2022 ALCF Simulation, Data, and Learning Workshop

Introduction of AI Testbed and hands-on

Zhen Xie, Murali Emani, Siddhisanket Raskar, Varuni Sastry, William Arnold, Bruce Wilson, Rajeev Thakur, Venkatram Vishwanath

Argonne Leadership Computing Facility (ALCF)

Argonne National Laboratory, Lemont, IL 60439

Surge of Scientific machine learning

Simulations/ surrogate models

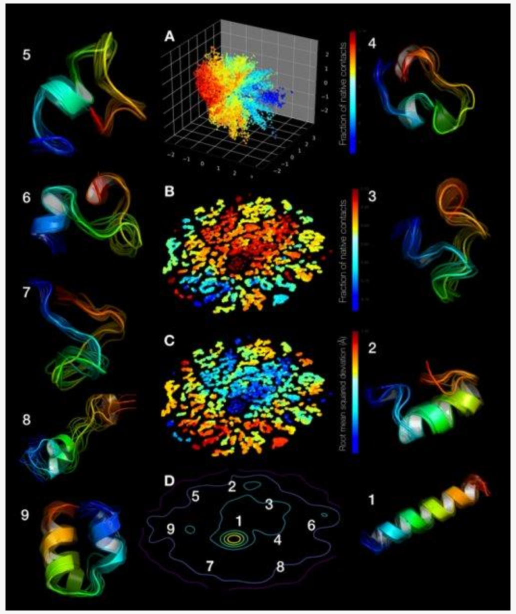
– Replace, in part, or guide simulations with AI-driven surrogate models

Data-driven models

– Use data to build models without simulations

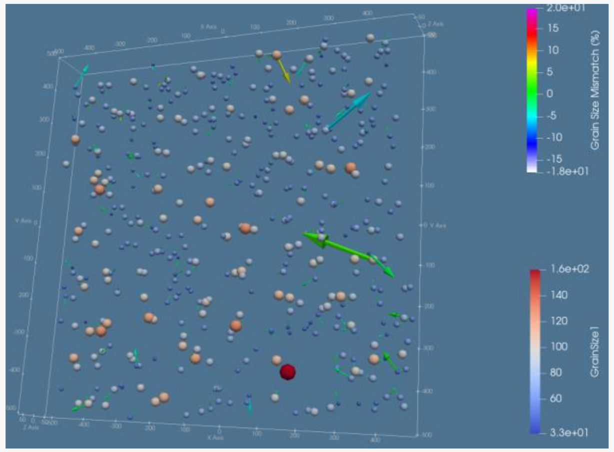
Co-design of experiments

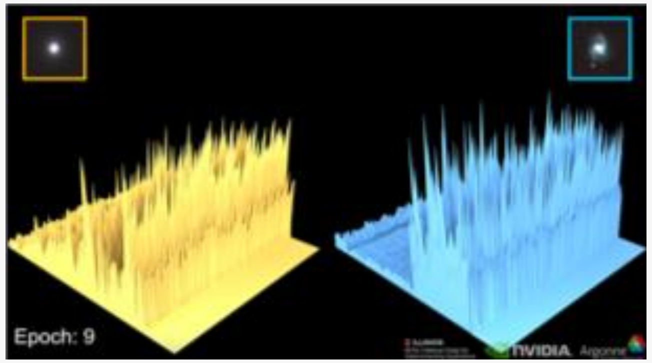
– AI-driven experiments



Protein-folding

Braggs Peak

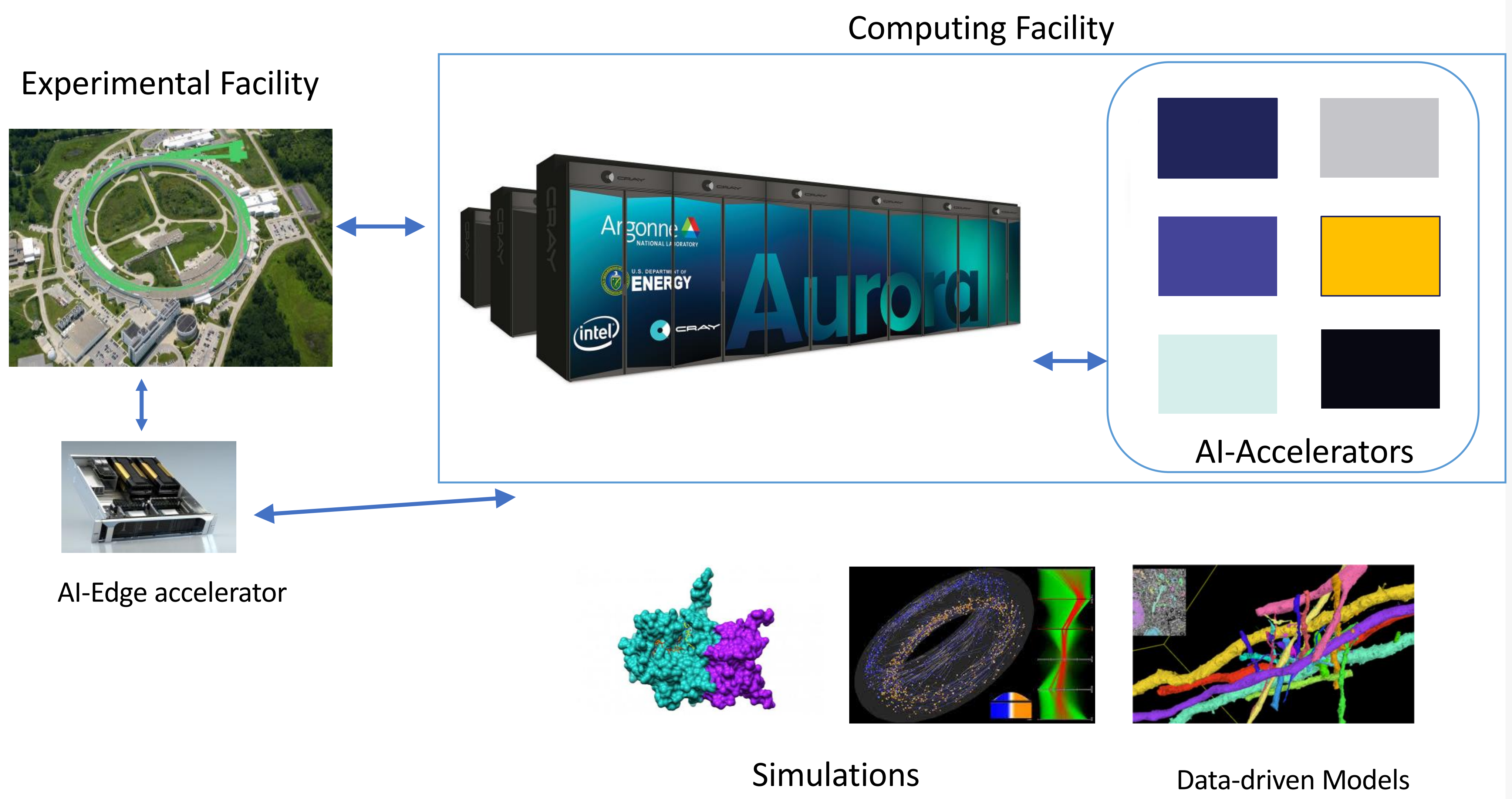




Galaxy Classification

Design infrastructure to facilitate and accelerate AI for Science applications

Integrating AI systems in facilities



AI PATHFINDING

Goals of ALCF AI Activities at Argonne

Accelerate science by effective coupling of AI-systems, exascale supercomputers and experimental facilities

1. Maturity of software and hardware for science
2. Ability to scale hardware and integrate with facility
3. Application at scale to science

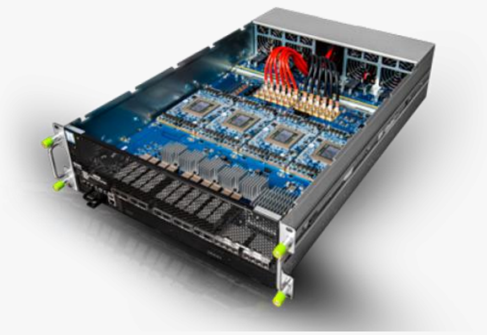
ALCF AI Testbeds

https://www.alcf.anl.gov/alcf-ai-testbed

Cerebras (CS-2)

SambaNova

Graphcore

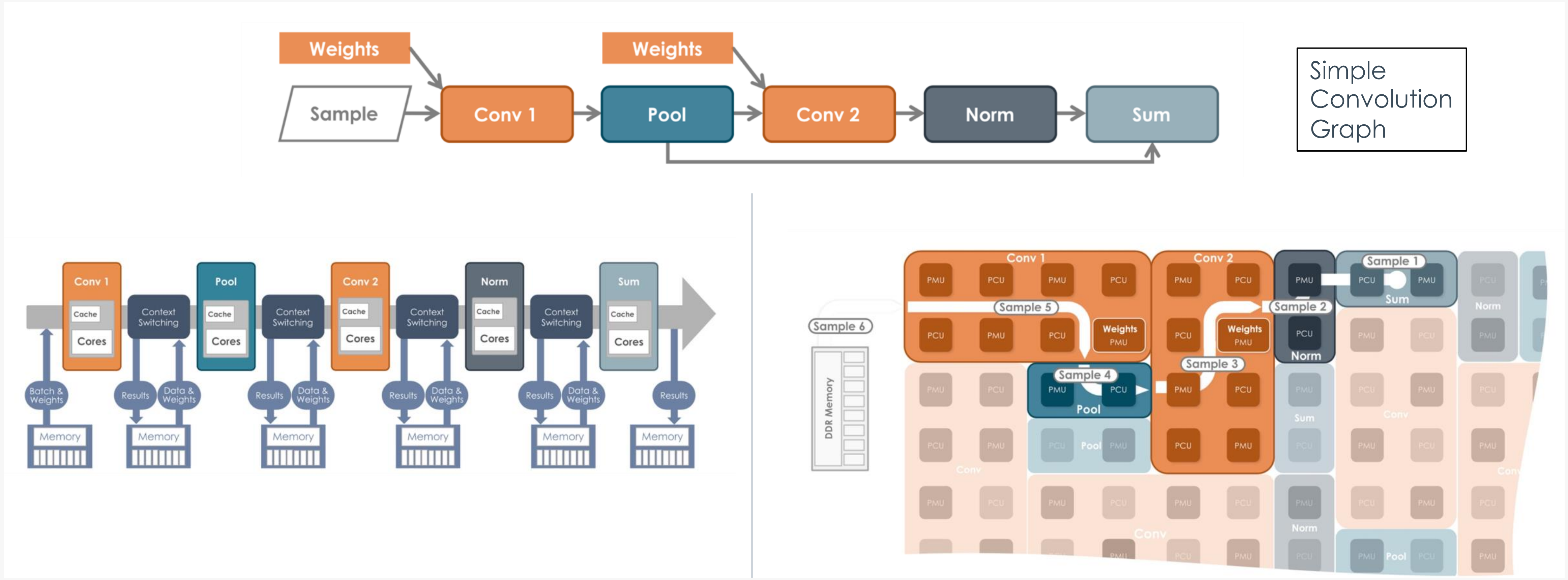


Habana

Groq

- Infrastructure of next-generation machines with hardware accelerators customized for artificial intelligence (AI) applications.
- Provide a platform to evaluate usability and performance of machine learning based HPC applications running on these accelerators.
- The goal is to better understand how to integrate AI accelerators with ALCF's existing and upcoming supercomputers to accelerate science insights

Dataflow Architectures



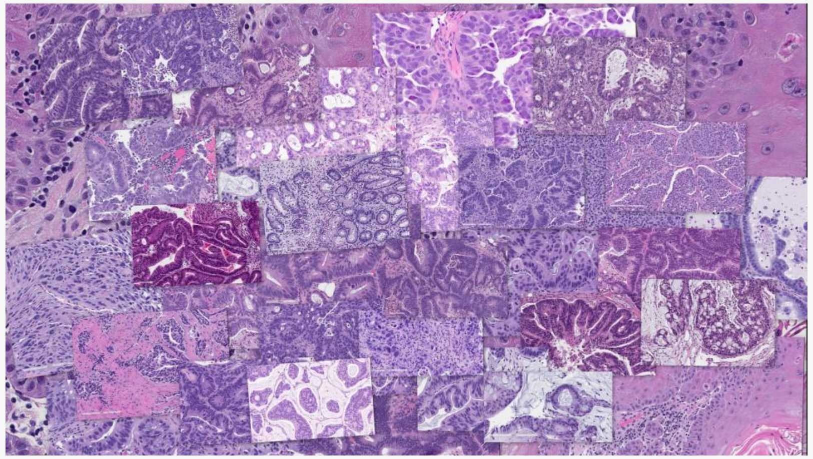
GPU accelerators: Each kernel is launched onto the device and bottlenecks include memory bandwidth and kernel-launch latencies

Dataflow: Kernels are spatially mapped onto the accelerator and data flows on-chip between them reducing memory traffic

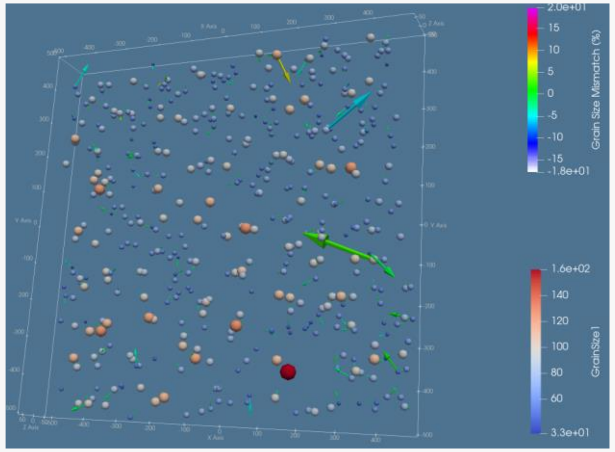
Image Courtesy: Sumti Jairath, SambaNova

	Cerebras CS2	SambaNova Cardinal SN10	Groq GroqCard	GraphCore GC200 IPU	Habana Gaudi1	NVIDIA A100
Compute Units	850,000 Cores	640 PCUs	5120 vector ALUs	1472 IPUs	8 TPC + GEMM engine	6912 Cuda Cores
On-Chip Memory	40 GB	>300MB	230MB	900MB	24 MB	192KB L1 40MB L2
Process	7nm	7nm	14nm	7nm	7nm	7nm
System Size	2 Nodes	2 nodes (8 cards per node)	4 nodes (8 cards per node)	1 node (8 cards per node)	2 nodes (8 cards per node)	Several systems
Estimated Performance of a card (TFlops)	>5780 (FP16)	>300 (BF16)	>188 (FP16)	>250 (FP16)	>150 (FP16)	312 (FP16), 156 (FP32)
Software Stack Support	Tensorflow, Pytorch	SambaFlow, Pytorch	GroqAPI, ONNX	Tensorflow, Pytorch, PopArt	Synapse AI, TensorFlow and PyTorch	Tensorflow, Pytorch, etc
Interconnect	Ethernet-based	Infiniband	RealScale TM	IPU Link	Ethernet-based	NVLink

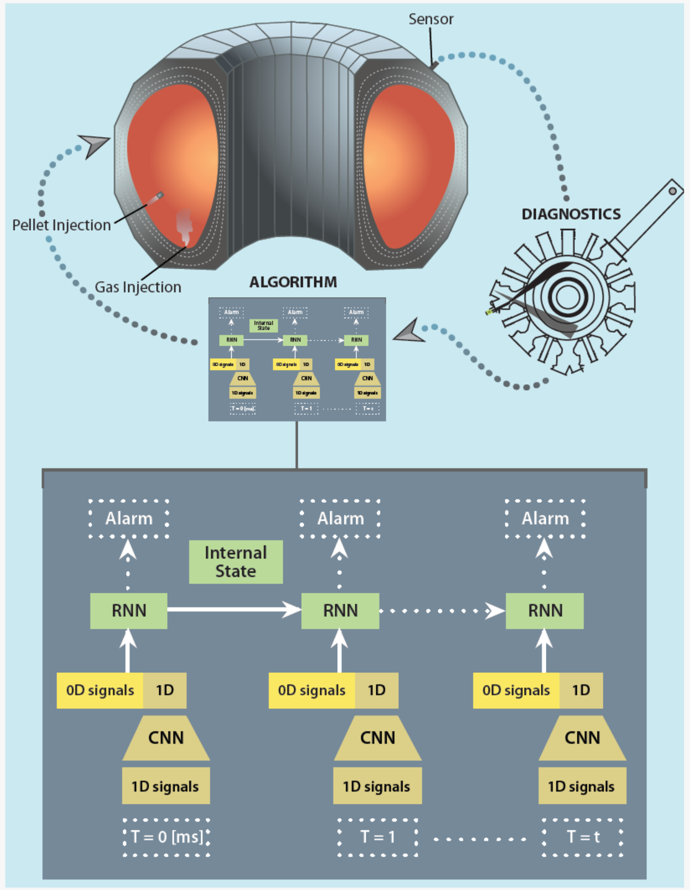
AI FOR SCIENCE APPLICATIONS ON AI TESTBED



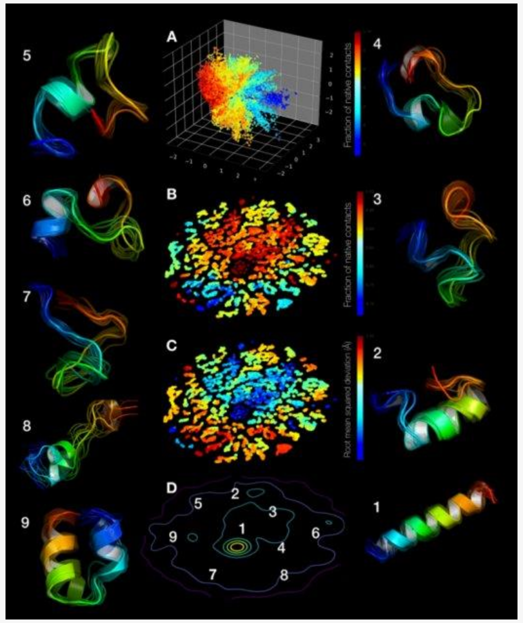
Cancer drug response prediction



Imaging Sciences-Braggs Peak



Tokomak Fusion Reactor operations



Protein-folding(Image: NCI)

and more..

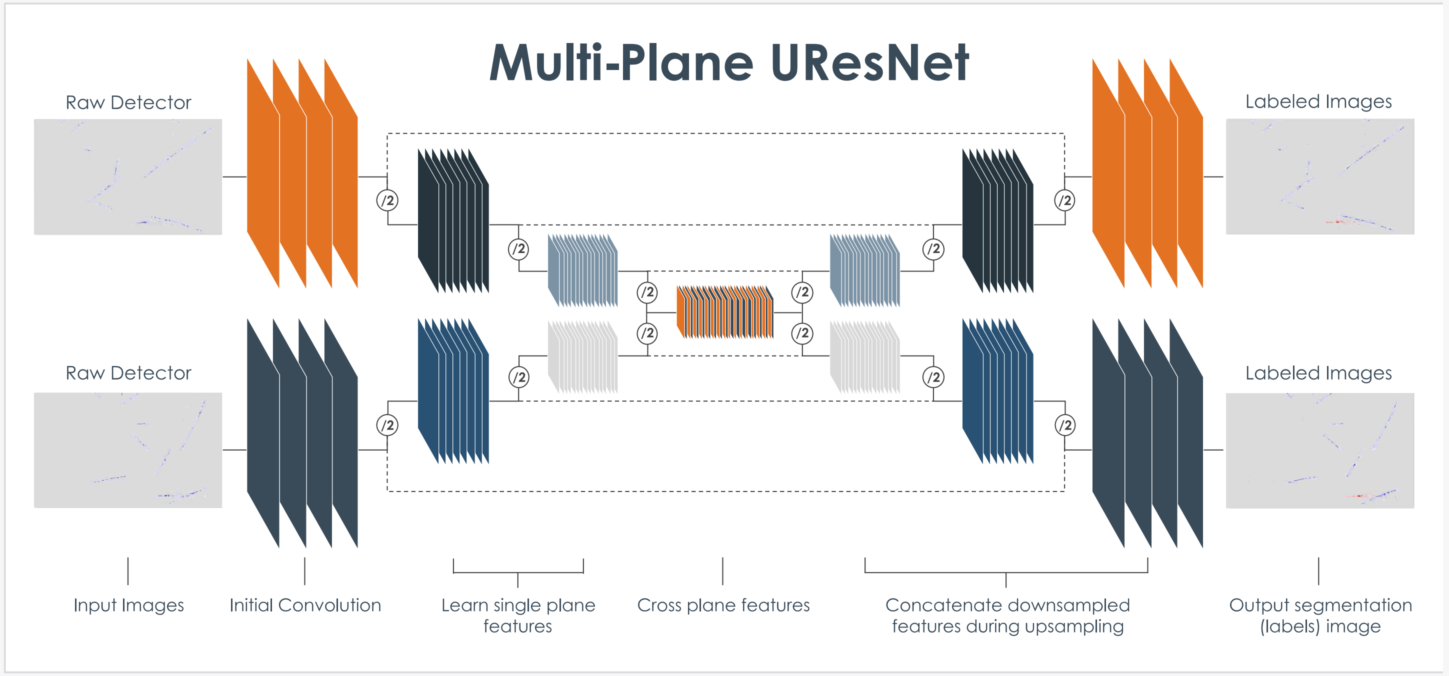
COSMIC TAGGER ON SAMBANOVA DATASCALE

Goal:

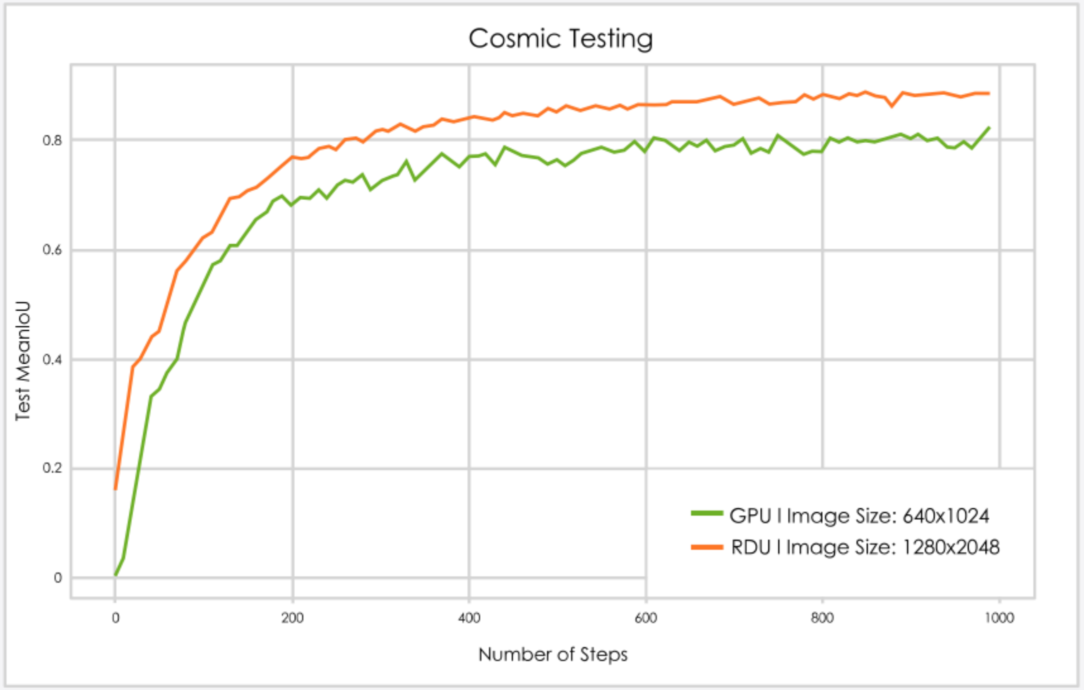
Image segmentation task for liquid argon time projection chamber (LArTPC) detectors in Neutrino Physics experiments to classify each input pixel into one of three classes – Cosmic, Muon, or Background

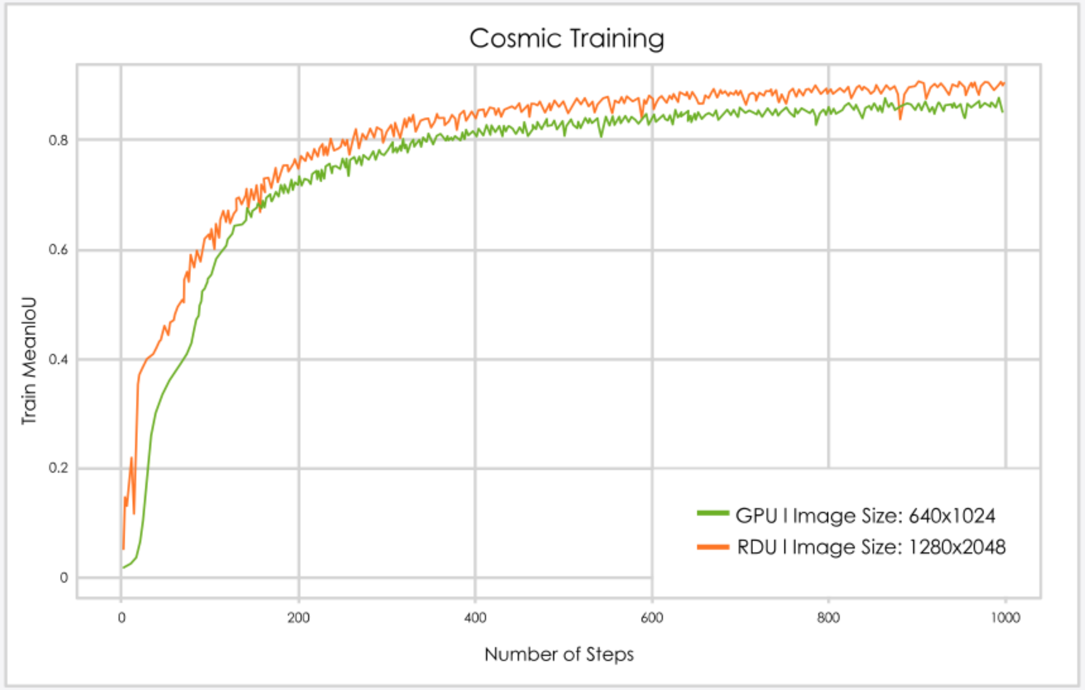
Challenges:

Models and acquired images are limited by the size one can fit on current systems. These are expected to grow with future experiments



Cosmic tagger on Sambanova datascale

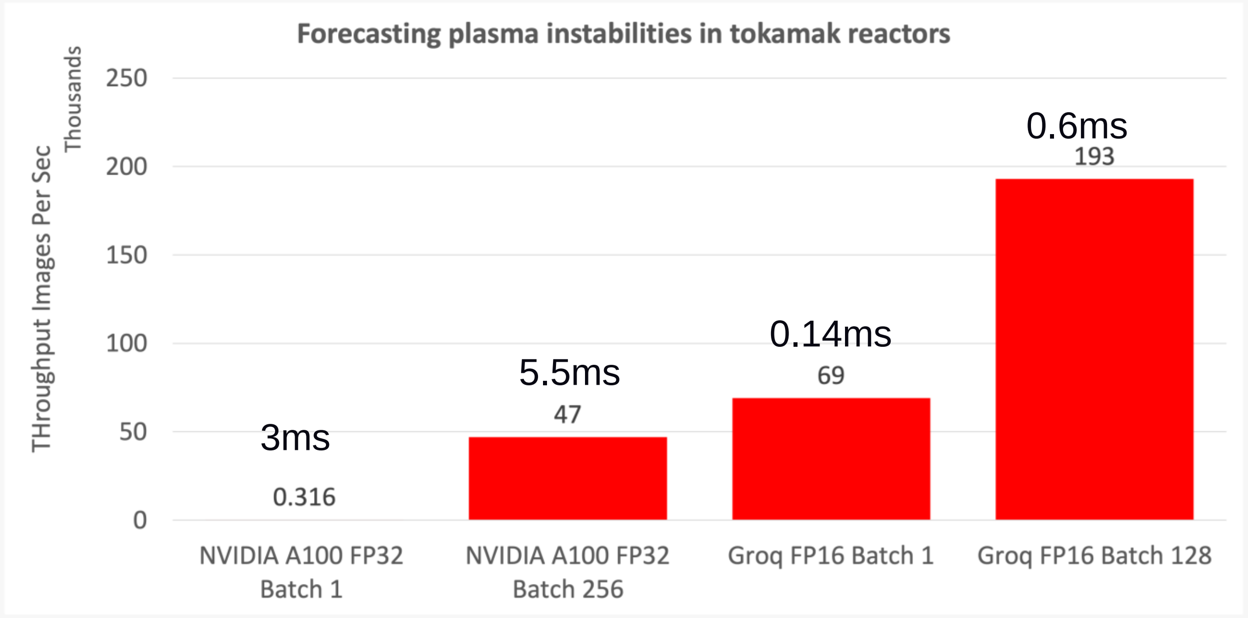




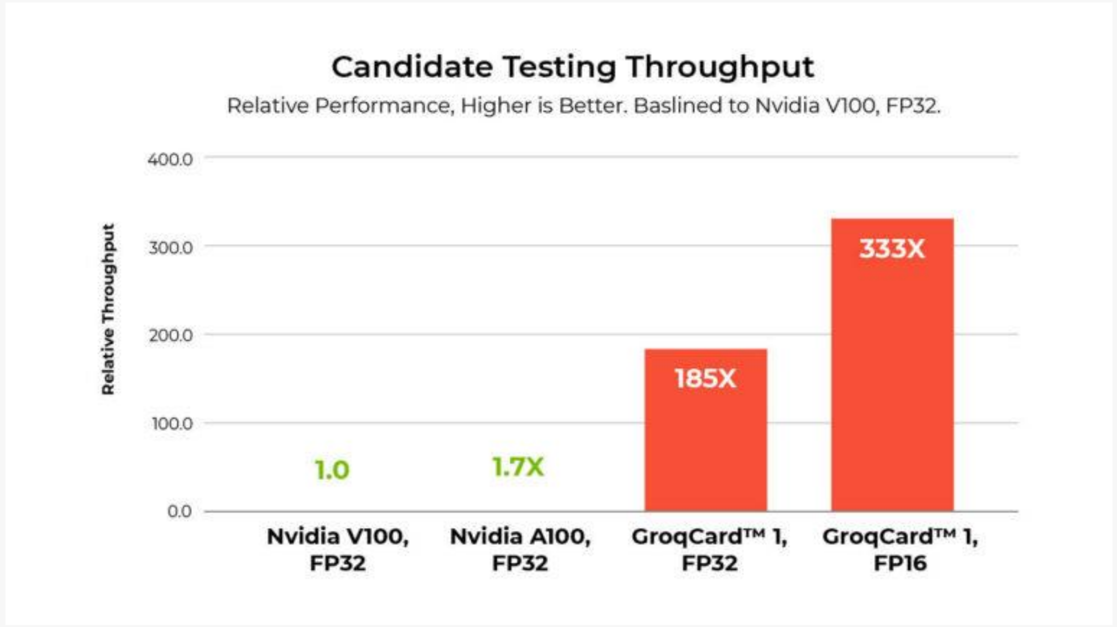
SambaNova RDUs able to accommodate larger image sizes and achieve higher accuracy

M. Emani et al., "Accelerating Scientific Applications With SambaNova Reconfigurable Dataflow Architecture," in Computing in Science & Engineering, vol. 23, no. 2, pp. 114-119, 1 March-April 2021, doi: 10.1109/MCSE.2021.3057203.

Early Experience with Inference on Groq



Forecasting Plasma Instability in Tokamak



COVID19 Candidate drug molecule screening

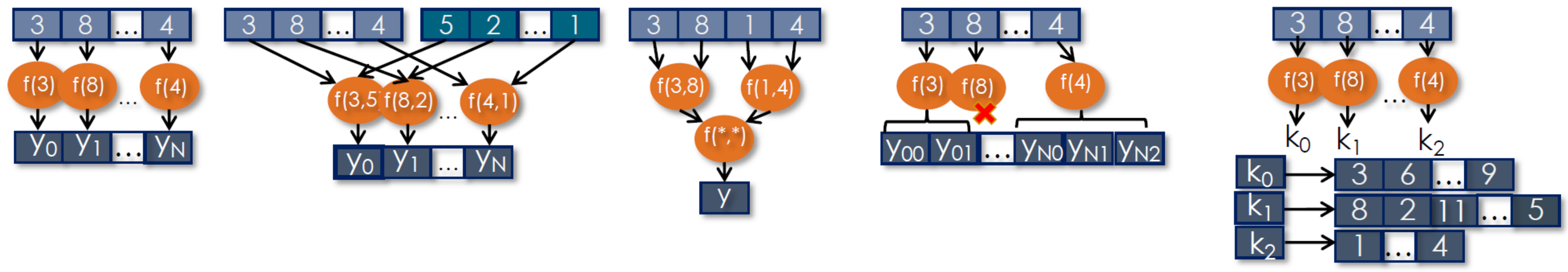
Promising results using GroqChip for science Inference use-cases with respect to latency and throughput in comparison to GPUs

Hardware

Source: White Paper: AI Changes Everything - SambaNova Systems (https://sambanova.ai/wp-content/uploads/2021/06/SambaNova_AIisChangingEverything2021_Whitepaper_English.pdf)

Motivation of hardware design

- A Flexible Dataflow Substrate: Parallel Patterns
  - Looping abstractions with extra information on parallelism & access patterns

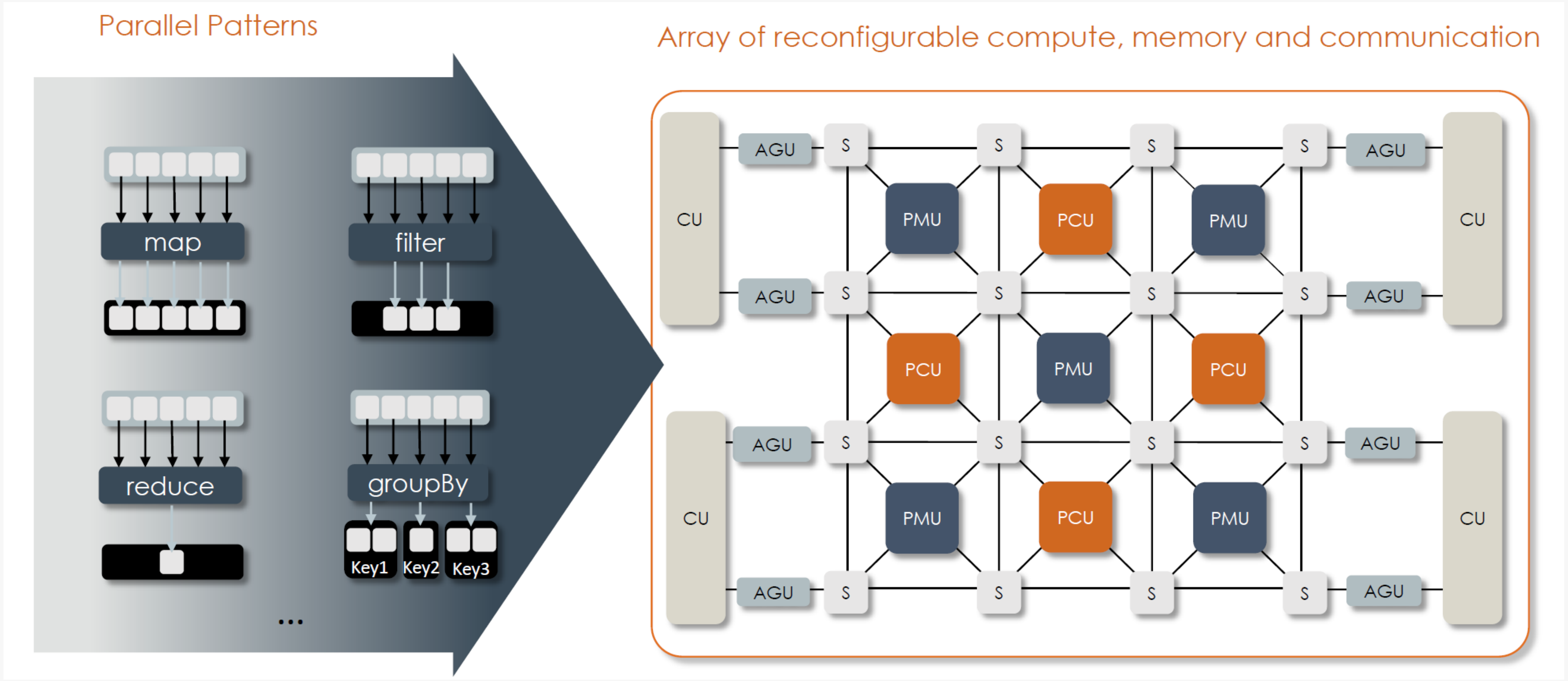


Map	**Zip**	**Reduce**	**FlatMap**	**GroupBy**
element-wise function f	element-wise function f (multi-collection)	combine all elements with f (f is associative)	element-wise function ≥0 values out per element	group elements into buckets based on key

y = vector + 4
y = vector * 10
y = sigmoid(vector)
	y = vecA + vecB
	y = vecA / vecB
	y = max(vecA,vecB)
		y = vector.sum
		y = vector.product
		y = max(vector)
			SELECT * FROM vector
			WHERE elem < 5
				vector.groupBy{e => e % 3}

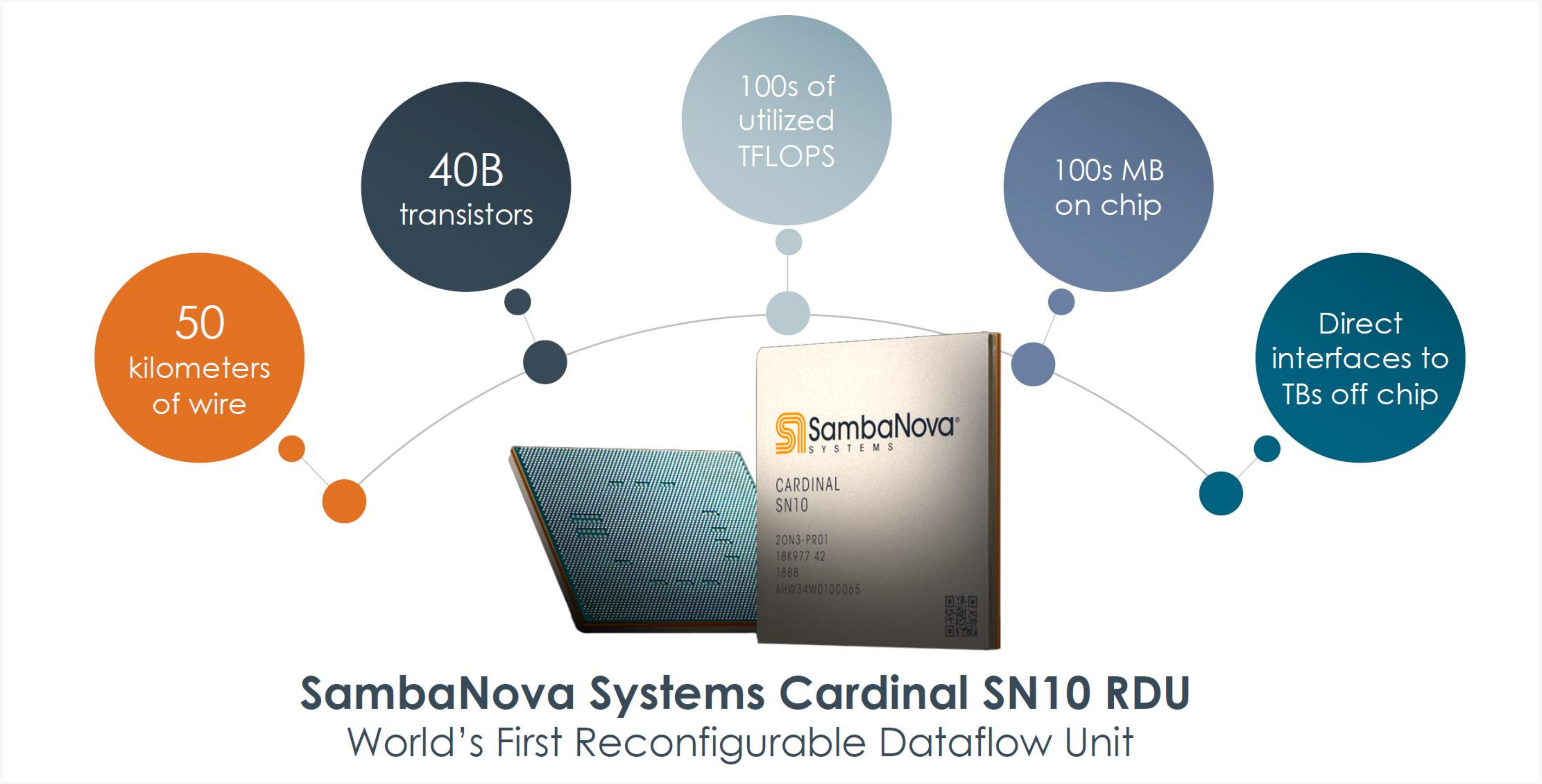
Motivation of hardware design

- Reconfigurable Dataflow Architecture (RDA)



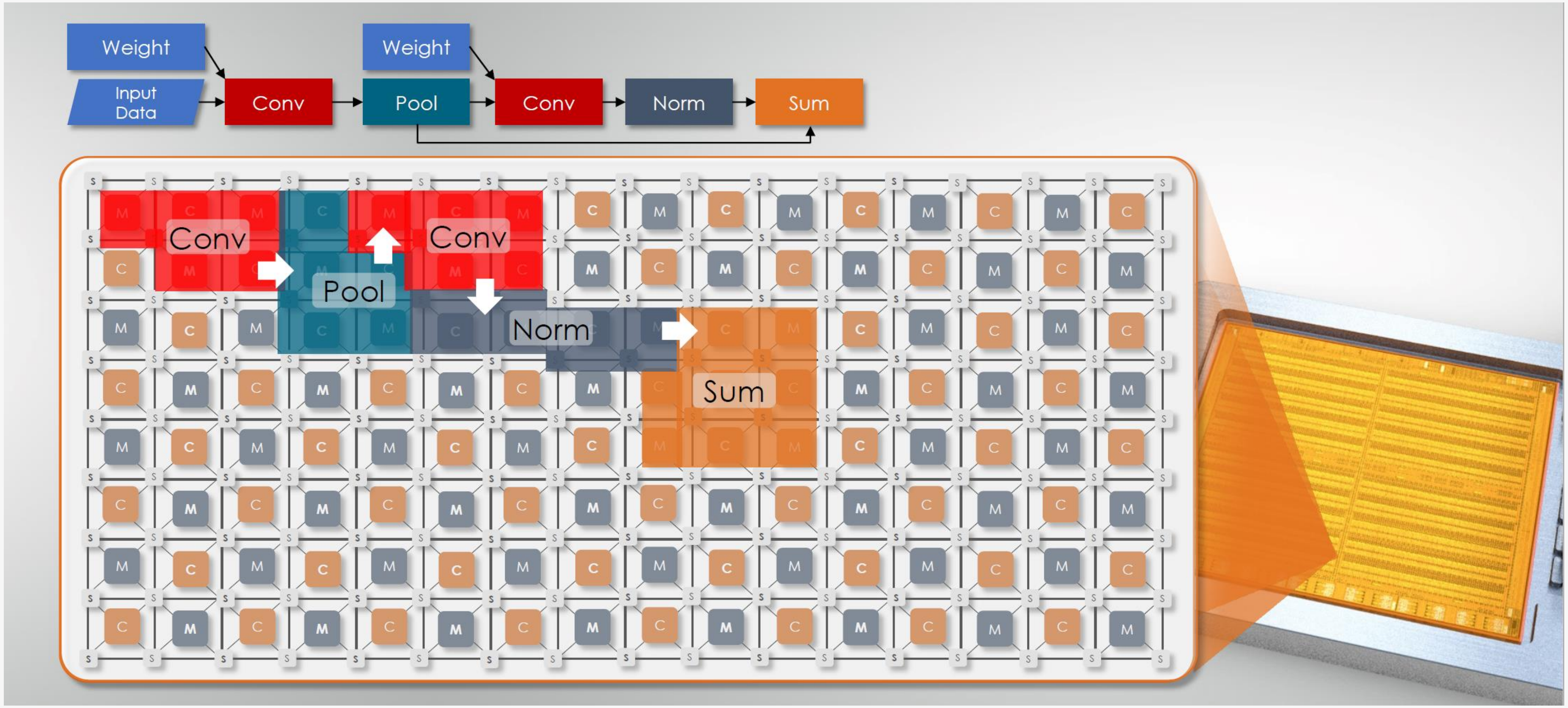
Hardware design

- Reconfigurable Dataflow Architecture (RDU)



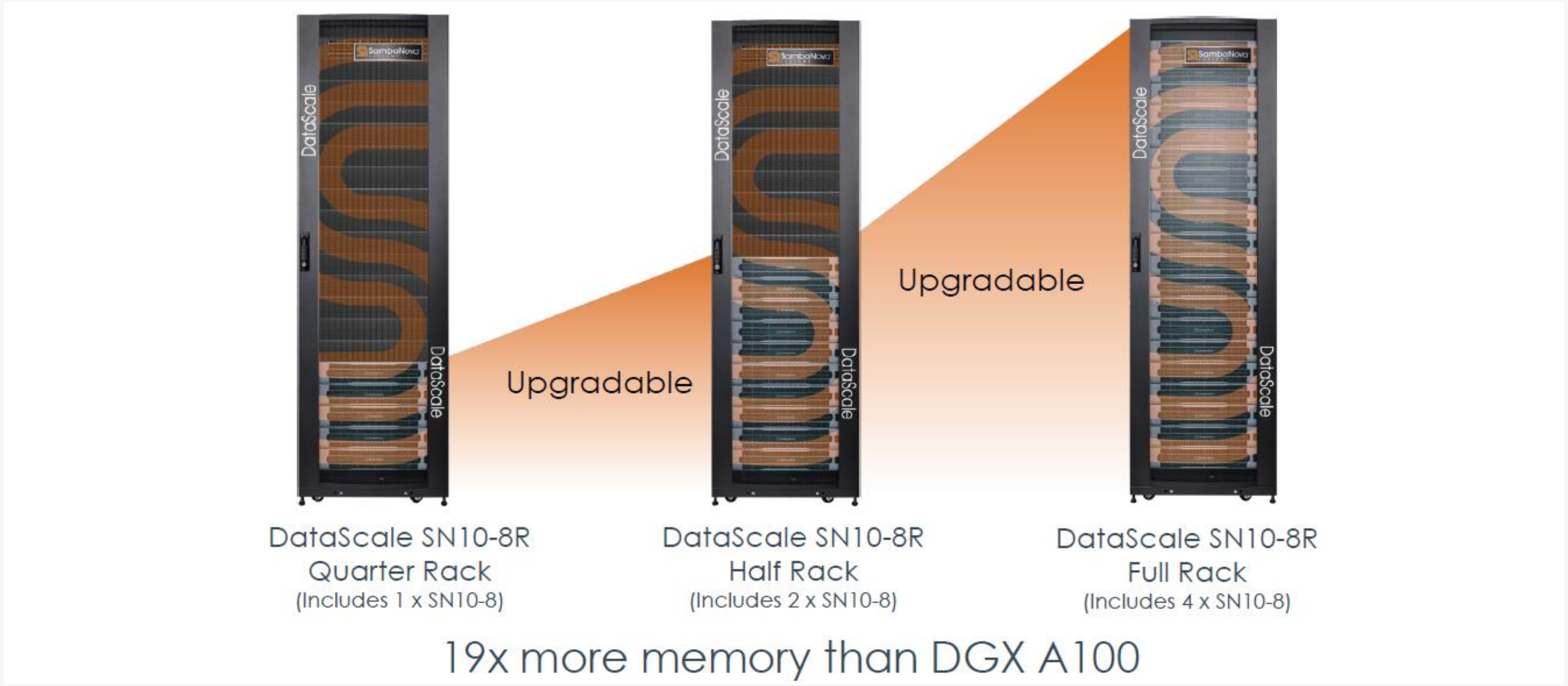
Hardware design

- Rapid Dataflow Compilation to RDU



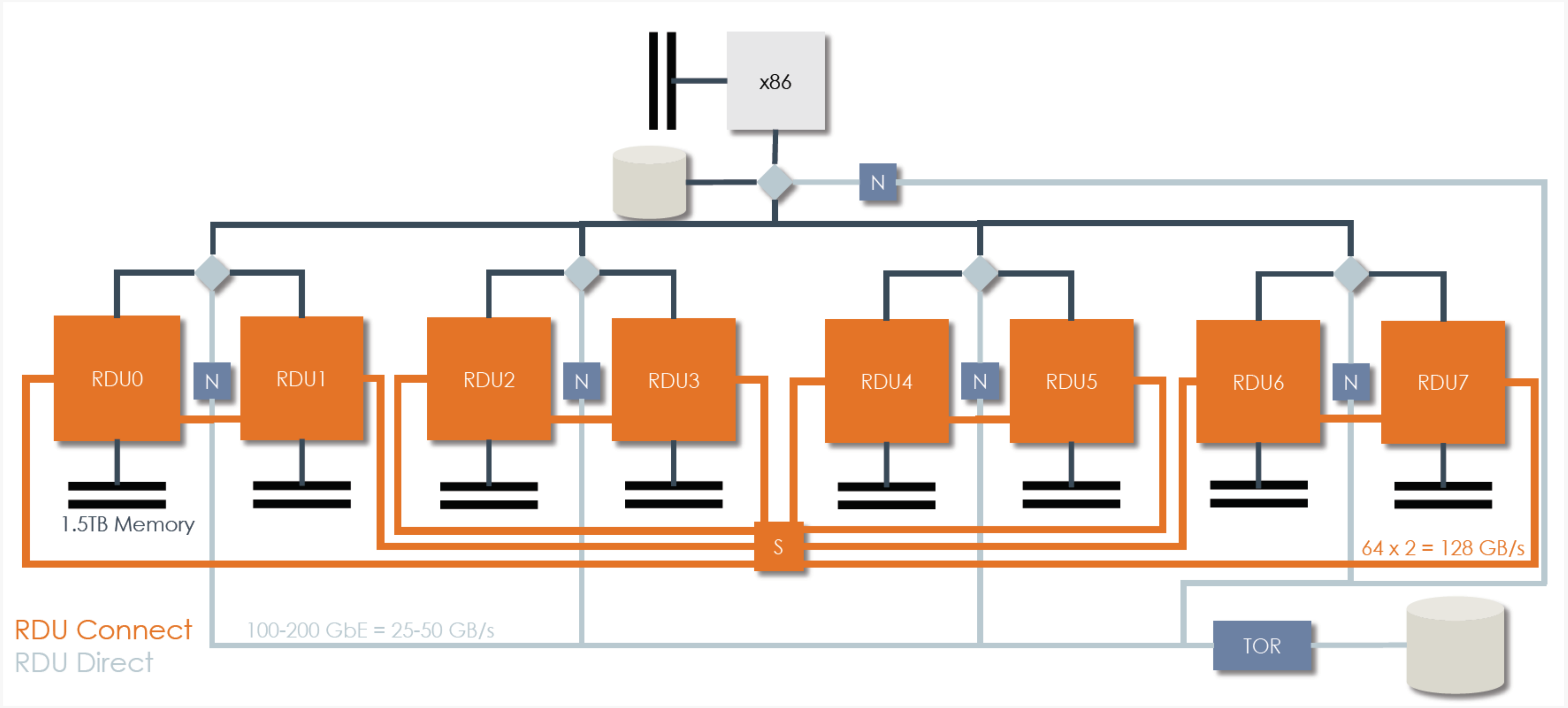
Hardware design

- DataScale SN10-8R: Scalable performance for training and inference



Hardware design

- Excelling at Model and Data Parallel Execution Models

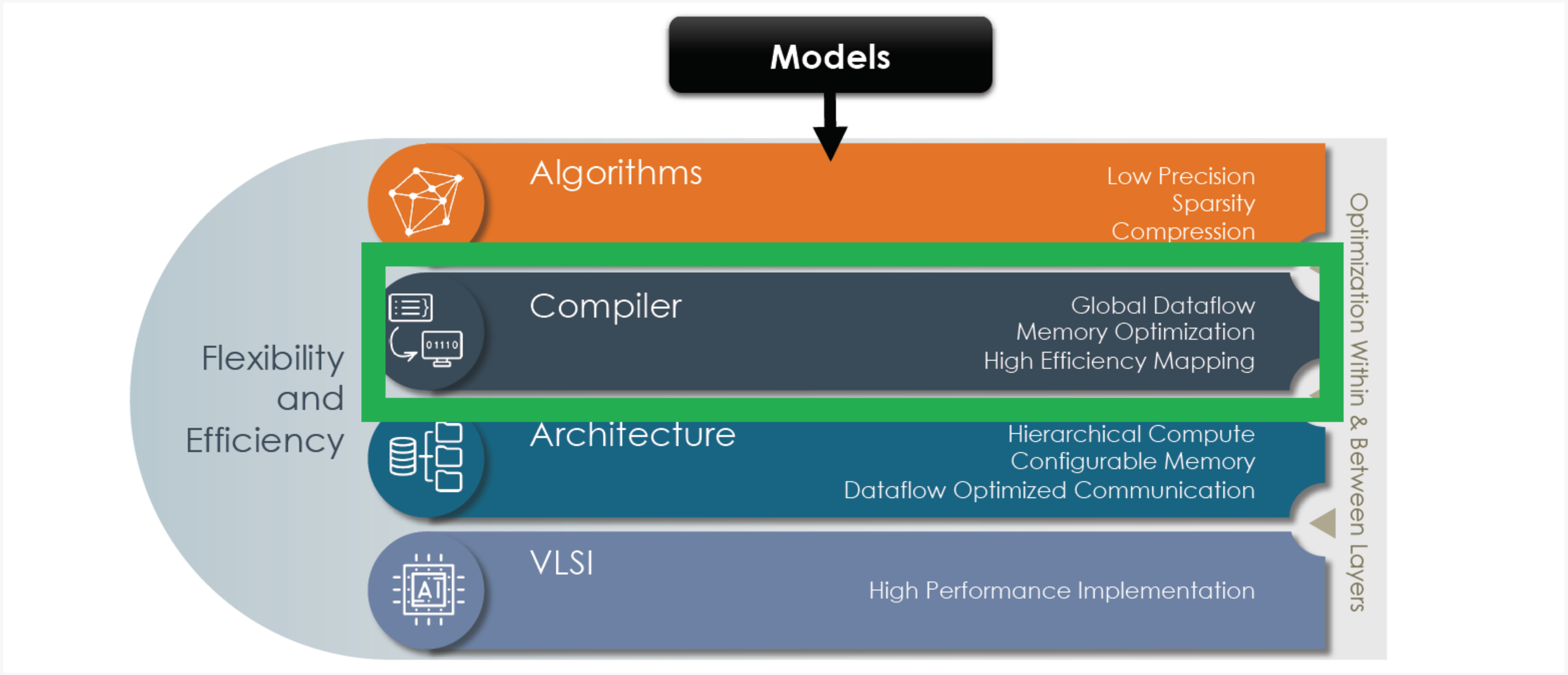


SambaNova™ SYSTEMS

Software

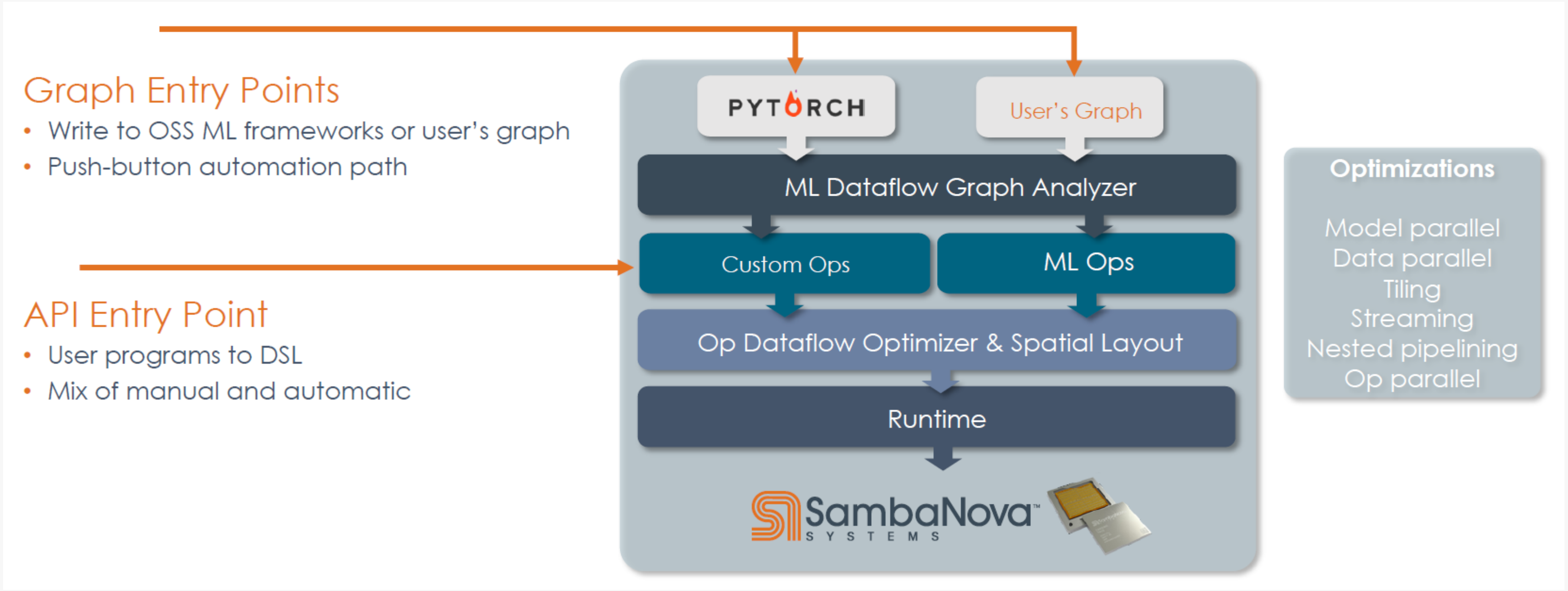
Software design

- Full stack co-engineering yields optimizations where best delivered with the highest impact



Software design

- SambaFlow Open Software for DataScale Systems



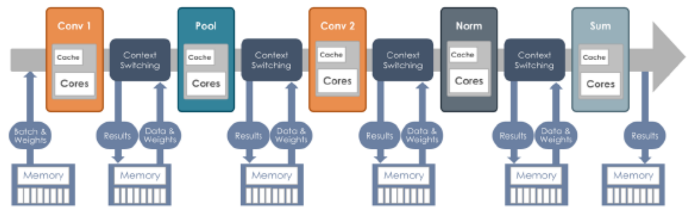
Software design

- SambaFlow Open Software for DataScale Systems

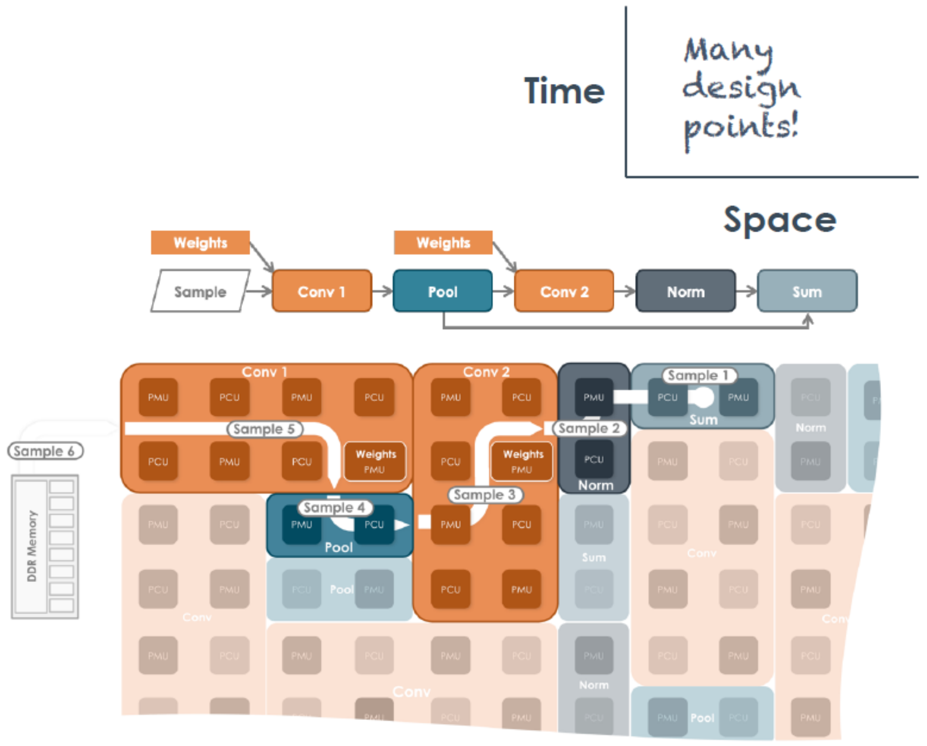
Spatial Dataflow within RDU

```
t1 = conv(in)
t2 = pool(t1)
t3 = conv(t2)
t4 = norm(t3)
t5 = sum(t4)
```

Time: Kernel by Kernel Execution



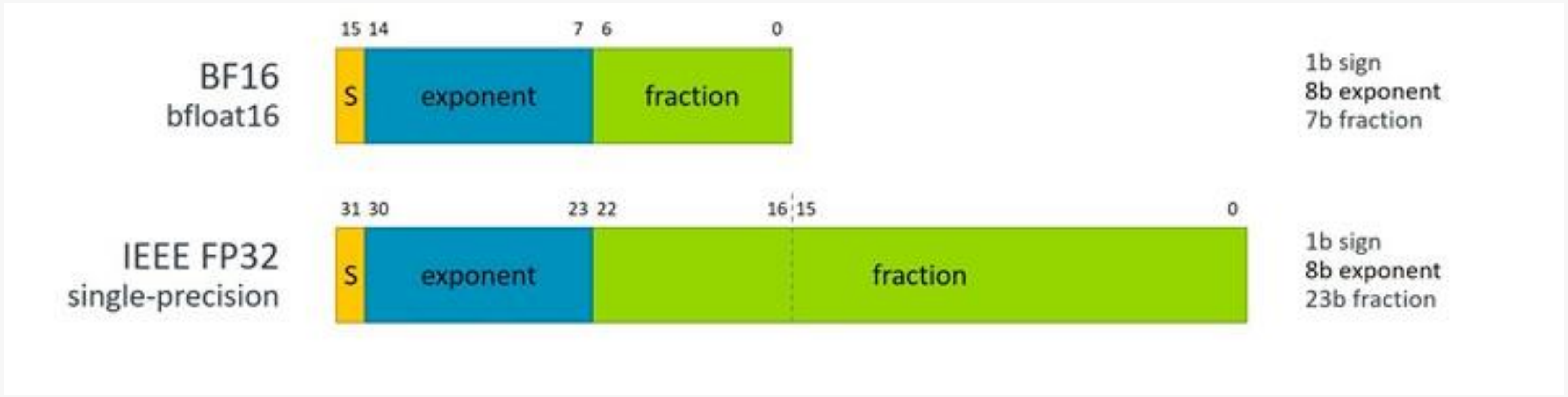
Traditional compilers map operations to processor instructions in time
Communication through the memory hierarchy is implicit and handled by hardware



Dataflow compilers map operations to instructions in time and in space and program the communication between them
SambaFlow eliminates overhead and maximizes utilization

Software design

- A bit about precision
  - The main idea of bfloat16 to provide a 16-bit floating point format that has the same dynamic range as a standard IEEE-FP32, but with less accuracy. That amounted to matching the size of the FP32 exponent field at 8 bits and shrinking the size of the FP32 fraction field down to 7 bits. With bfloat16, SambaNova can provide better training throughput.



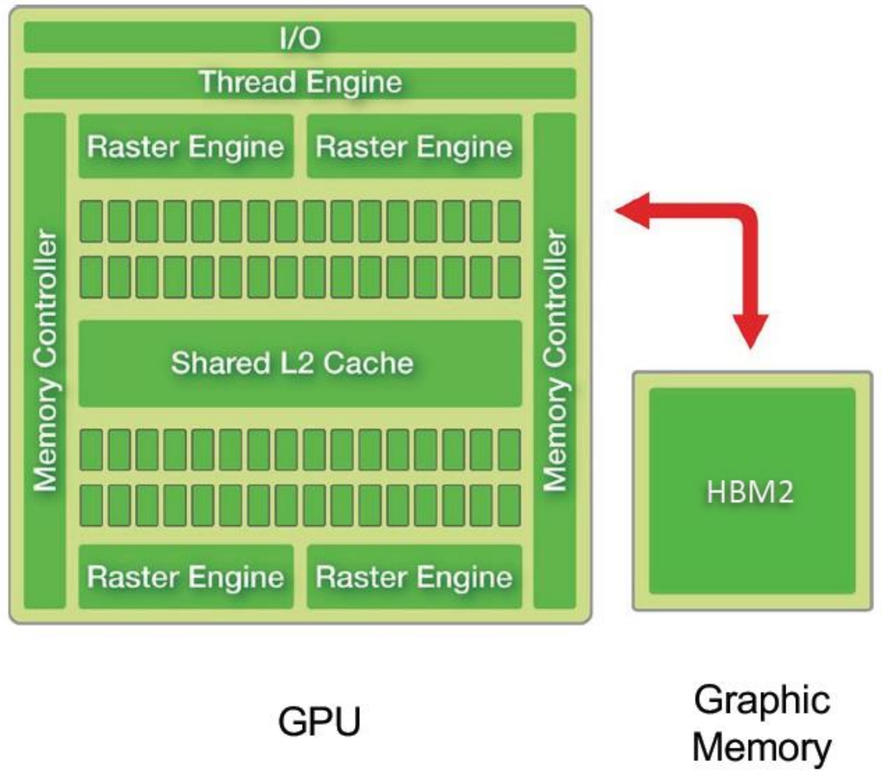
Hardware

Source: White Paper for Cerebras System (https://www.cerebras.net/category/whitepaper/)

Motivation of hardware design

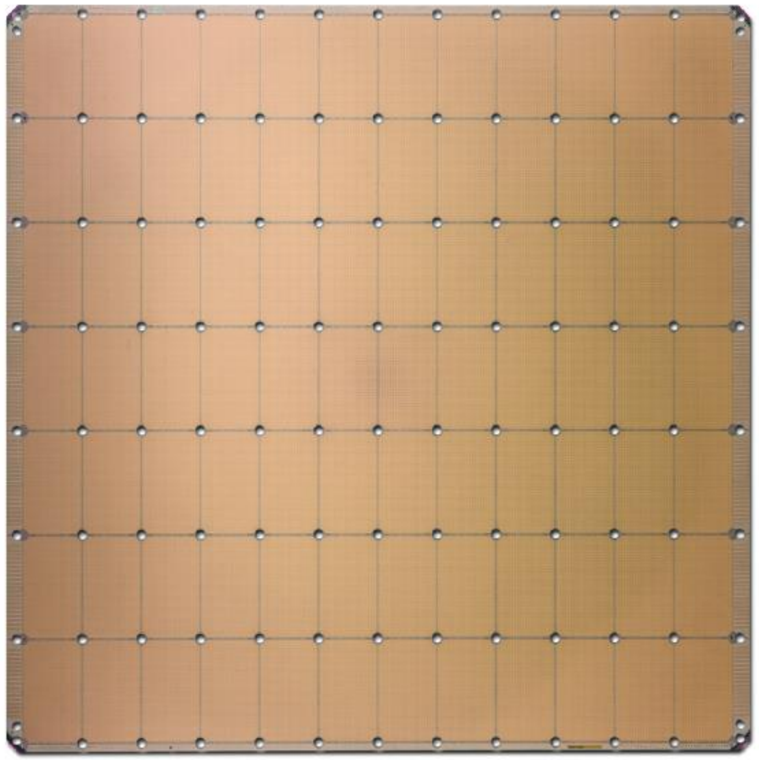
- GPU approach

- < 10% silicon area used for Deep Learning
  - > 90% used for graphics: Raster Engines, Shaders, Texture Maps, Thread and Instruction Control
- Memory is far from graphics core
  - Little on-chip memory
  - Cache memory hierarchy
- Graphics cores not built for communication
  - On chip: low bandwidth, through memory
  - Off chip: even lower bandwidth, PCIe/NVLink
- Designed for dense-matrix operations
  - Sparsity devastates performance
  - Implemented as CPU co-processor



Hardware design

- Cerebras CS-2: The world's only purpose-built Deep Learning solution



Cerebras WSE-2
2.6 Trillion Transistors
46,225 mm² Silicon

Largest GPU
54.2 Billion Transistors
826 mm² Silicon

Cerebras Wafer Scale Engine (WSE)

The Most Powerful Processor for AI

400,000 AI-optimized cores

46,225 mm^2 silicon

1.2 trillion transistors

18 Gigabytes of On-chip Memory

9 PByte/s memory bandwidth

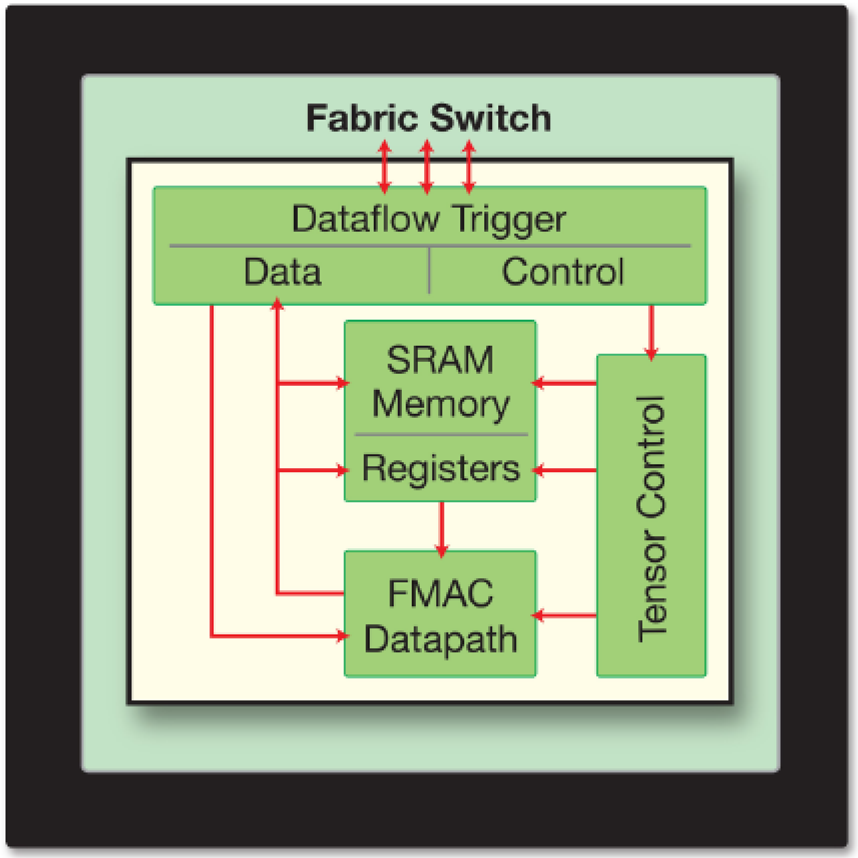
100 Pbit/s fabric bandwidth

TSMC 16nm process

Hardware design

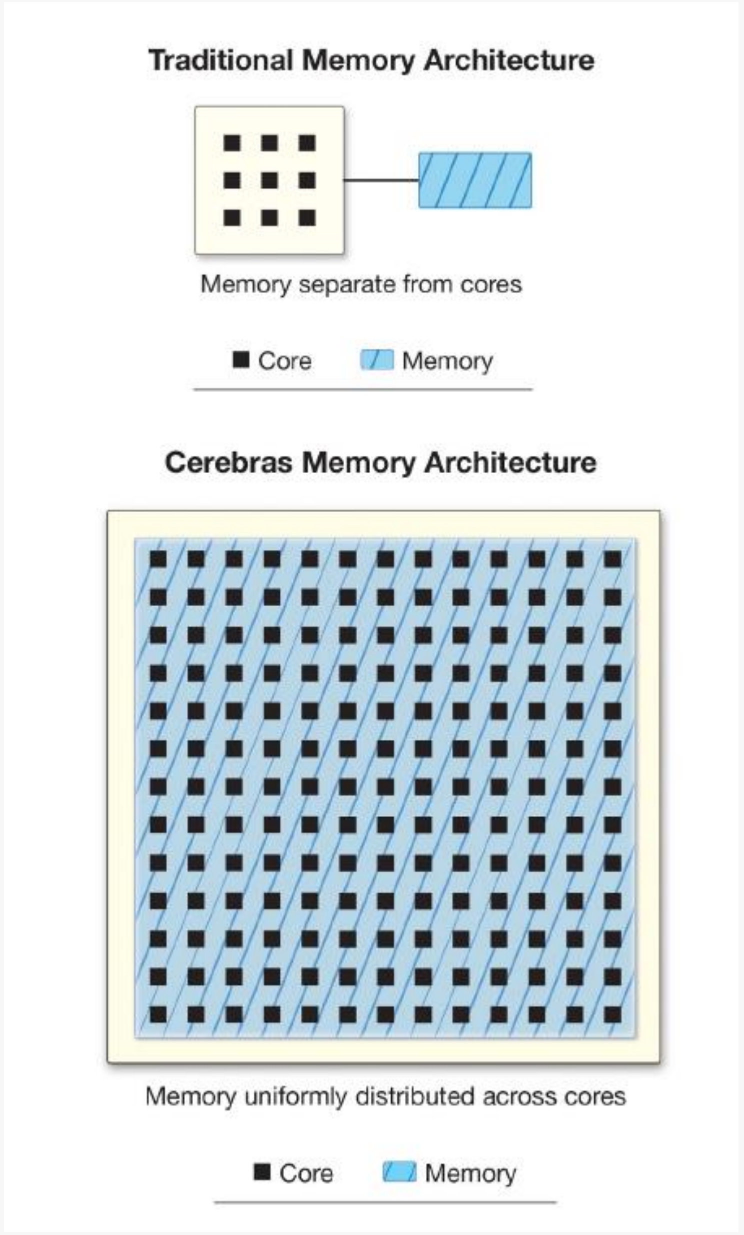
- The CS WSE architecture is built for deep learning

- AI-optimized **compute**
  - Fully-programmable core, ML-optimized extensions
    - e.g. arithmetic, logical, load/store, branch
  - Dataflow architecture optimized for sparse, dynamic workloads
    - Higher performance and efficiency for sparse NN



Hardware design

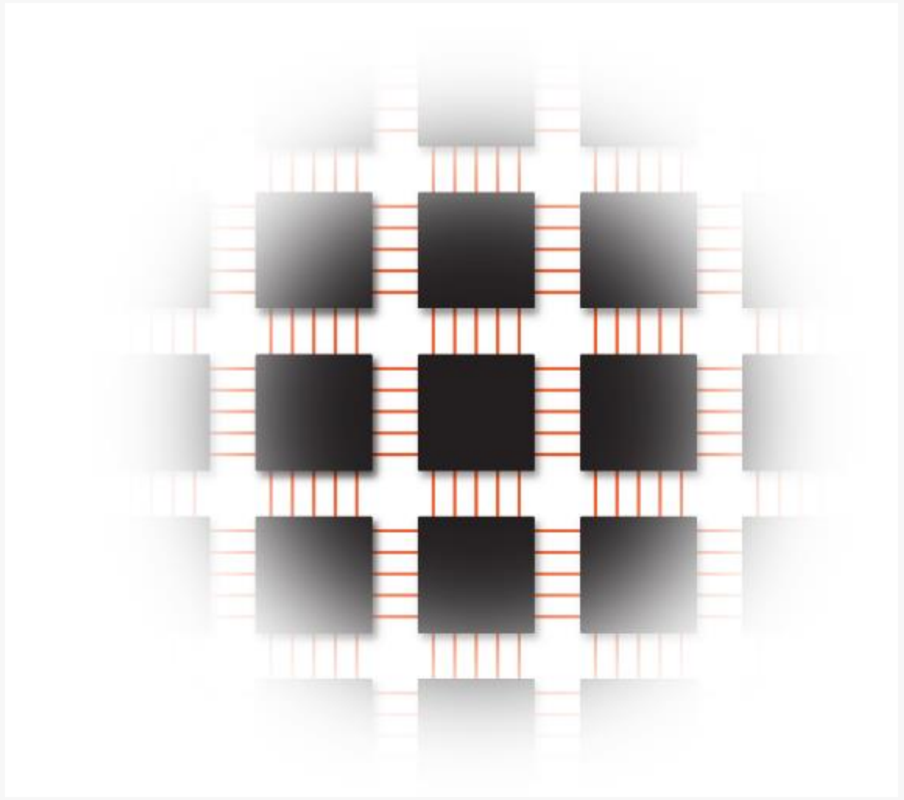
- The CS WSE architecture is built for deep learning
- AI-optimized **memory**
  - Traditional memory architectures shared memory far from compute
  - The right answer is distributed, high performance, on-chip memory


Traditional Memory Architecture
Memory separate from cores
■ Core
▨ Memory
Cerebras Memory Architecture
Memory uniformly distributed across cores
■ Core
▨ Memory

Hardware design

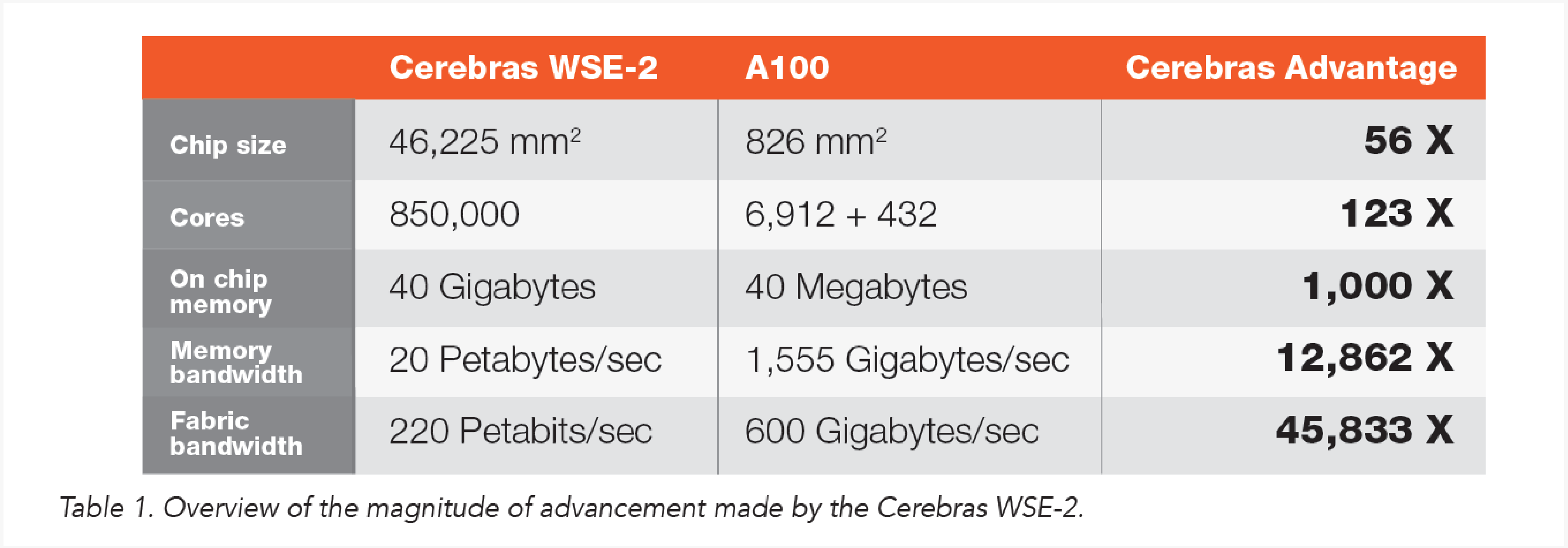
- The CS WSE architecture is built for deep learning

- AI-optimized **communication**
  - High bandwidth, low latency cluster-scale networking on chip
  - Fully-configurable to user-specified topology



Hardware design

- Cerebras CS-2: Comparison with NVIDIA A100 GPU



	Cerebras WSE-2	A100	Cerebras Advantage
Chip size	46,225 mm^2	826 mm^2	56 X
Cores	850,000	6,912 + 432	123 X
On chip memory	40 Gigabytes	40 Megabytes	1,000 X
Memory bandwidth	20 Petabytes/sec	1,555 Gigabytes/sec	12,862 X
Fabric bandwidth	220 Petabits/sec	600 Gigabytes/sec	45,833 X

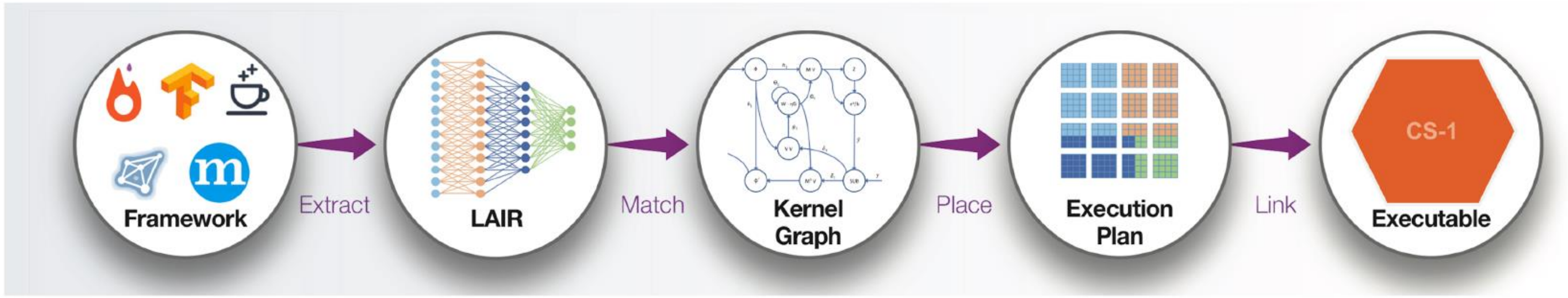
Table 1. Overview of the magnitude of advancement made by the Cerebras WSE-2.

cerebras

Software

Software design

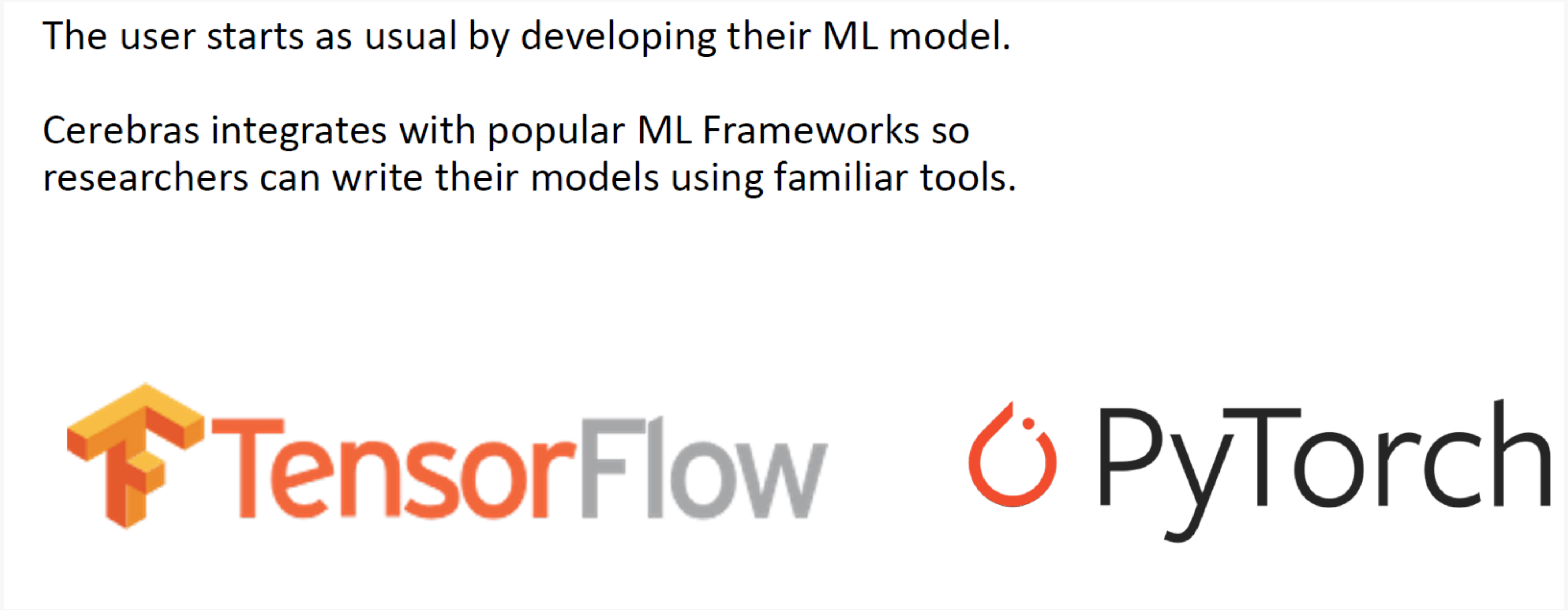
- Cerebras Software Stack handles graph compilation



- **Extract** – Obtain graph representation of model from framework and express it in our intermediate form.
- **Match** – Consult kernel library for kernels that implement portions of model.
- **Place & Route** – Assign kernels to regions of fabric guided by graph connectivity and kernel performance functions.
- **Link** – Create executable output that can be loaded and run by CS-1.

Software design

- Program using familiar ML Frameworks



The user starts as usual by developing their ML model.

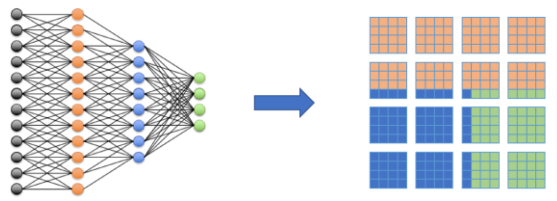
Cerebras integrates with popular ML Frameworks so researchers can write their models using familiar tools.

Software design

- Model extraction from ML Framework -> LAIR

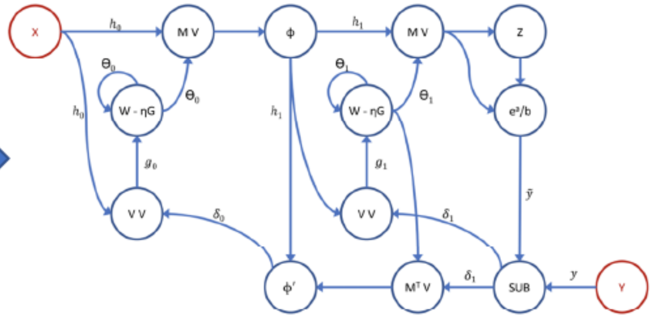
Cerebras LAIR (**L**inear **A**lgebra **I**ntermediate **R**epresentation) is the standard input into the Cerebras software stack.

We extract the explicit linear algebra graph representation of the model from the ML Framework and translate it into LAIR.

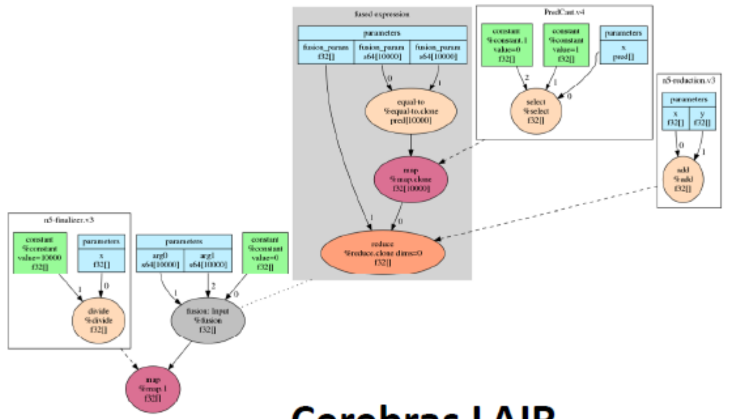


TensorFlow

PyTorch

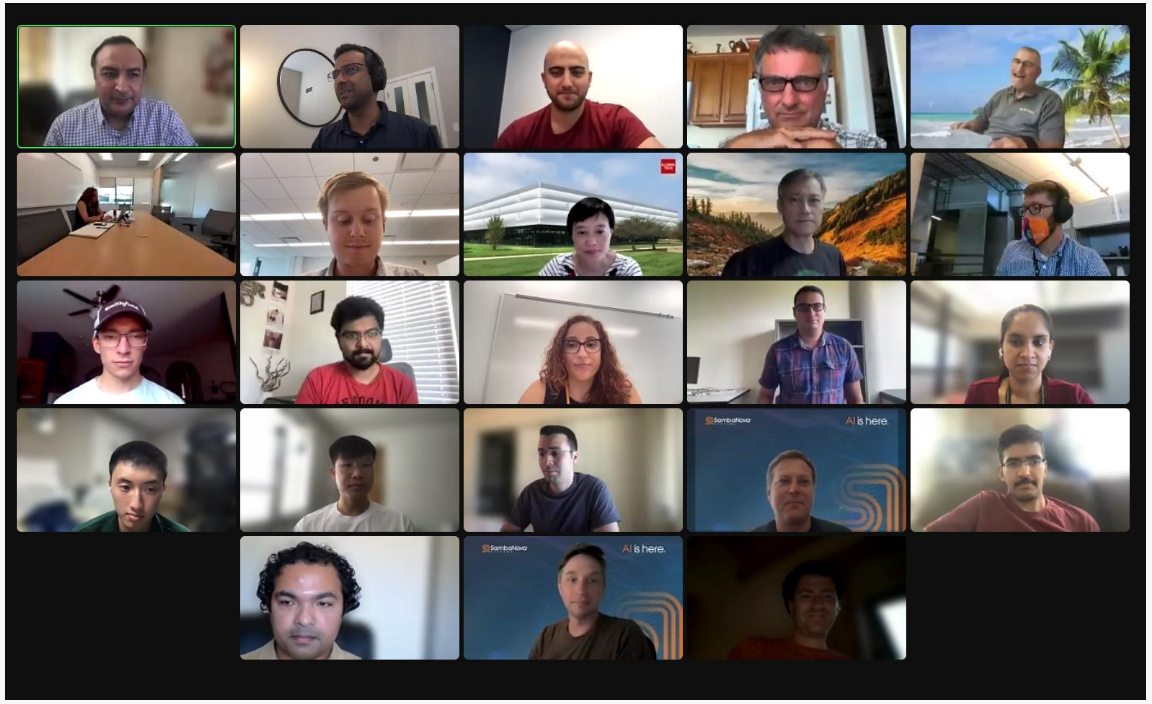


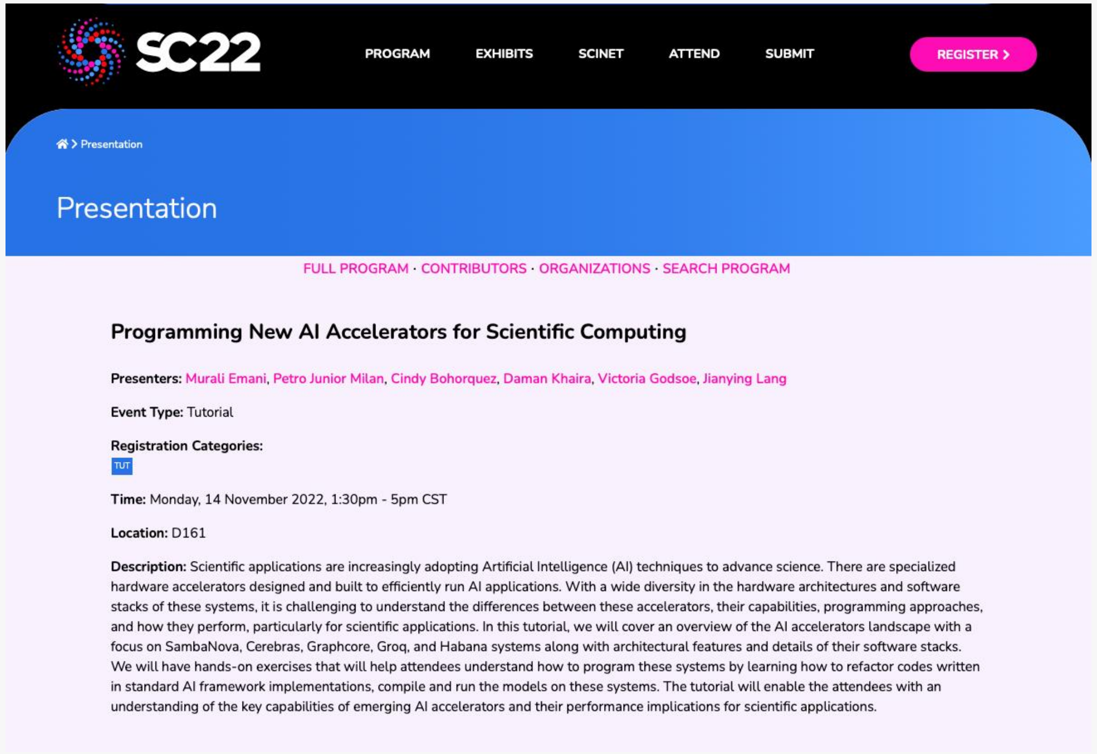
Framework IR



Cerebras LAIR

AI Testbed Community Engagement





FULL PROGRAM · CONTRIBUTORS · ORGANIZATIONS · SEARCH PROGRAM

Programming New AI Accelerators for Scientific Computing

Presenters: Murali Emani, Petro Junior Milan, Cindy Bohorquez, Daman Khaira, Victoria Godsoe, Jianying Lang

Event Type: Tutorial

Registration Categories:
TUT

Time: Monday, 14 November 2022, 1:30pm - 5pm CST

Location: D161

Description: Scientific applications are increasingly adopting Artificial Intelligence (AI) techniques to advance science. There are specialized hardware accelerators designed and built to efficiently run AI applications. With a wide diversity in the hardware architectures and software stacks of these systems, it is challenging to understand the differences between these accelerators, their capabilities, programming approaches, and how they perform, particularly for scientific applications. In this tutorial, we will cover an overview of the AI accelerators landscape with a focus on SambaNova, Cerebras, Graphcore, Groq, and Habana systems along with architectural features and details of their software stacks. We will have hands-on exercises that will help attendees understand how to program these systems by learning how to refactor codes written in standard AI framework implementations, compile and run the models on these systems. The tutorial will enable the attendees with an understanding of the key capabilities of emerging AI accelerators and their performance implications for scientific applications.

- SambaNova AI training workshop – July 19-20, 2022
- ATPESC H/W Architecture Day on August 1, 2022 covered five AI accelerators
- ALCF AI for Science training series for students in the fall will include the AI testbed
- Cerebras CS-2 training workshop held for August 2022

SC'22 Tutorial on Programming AI accelerators for Scientific Computing ***in collaboration with Cerebras, Intel Habana, Graphcore, Groq and SambaNova accepted***

Getting Started on ALCF AI Testbed:

Apply for a Director’s Discretionary (DD) Allocation Award

Director’s Discretionary (DD) awards support various project objectives from scaling code to preparing for future computing competition to production scientific computing in support of strategic partnerships.

Cerebras CS-2 and SambaNova Datascale are available for allocations

Allocation Request Form

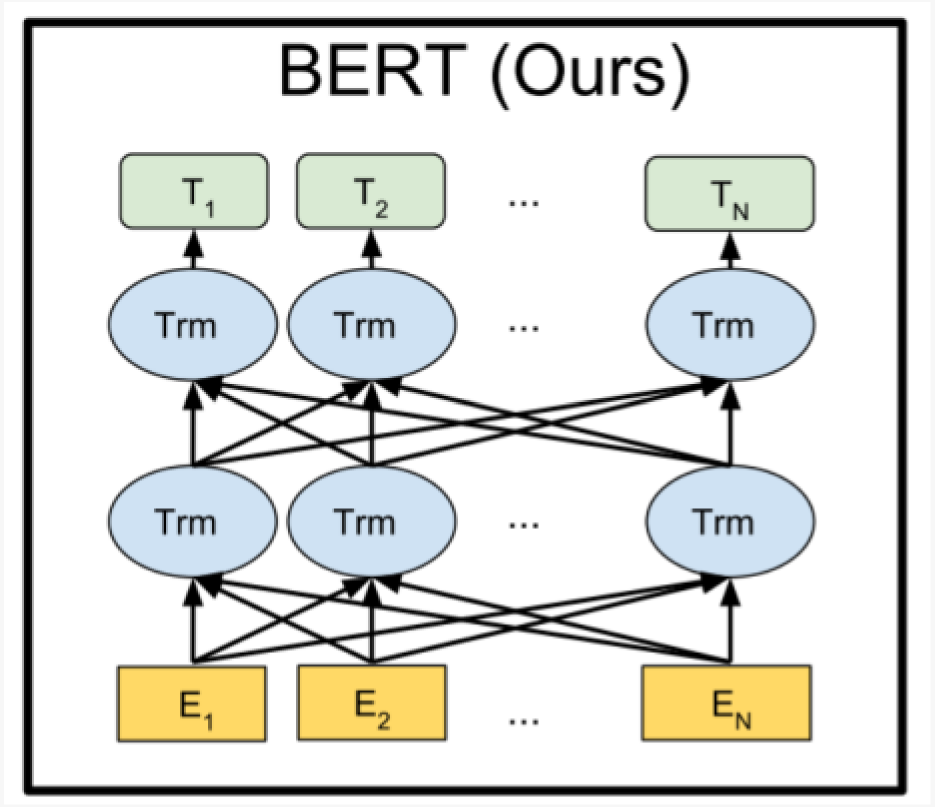
AI Testbed User Guide

Hands-on Section on SambaNova and Cerebras

BERT (language model) on hands-on section

- Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google.

- The original English-language BERT has two models:
  - (1) BERT_BASE: 12 encoders with 12 bidirectional self-attention heads;
  - (2) BERT_LARGE: 24 encoders with 16 bidirectional self-attention heads.



SambaNova

- 1. Login to sn:

```
ssh ALCFUserID@sambanova.alcf.anl.gov
ssh sm-01
```

- 2. SDK setup:

```
source /software/sambanova/envs/sn_env.sh
```

- 3. Copy scripts:

```
cp
/var/tmp/Additional/slurm/Models/ANL_Acceptance_RC1_11_5
/bert_train-inf.sh ~/
```

- 4. Run scripts:

```
cd ~; ./bert_train-inf.sh;
```

Cerebras

- 1. Login to CS-2:

```
ssh ALCFUserID@cerebras.alcf.anl.gov
ssh cs2-01-med1
```

- 2. Copy scripts:

```
cp -r /software/cerebras/model_zoo ~/
cd modelzoo/transformers/tf/bert
modify data_dir to
"/software/cerebras/dataset/bert_large/msl128/" in
configs/params_bert_large_msl128.yaml
```

Cerebras

3. Run scripts:

```
MODELDIR=model_dir_bert_large_msl128_$(hostname)
rm -r $MODELDIR
time -p csrun_cpu python run.py --mode=train --
compile_only --params
configs/params_bert_large_msl128.yaml --model_dir
$MODELDIR --cs_ip $CS_IP
time -p csrun_wse python run.py --mode=train --params
configs/params_bert_large_msl128.yaml --model_dir
$MODELDIR --cs_ip $CS_IP
```

Thanks!

Zhen Xie, Murali Emani, Siddhisanket Raskar, Varuni Sastry, William Arnold, Bruce Wilson, Rajeev Thakur, Venkatram Vishwanath

Argonne Leadership Computing Facility

2022 Argonne Training Program on Extreme-Scale Computing (ATPESC)