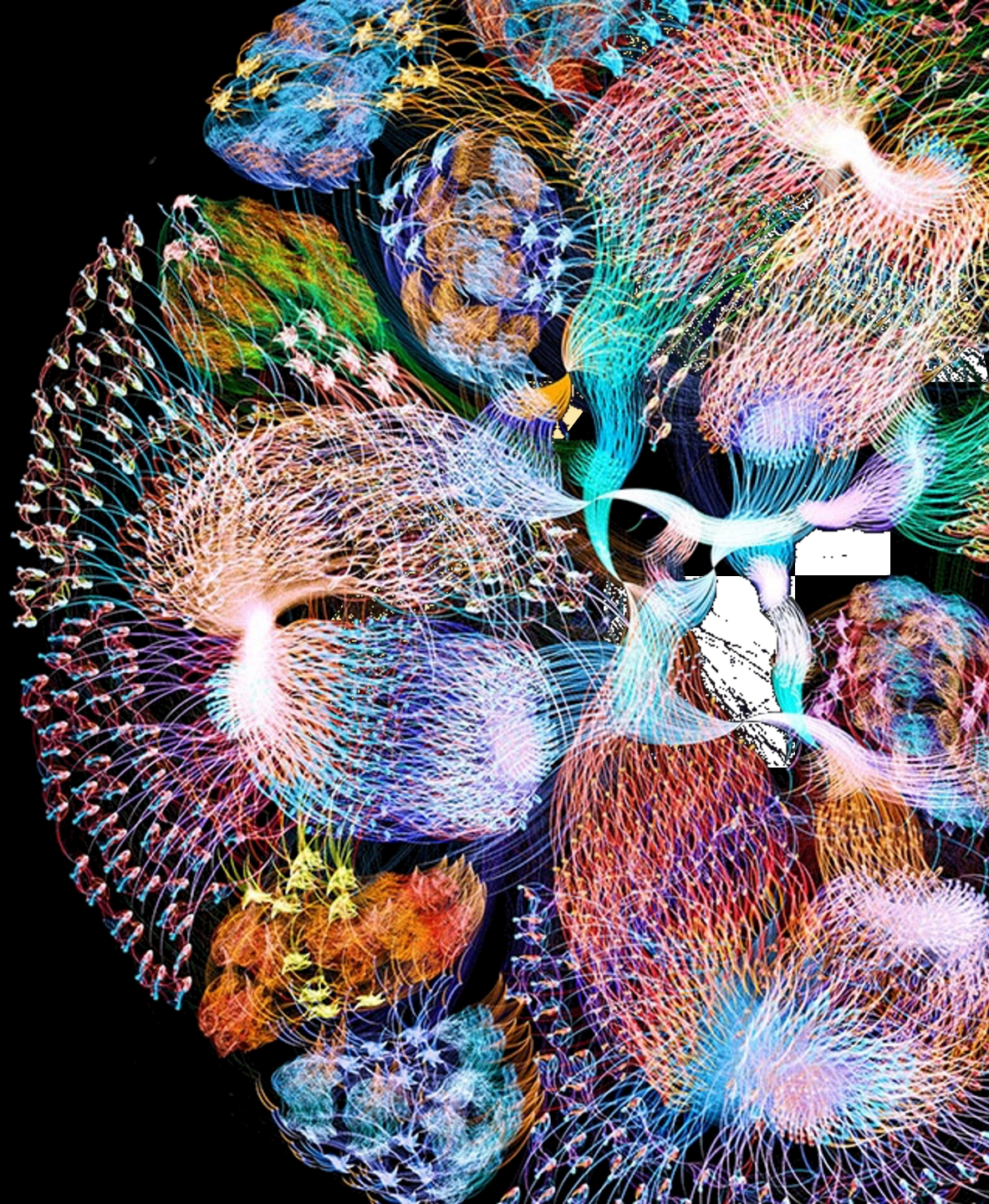
>20X SPEEDUP



Markov Chain Monte Carlo – Probabilistic model with TensorFlow Probability, representative of workload used by Carmot Capital
Neural network with 3 fully-connected layers (num units in 1st layer=40, #dimensions in training set =22, #leapfrog steps=1000
Results multiplied by number of calcs in sliding window=200
1x Bow-2000 using TensorFlow | FP 32.32 | SDK 2.6 | 1600 samples | https://www.graphcore.ai/performance-results
1x A100 (A100-SXM4-40GB) | FP 32.32 | 1600 samples

GRAPHCORE

CUSTOMER CASE STUDIES



GRAPHCORE

HIGH ENERGY PHYSICS – PARTICLE DETECTOR

Model: CookieNetAE is a deep neural network designed to estimates the energy-angle dependent probability density function of electrons' energy.

Dataset: The training dataset 900,000 pairs of 128x128 images, a grainy input image and a smooth target image. The inference dataset consisted of 50,000 unique 128x128 images.

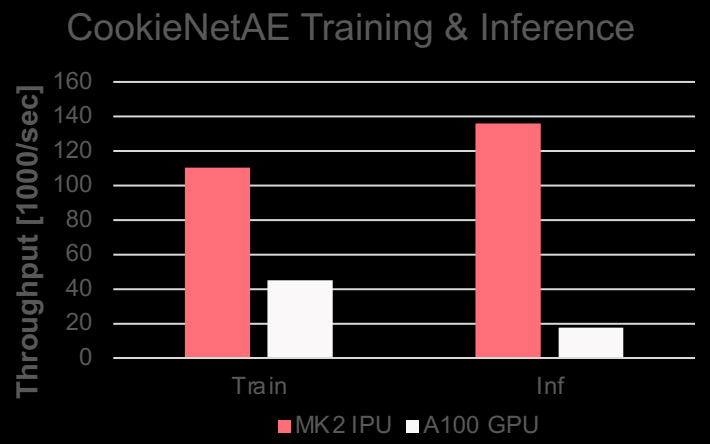
Task: Inference **-** Fast de-noising of detector images. Training – Quick "at-detector" re-training of model.

Goals:

- Improve inference throughput at detector
- Compress time-to-train

Results:

- Inference – **7.7x throughput** vs GPU
- Training – **2.4x TTT improvement** vs GPU



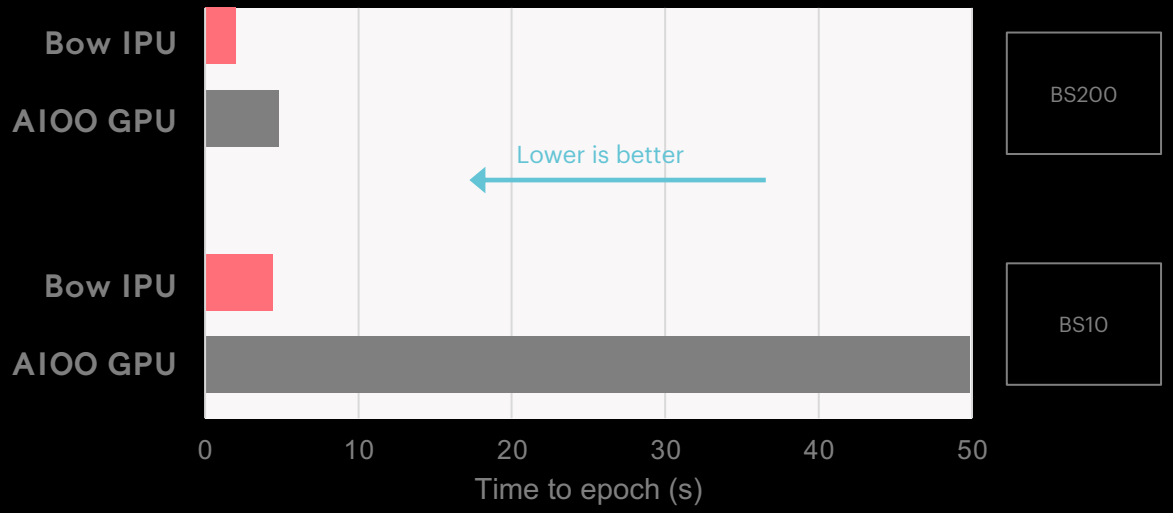
Machine Learning via direct attached accelerator for streaming data processing at high shot rate x-ray free electron lasers paper in review with Frontiers in Physics

GNN: CUSTOMER SUCCESS

10x FASTER FOR TGN GRAPH ML MODEL

Michael Bronstein
Head of Graph ML Research
@Twitter, DeepMind Professor of AI,
University of Oxford

IPU Advantage for Dynamic Temporal Graph Networks



Using dynamic graph ML model, TGN, on IPU to achieve 10x faster time to result. Many real-world graphs are dynamic and evolve over time, including social networks, financial transactions, and recommender systems. In many cases, it is the dynamic behaviour of such systems that conveys important insights.

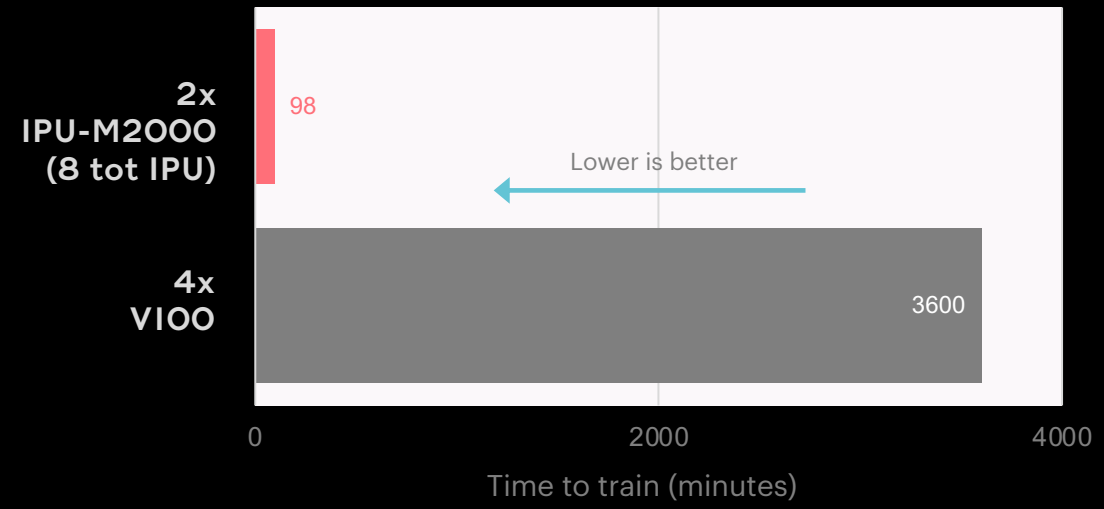
GNN: CUSTOMER SUCCESS

36X FASTER FOR SCHNET GRAPH ML MODEL

(8 IPU VS 4 GPU)

Pacific Northwest
NATIONAL LABORATORY

IPU Advantage for molecular dynamics SchNet GNN



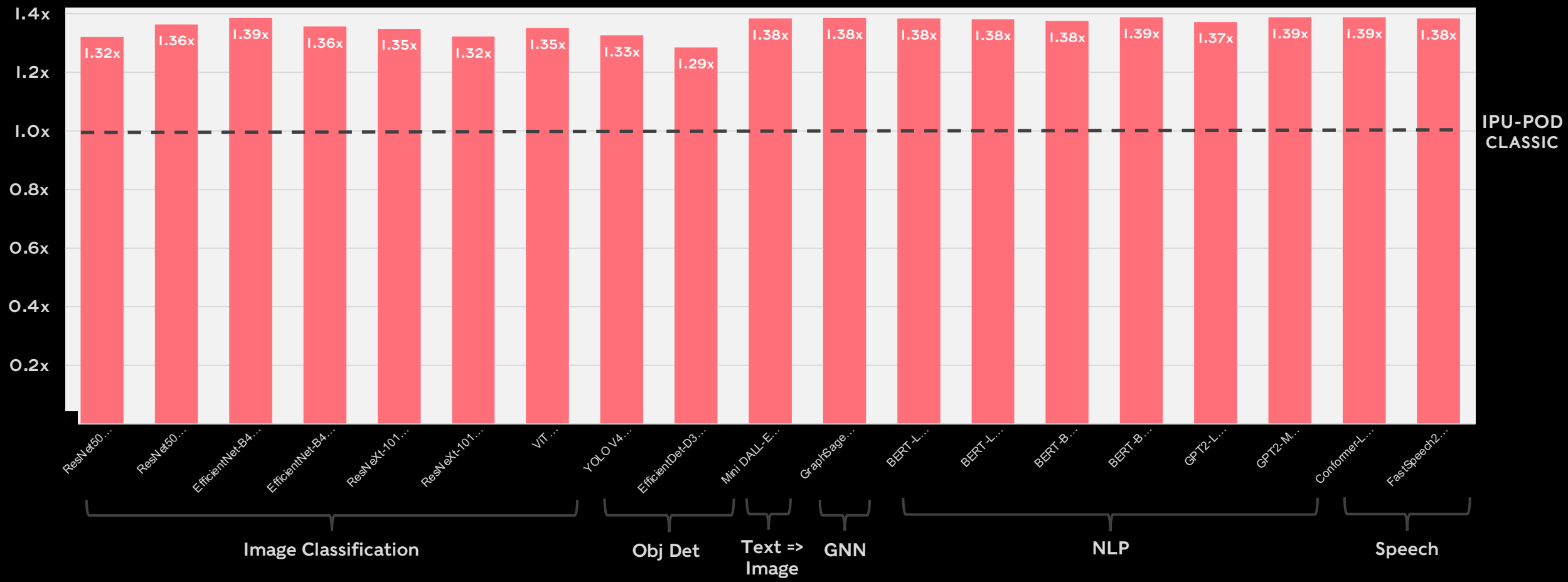
US Department of Energy National Lab, PNNL, training SchNet Graph Neural Network[1] with a 500k water clusters dataset[2] to predict the potential energy per cluster & seeing **36x faster time to result with IPU**

[1] JK. T. Schütt1, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. "SchNet – A deep learning architecture for molecules and materials" *J. Chem. Phys.* **148**, 241722 (2018).

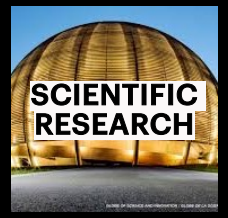
[2] Jenna A. Bilbrey, Joseph P. Heindel, Malachi Schram, Pradipta Bandyopadhyay, Sotiris S. Xantheas, and Sutanay Choudhury. "A look inside the black box: Using graph-theoretical descriptors to interpret a Continuous-Filter Convolutional Neural Network (CF-CNN) trained on the global and local minimum energy structures of neutral water clusters" *J. Chem. Phys.* **153**, 024302 (2020).

BOW REAL WORLD PERFORMANCE GAINS

BOW POD VS IPU-POD CLASSIC: UP TO 40% IMPROVEMENT



Bow Pod64 vs IPU-POD64 | G16-EfficientNet-B4 Training/TensorFlow/ | Vision Transfomer (ViT) Training/PyTorch | GPT2-Large Training/PyTorch | Conformer SpeechToText Automatic Speech Recognition/PyTorch |
Bow Pod16 vs IPU-POD16 | ResNet-50 v1.5 Training/TensorFlow | ResNeXt-101 Training/ TensorFlow | Mini DALL-E Training/PyTorch | GraphSage Training/TensorFlow2 | BERT-Large Ph1 Pre-Training (SL128)/PopART | BERT-Base Ph1 Pre-Training (SL128)/PopART | GPT2-Medium Training/PyTorch | FastSpeech2 Training/TensorFlow2
Bow-2000 vs IPU-M2000 | ResNet50 v1.5 Inference/PyTorch | EfficientNet-B4 Inference/PyTorch| ResNeXt-101 Inference/TensorFlow | | YOLOv4 Inference/PyTorch | EfficientDet-D3 Inference/TF2 with Keras | BERT-Large Inference (SL128)/PopART | BERT-Base Inference (SL128)/PopART
Comparing Preliminary Results (Pre-SDK2.5)

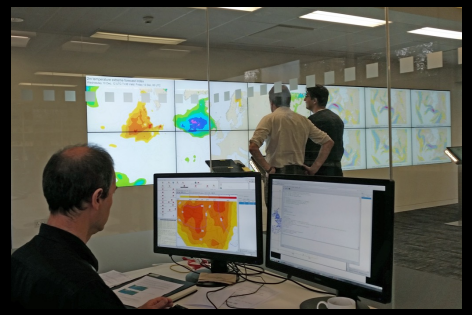


Accelerating HPC with AI using IPU

European Centre for Medium-range Weather Forecasting

ECMWF

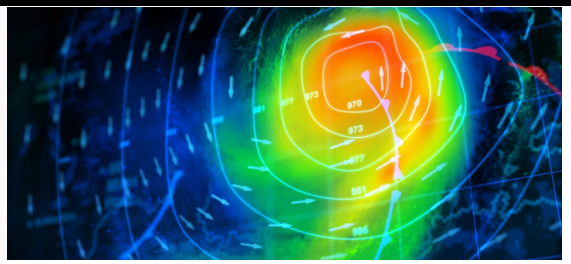
Graphcore IPU trained an ECMWF weather forecasting model **5x faster than a leading GPU (and potentially up to 50x faster than CPUs)**



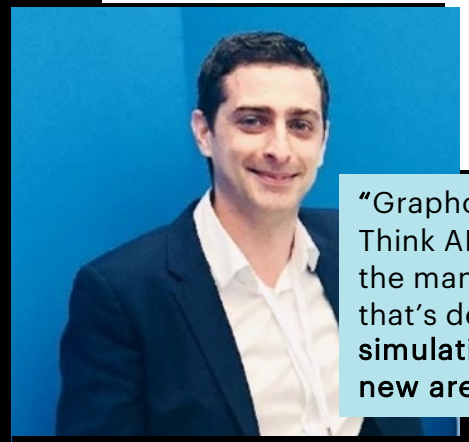
Mar 09, 2022 \ AI, HPC

AI FOR SIMULATION: HOW GRAPHCORE IS HELPING TRANSFORM TRADITIONAL HPC

Written By:
Alex Titterton



Atos



"Graphcore plays a central role in Atos' Think AI solution, helping customers take advantage of the many benefits that AI is bringing to HPC – whether that's delivering **faster and more accurate simulations**, **improving cost efficiency**, or opening up **new areas of research and commercial applications**."

Cedric Bourrasset
Head of HPC AI at Atos

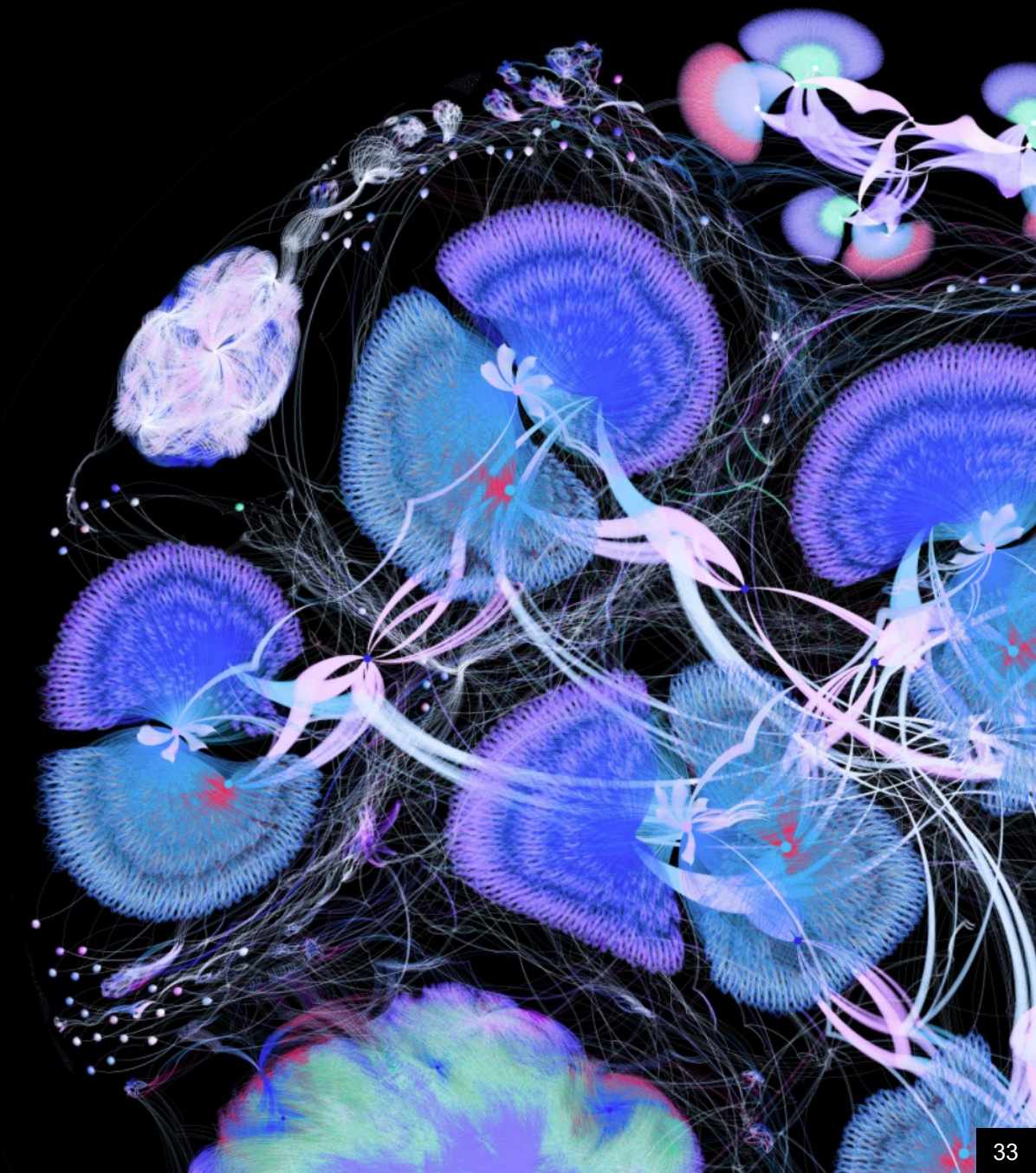
"Without any changes to the model definition or its hyperparameters, a single IPU processor was able to train this MLP model **5x faster than an A100 GPU** and some 50x faster than either the MLP model or the traditional parametrisation scheme performed on a CPU."

"Graphcore's IPU, designed from the ground up for machine intelligence, is the ideal platform on which to build, explore and grow the next generation of machine learning-driven solvers, emulators and simulations."

More information can be found in the full technical blog post:
https://www.graphcore.ai/posts/ai-for-simulation-how-graphcore-is-helping-transform-traditional-hpc

https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2021MS002477

ARGONNE APPLICATIONS



CANDLE UNO

CANcer distributed Learning Environment (CANDLE)

Model: CANDLE Uno for precision medicine related to cancer care

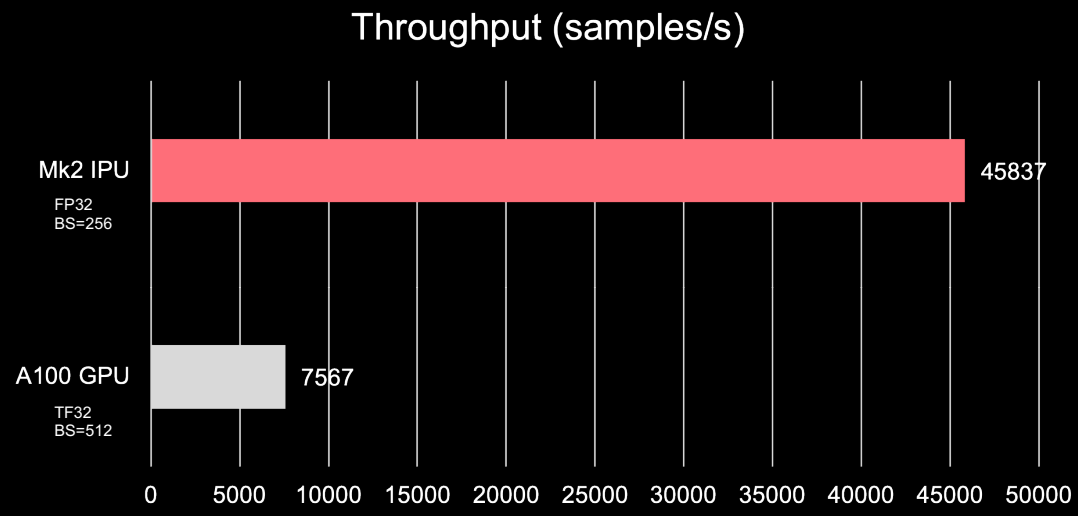
Dataset: dataset with 20 million training samples

Task: train model to predict tumor-drug response

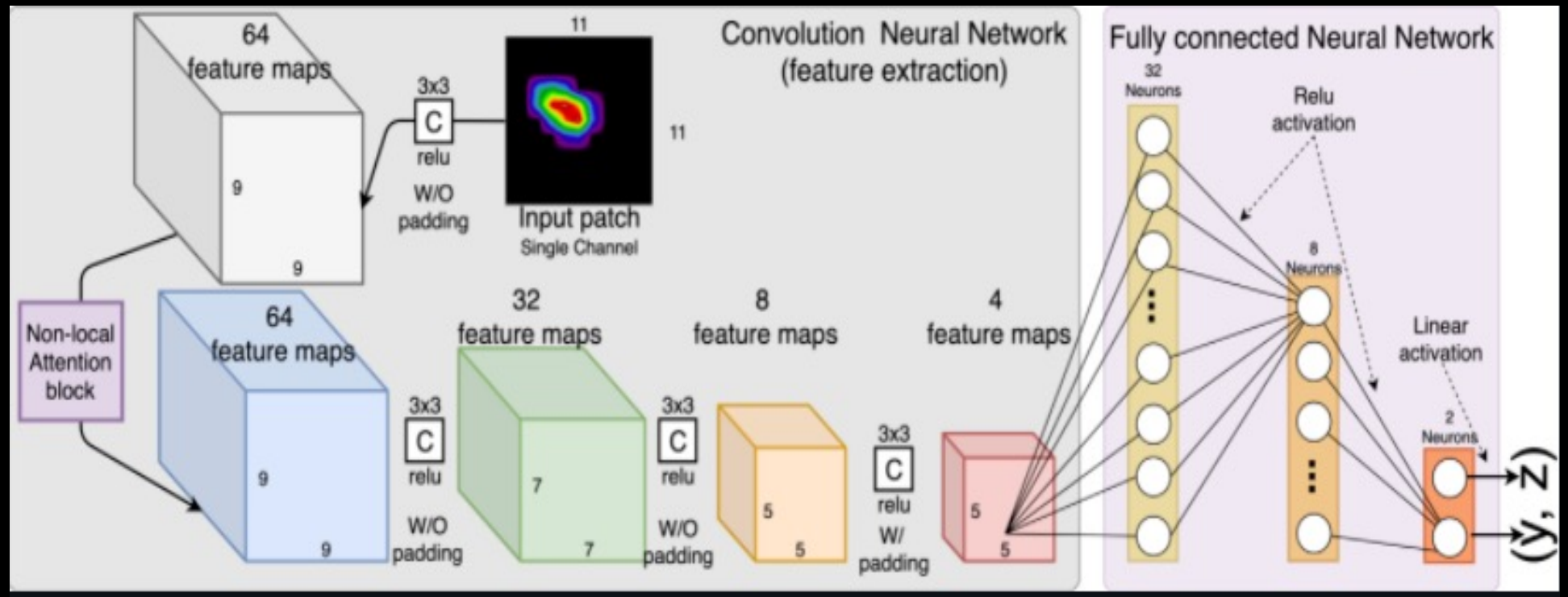
Goals:

- Reduce time-to-train
- Demonstrate efficient scaling

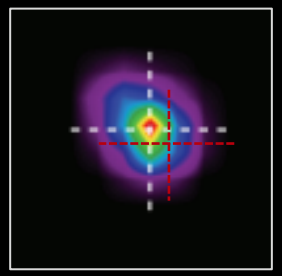
Results: Increased training throughput by
6x @ higher precision



BraggNN Model Details



Peak Location with
Sub-Pixel Accuracy



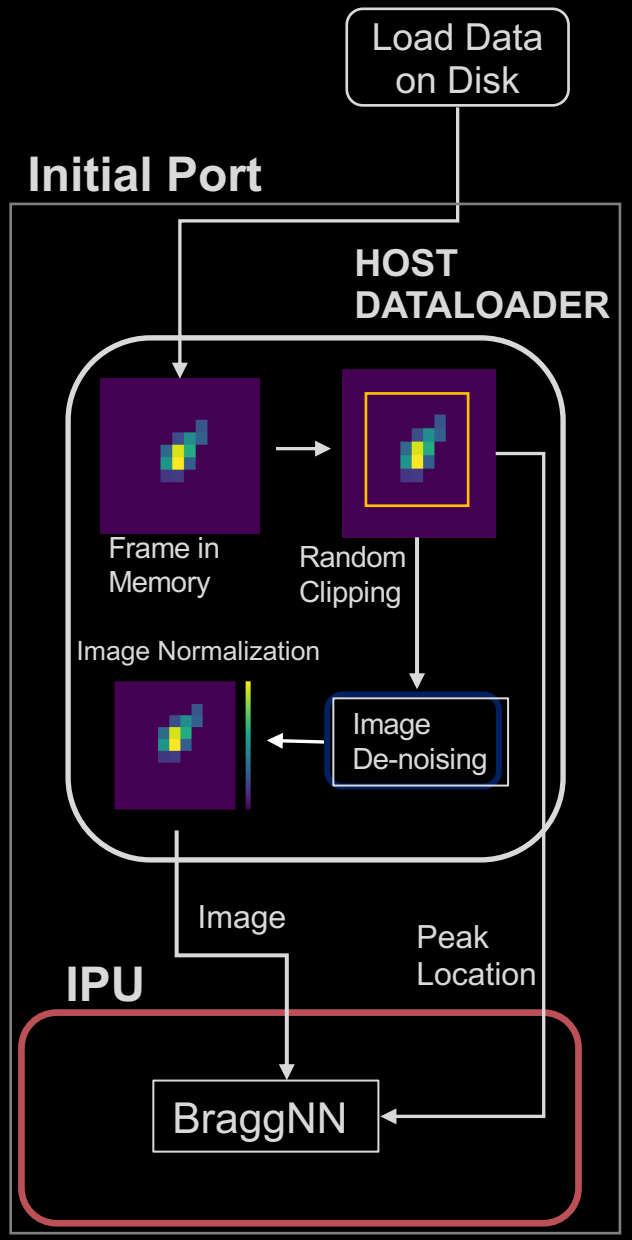
Framework: PyTorch
Input: An 11x11 (single channel) image generated on the fly by cropping a larger detector image
Output: Normalized location of the peak in (y, z) coordinates
Loss: Mean square error between predicted location and target location
GPU Performance Reference: ~800 seconds for 500 epochs with minibatch size: 512

GRAPHCORE

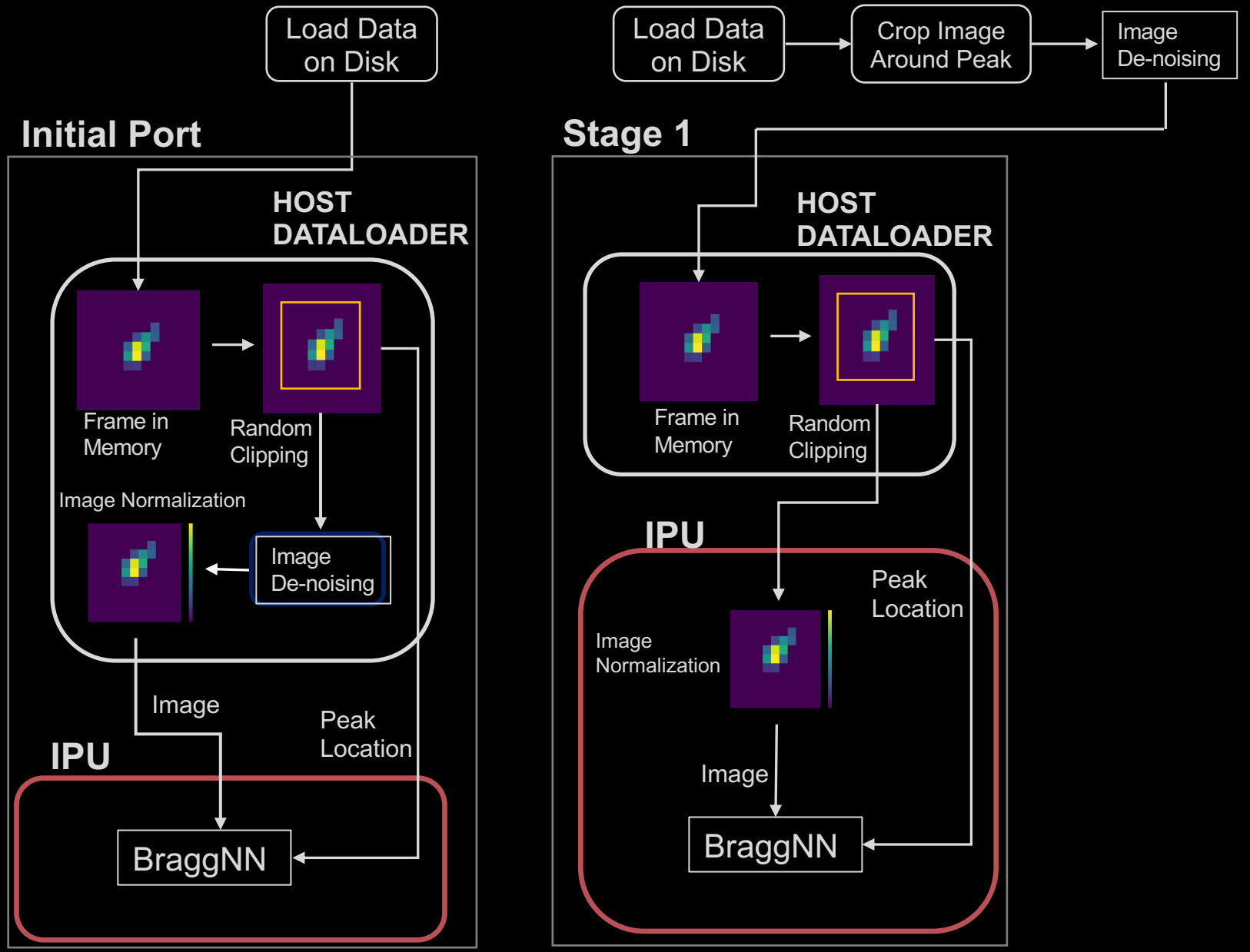
Goal: Reduce the time to train to < **2 minutes** & show excellent scaling over multiple IPUs

Zhengchun L. et al. "BraggNN: Fast X-ray Bragg Peak Analysis Using Deep Learning", 2021

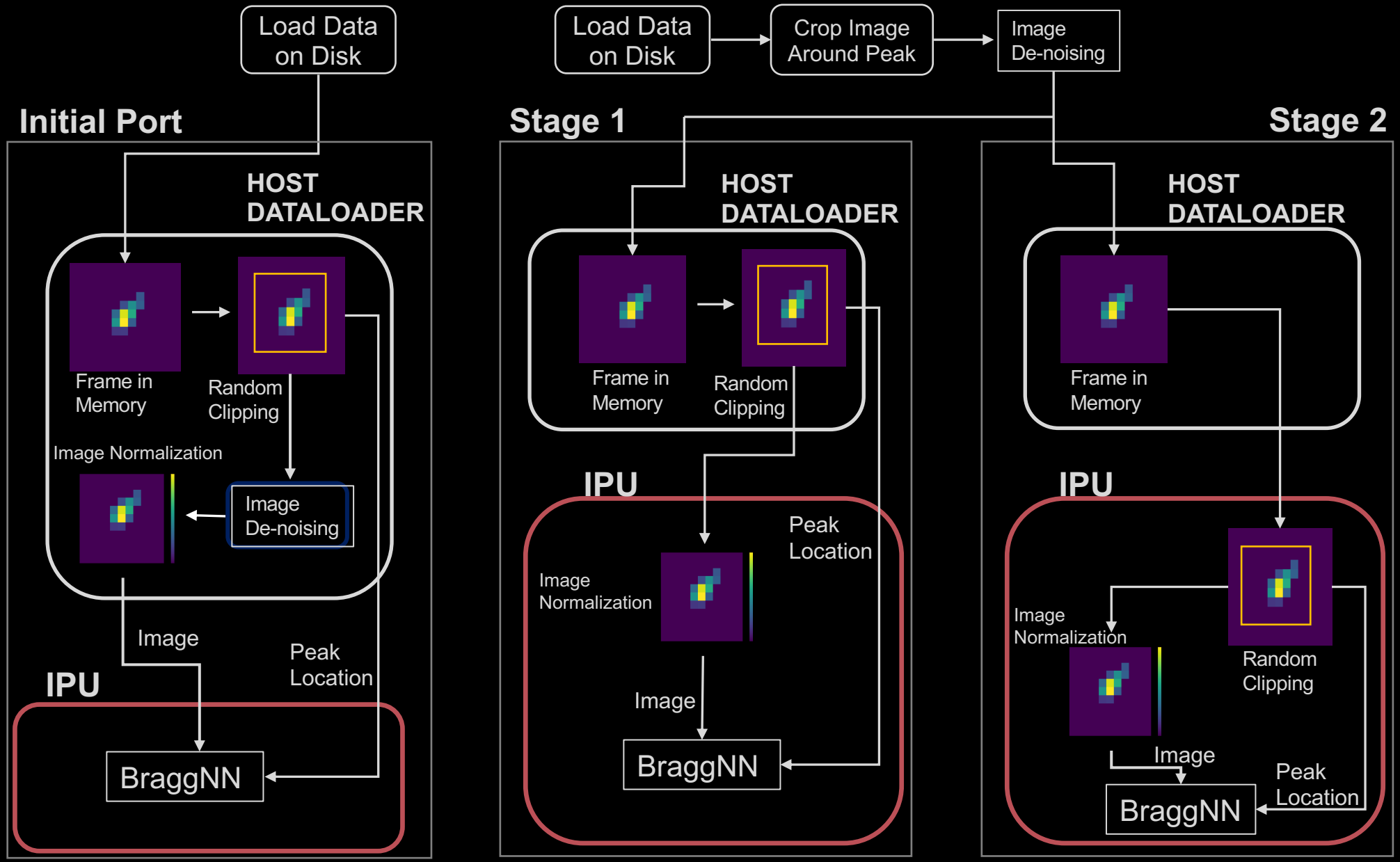
DATAFLOW MODIFICATION FOR FASTER TRAINING



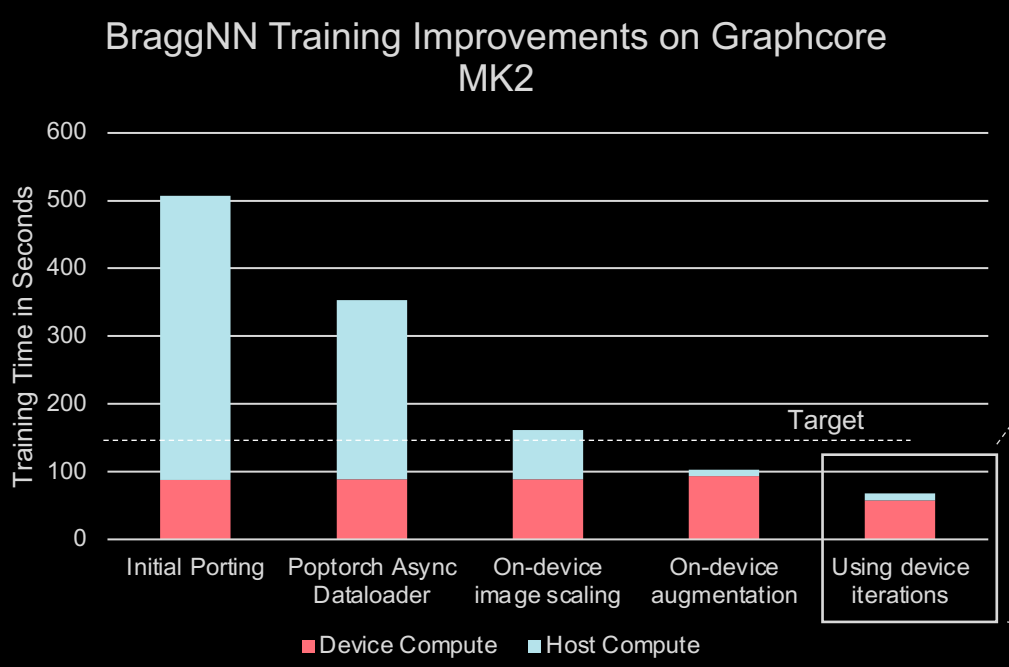
DATAFLOW MODIFICATION FOR FASTER TRAINING

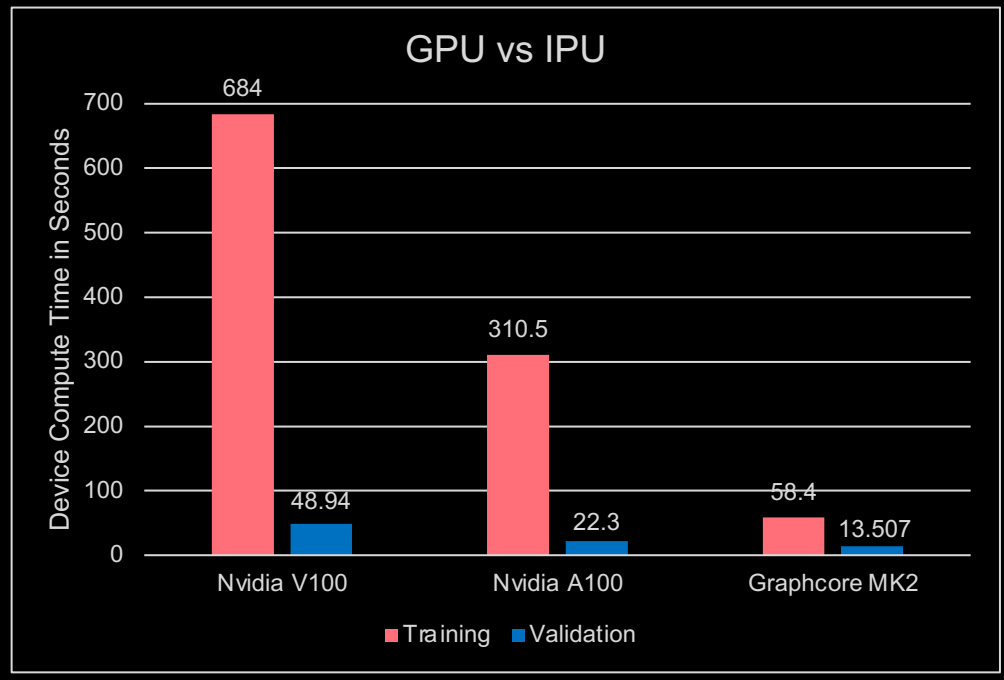


DATAFLOW MODIFICATION FOR FASTER TRAINING



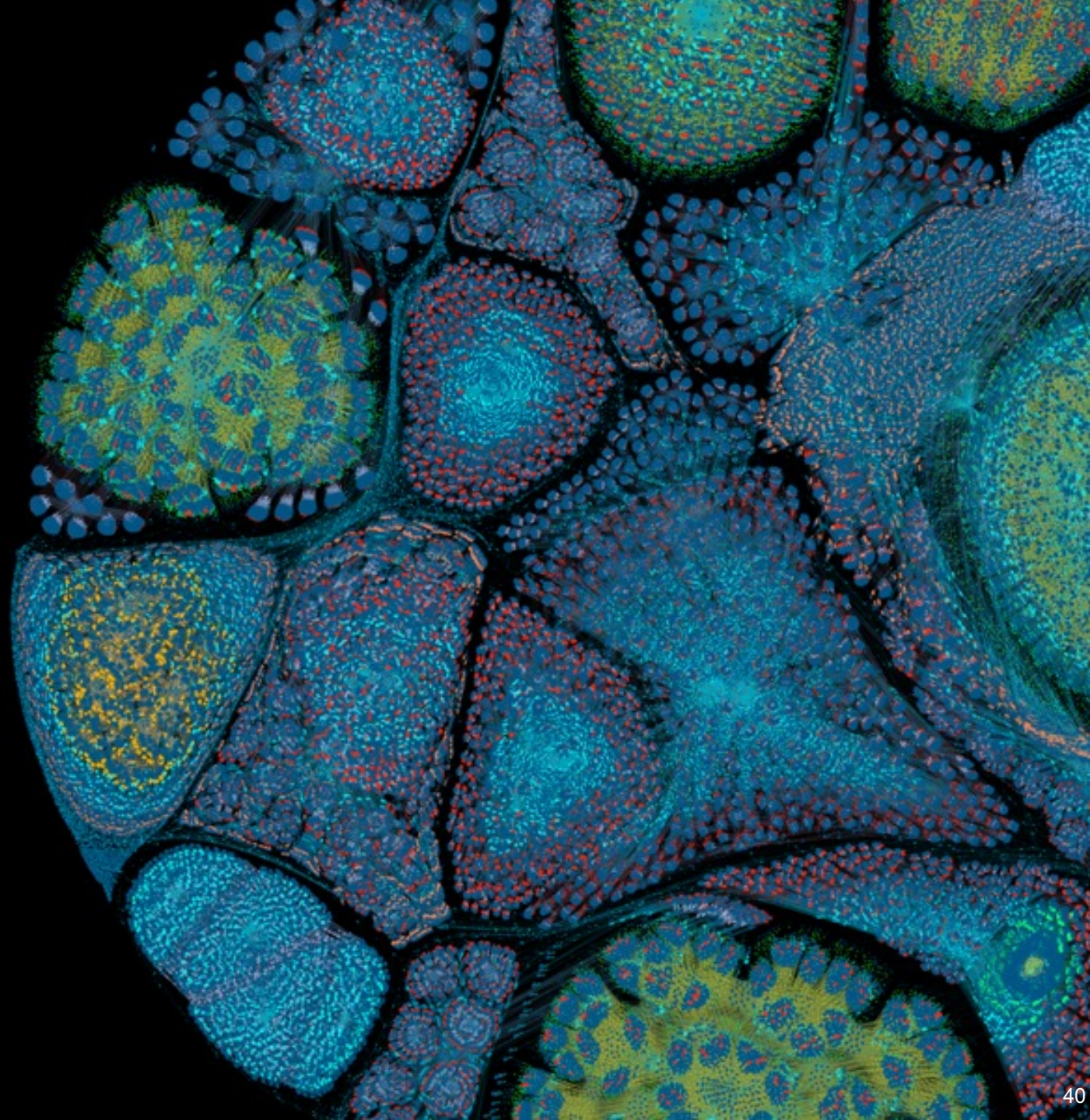
BRAGGNN SUMMARY





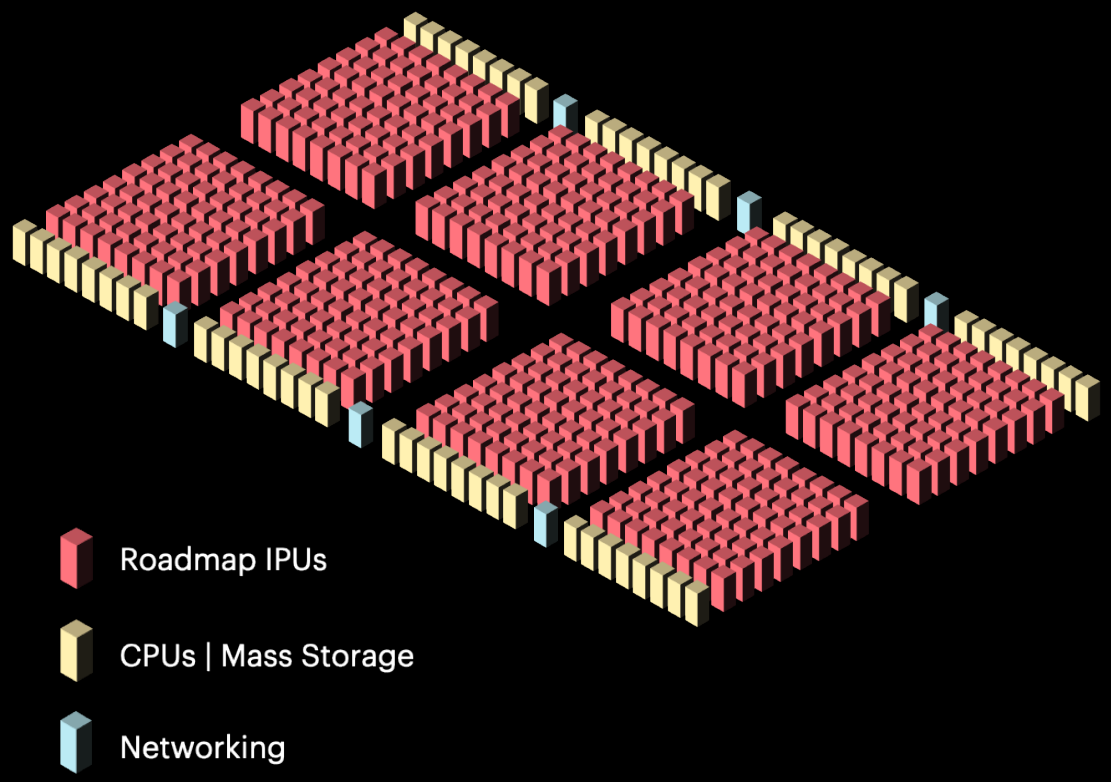
- Out of the box, training on MK2 IPU is **3-4x** faster than V100
- Flexibility of Graphcore Architecture allowed migration of image scaling and data augmentation to IPU
- Final version of BraggNN is **>5x** faster on IPU compared to A100 and **>11x** faster than V100

“GOOD” COMPUTER



GRAPHCORE

ANNOUNCING: THE 'GOOD' COMPUTER



Over 10 Exa-Flops of AI floating point compute from 8,192 roadmap IPUs

3D Wafer-on-Wafer logic stack

Up to 4 PB of memory with bandwidth of over 10 PB/s

Enabling AI models to be developed with 500 Tn parameters

Fully supported by Poplar® SDK

GRAPHCORE

THANK YOU

Richard Bohl
richardb@graphcore.ai

GRAPHCORE