

ARGONNE
ATPESC2026
EXTREME - SCALE COMPUTING

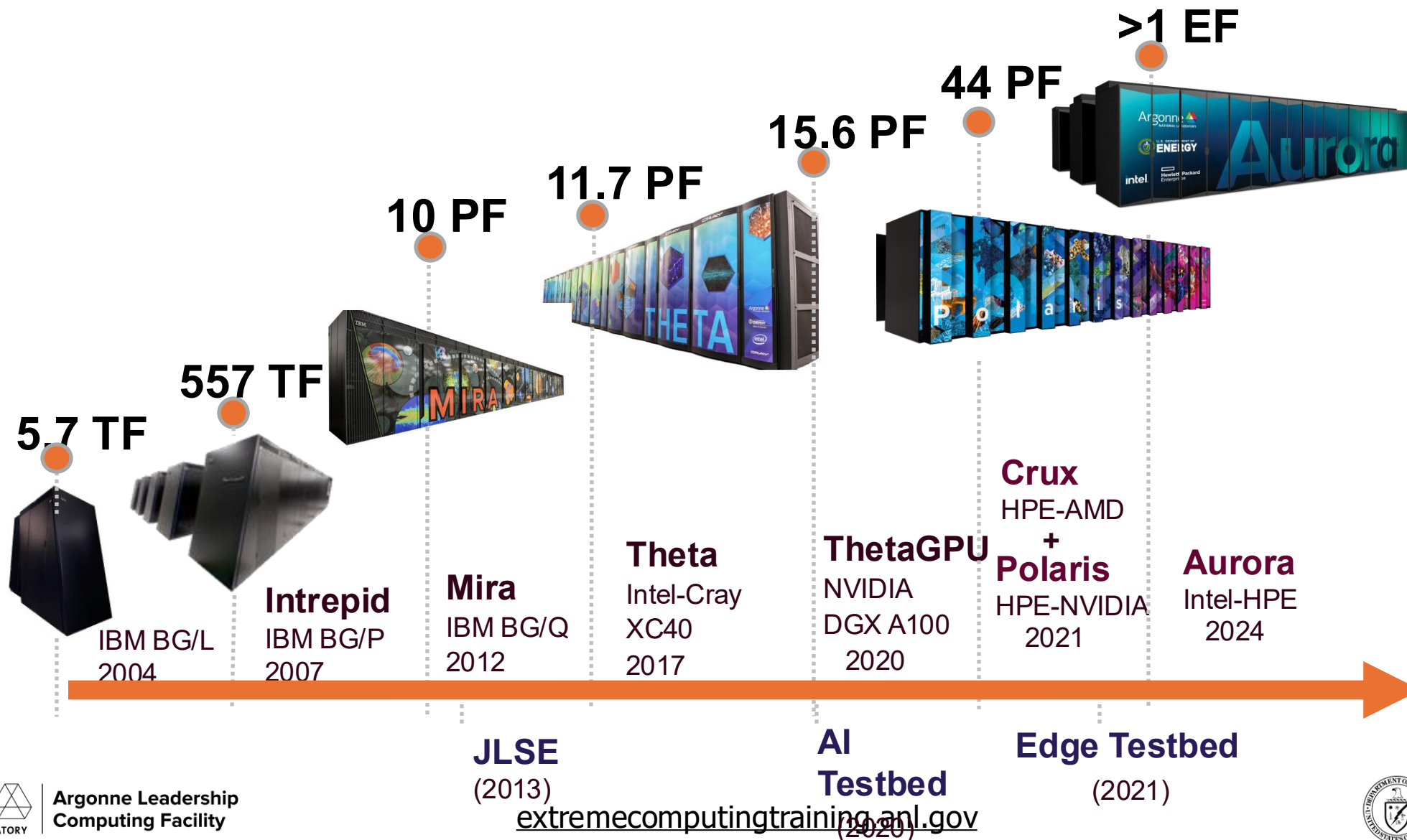
ALCF AI Testbed

08/04/2026

Varuni Sastry (vsastry@anl.gov)

Murali Emani (memani@anl.gov)

ALCF Systems Evolution



ALCF AI Testbed

- 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

ALCF AI Testbed Deployments



Cerebras (CS-3)

4x CS-3 nodes

Supports:

AI training and inference of massive AI models, CSL-SDK

Status:

Upgraded software stack. In production for pre-training and finetuning.



Metis

SambaNova SN-40L nodes

Supports:

AI Inference for models developed by Argonne researchers.

Status:

4x nodes being integrated with ALCF Inference service. Adding 2 nodes for a total of 6x.



Tenstorrent

2x Galaxy Nodes (64 wormhole cards)

Supports:

AI Inference (one node) and SDK development (one node).

Status:

Two nodes configured and available to Argonne users. Will be made available to ALCF users.



GroqRack

72 accelerators (9 nodes x 8 accelerators per node)

Supports:

AI inference using Language Processing Unit (LPU) technology.

Status:

Available for allocation requests.



Graphcore Bow Pod64

64 accelerators (4 nodes x 16 accelerators per node)

Supports:

ML and AI workloads.

Status:

Available for allocation requests.

ALCF Documentation

<https://docs.alcf.anl.gov/ai-testbed/getting-started/>

Argonne Leadership Computing Facility

ALCF Resources Science and Engineering Community and Outreach About Support

NEWS EVENTS PEOPLE CAREERS MyALCF

Search

Machines > AI Testbed

ALCF User Guides

Getting Started

Contribute to User Guides

User Support

Submit a Ticket

Get Help & Connect

Machines

Aurora

AI Testbed

Cerebras

Graphcore

Groq

SambaNova

Data Management

Crux

Polaris

Sophia

Running Jobs with PBS

Example Job Scripts

Machine Reservations

Data Management

File System & Storage

Transfer & Sharing

Services

Inference Endpoints

JupyterHub

Continuous Integration

ALCF AI Testbed



The [ALCF AI Testbed](#) houses some of the most advanced AI accelerators for scientific research. The goal of the testbed is to enable explorations into next-generation machine learning applications and workloads, enabling the ALCF and its user community to help define the role of AI accelerators in scientific computing and how to best integrate such technologies with supercomputing resources. The AI accelerators complement the ALCF's current and next-generation supercomputers to provide a state-of-the-art computing environment that supports pioneering research at the intersection of AI, big data, and high performance computing (HPC). The platforms are equipped with architectural features that support AI and data-centric workloads, making them well suited for research tasks involving the growing deluge of scientific data produced by powerful tools, such as supercomputers, light sources, telescopes, particle accelerators, and sensors. In addition, the testbed will allow researchers to explore novel workflows that combine AI methods with simulation and experimental science to accelerate the pace of discovery.

Table of contents

How to Get Access

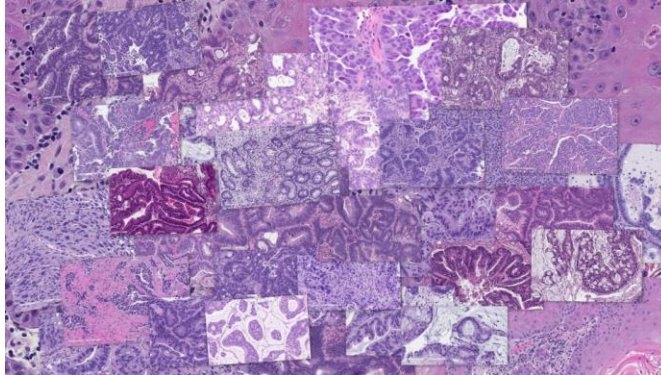
Getting Started

How to Contribute to Documentation

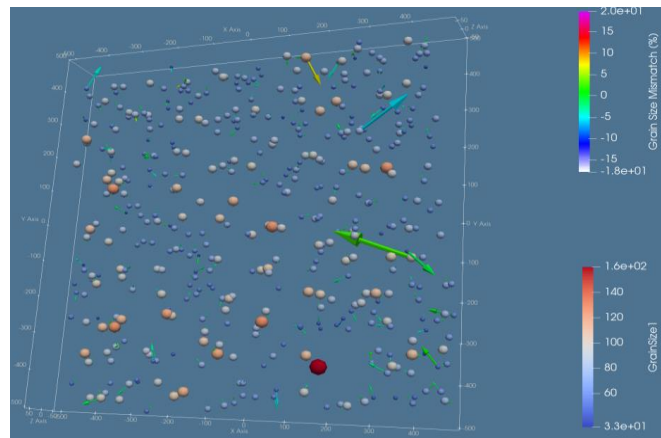
Does this doc need an update?

Open an Issue on GitHub

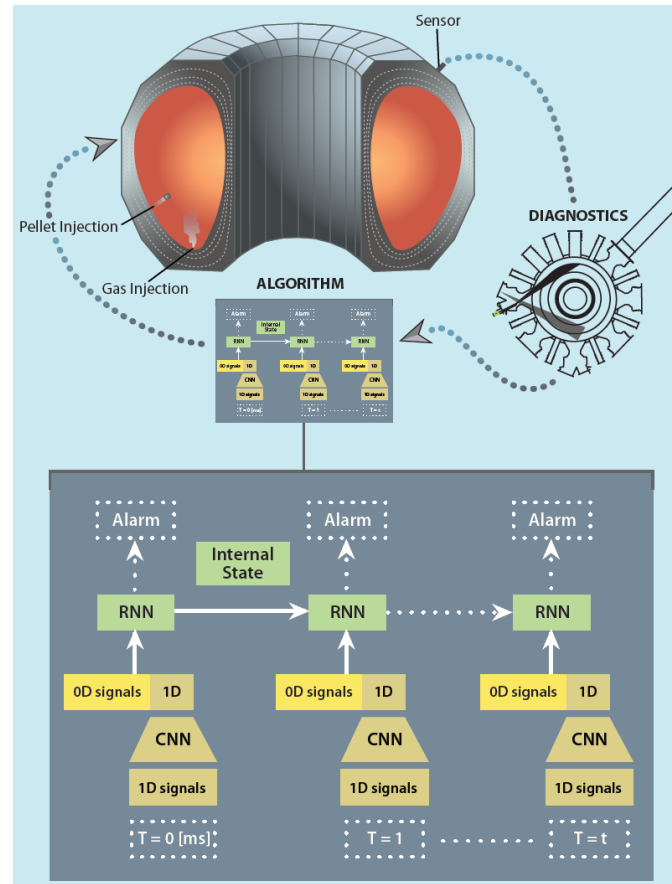
AI for Science and HPC applications on AI Testbed



Cancer drug response prediction
(Credit: Candle)



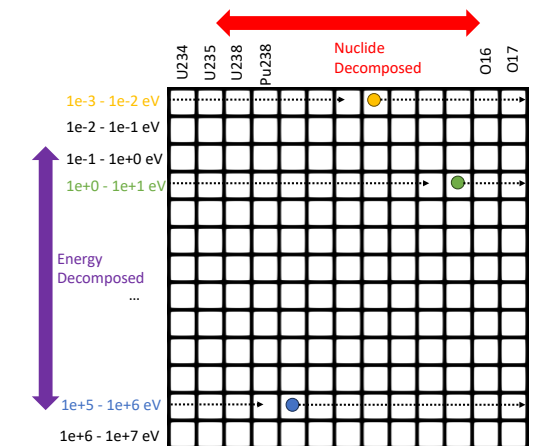
Imaging Sciences-Braggs Peak
(Credit: Z. Liu)



Tokamak Fusion Reactor operations
(Credit: K. Felker)



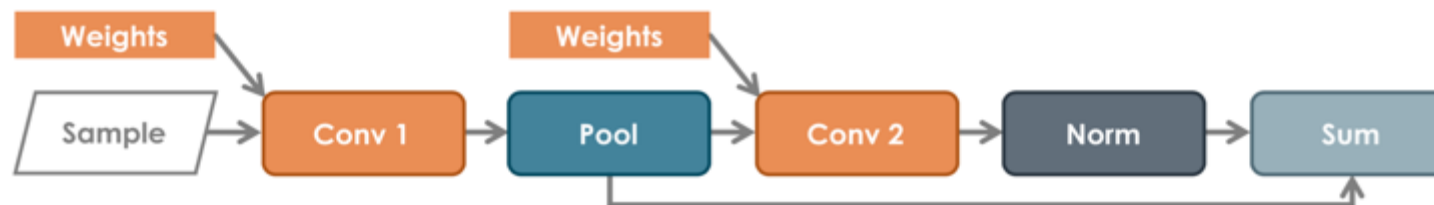
Protein-folding (Image: NCI)



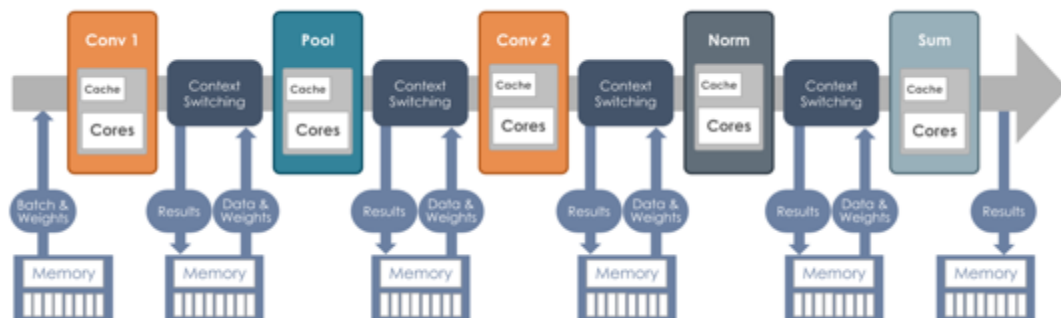
Monte Carlo Particle Transport for
Reactor Simulation (Credit: J. Tramm)

and more..
extremecomputingtraining.anl.gov

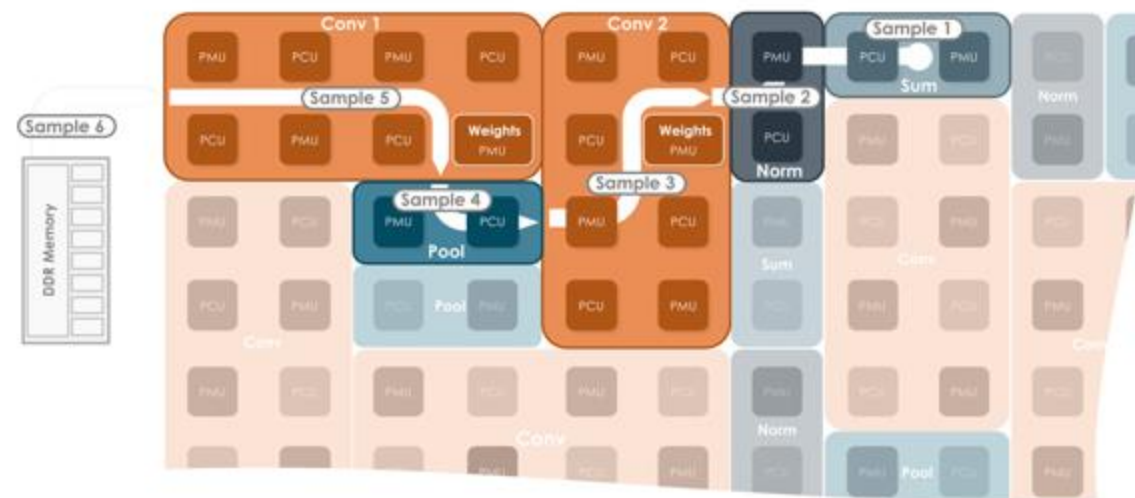
Dataflow architecture



Simple
Convolution
Graph



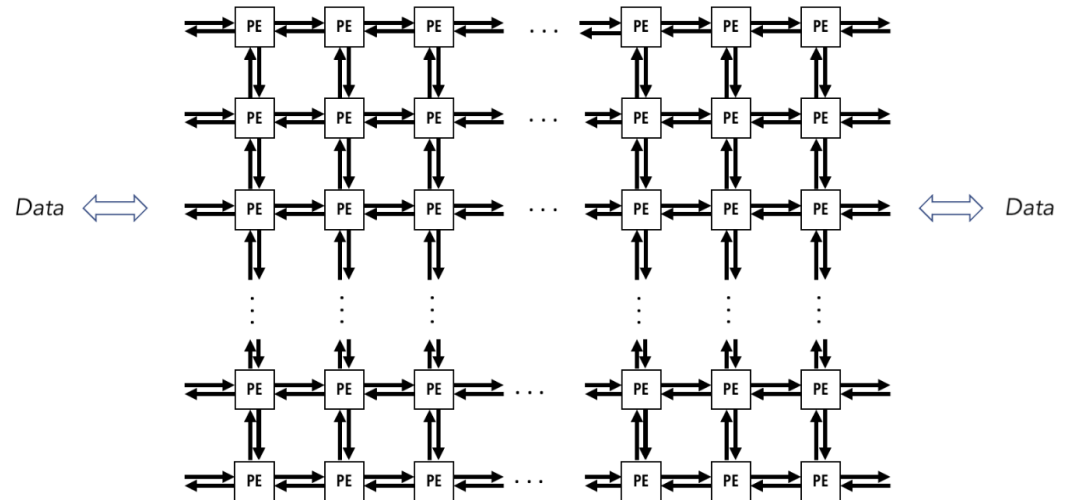
The GPU way: kernel-by-kernel
Bottlenecked by memory bandwidth
and host overhead



The Dataflow way: Spatial
Eliminates memory traffic and overhead

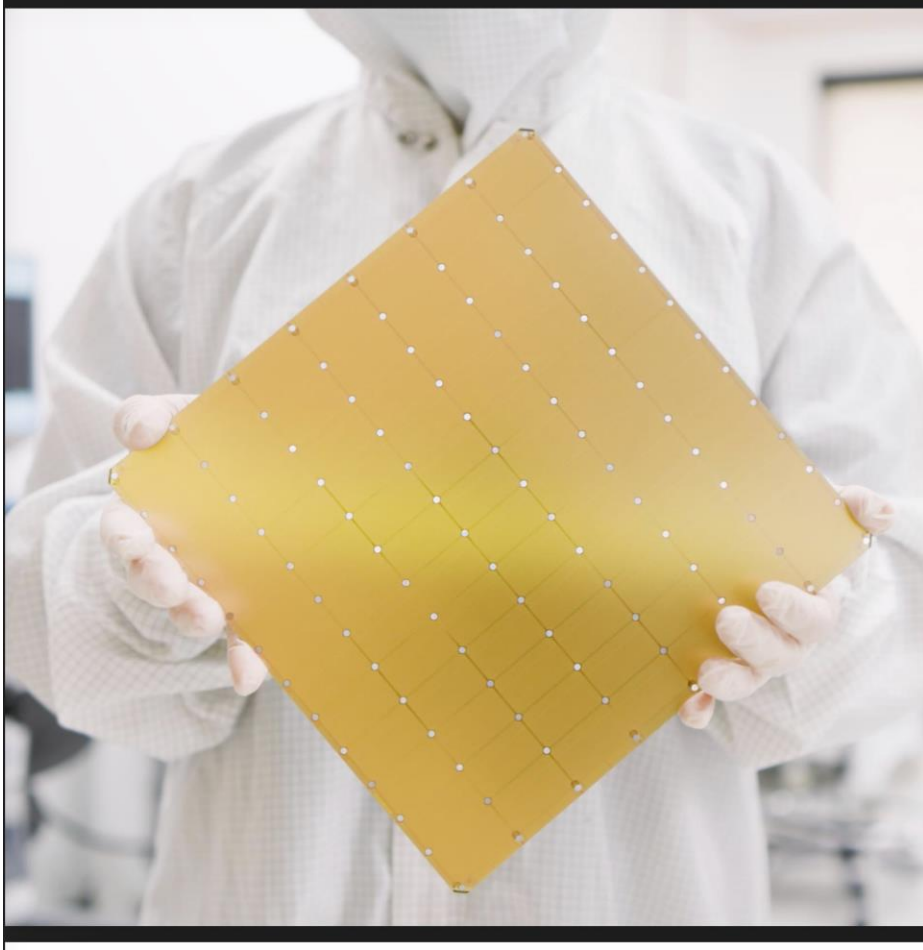
Image Courtesy: SambaNoya

Dataflow architecture



The diagram illustrates the architecture of a processing element (PE). On the left, a small circular inset shows a PE with four bidirectional connections to its neighbors. The main diagram shows a detailed view of a PE, enclosed in a dashed blue box. Inside the PE, there is a Fabric Router (a rounded rectangle) at the top. Below it is a shaded gray area containing an offramp (downward arrow) and an onramp (upward arrow). Below the offramp is a Processor (rectangle), and below the Processor is Memory (rectangle). Bidirectional arrows connect the Fabric Router to the offramp and onramp, and the Processor to the Memory. The PE is connected to a network of other PEs, indicated by bidirectional arrows on the left and right sides of the PE box.

- Image courtesy: SambaNova



Cerebras Wafer Scale Engine 3 (WSE-3)

The fastest AI chip on earth **again**

4 trillion transistors

46,225 mm² silicon

900,000 cores optimized for sparse linear algebra

5nm TSMC process

125 Petaflops of AI compute

44 Gigabytes of on-chip memory

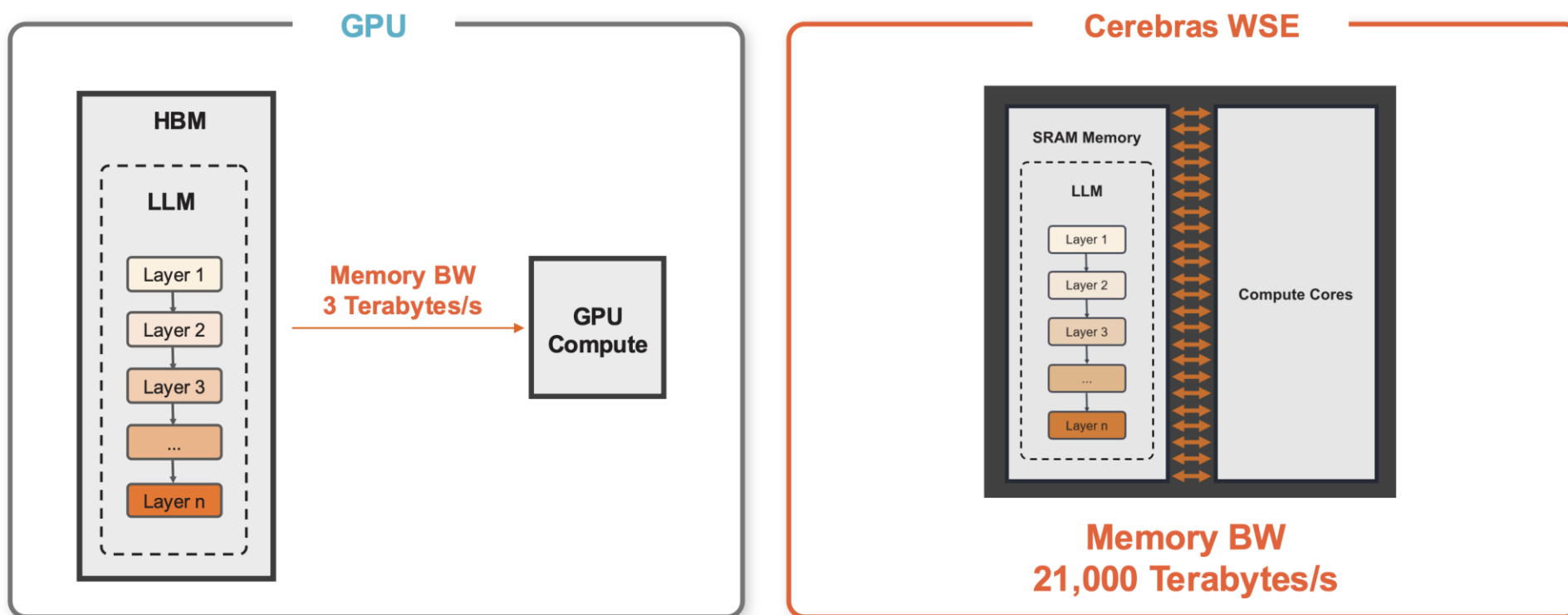
21 PByte/s memory bandwidth

214 Pbit/s fabric bandwidth

Courtesy: Cerebras

Cerebras

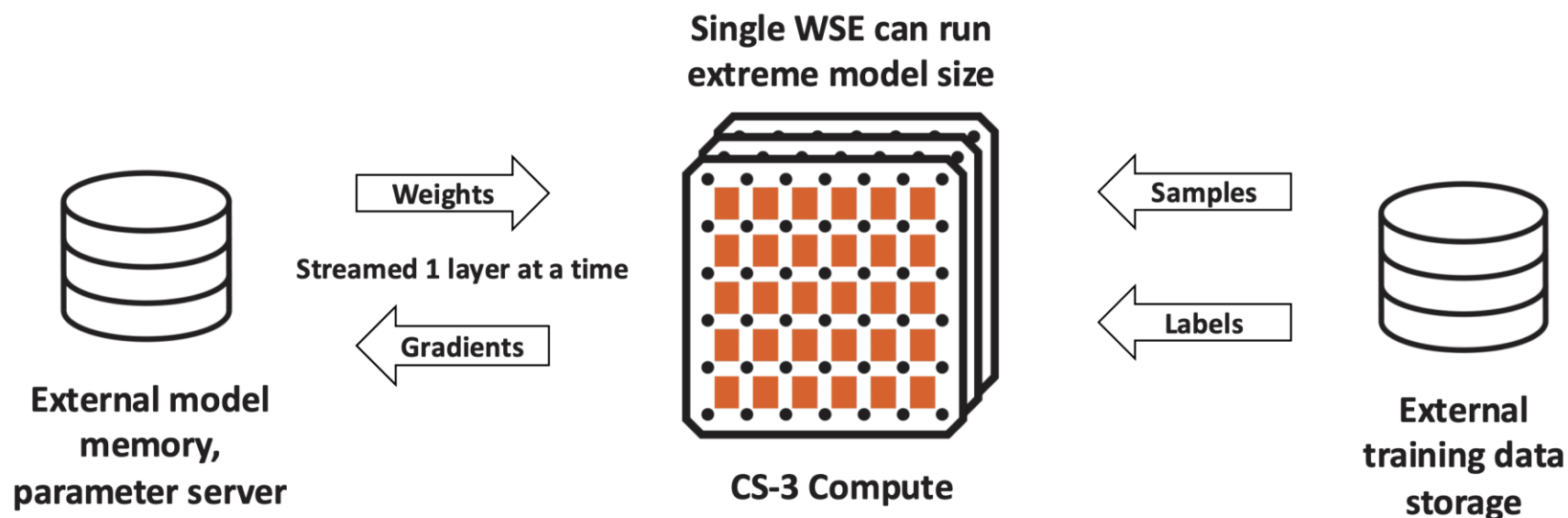
Cerebras WSE delivers 7,000x more memory bandwidth than Nvidia H100
Resulting in >70x faster inference speeds



The WSE puts massive amounts of high-performance SRAM directly on the wafer
Faster memory, closer to compute, delivers faster results

Weight Streaming Technology

Disaggregates Memory and Compute for Trillion Parameter Model Training

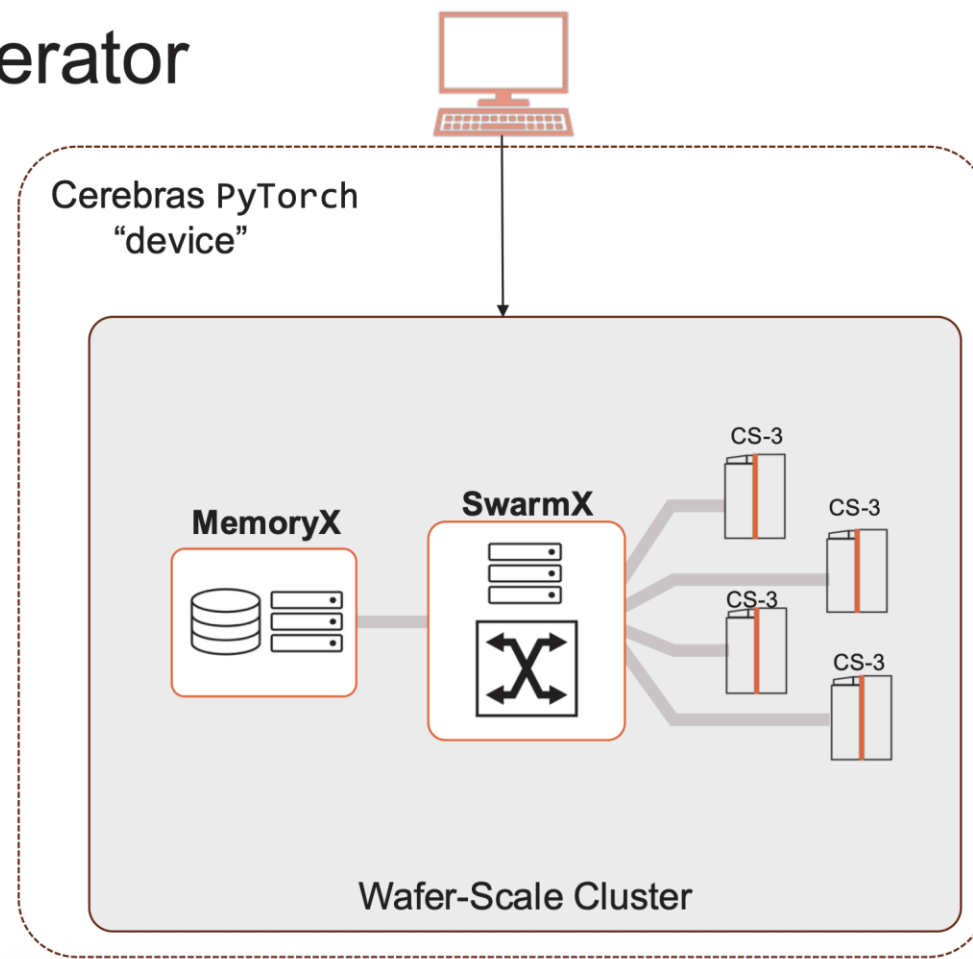


Scale model size and training speed independently

The Cluster *is* the ML Accelerator

Disaggregation of memory and compute

- Stores model weights off-wafer with fast streaming
- MemoryX size and compute device count can be scaled independently
- The entire cluster acts one remote accelerator
- All the complexity of distributed training, data parallelism, and scaling are handled automatically by the Wafer Scale Cluster
- Exposed to the user as a single PyTorch device!



The Cerebras Modelzoo

github.com/Cerebras/modelzoo

As a **starting** point

Reference model implementations:
GPT-3, LLaMa, Mistral, BERT, LLaVa, and much more!

Data preparation scripts

Configurations for multiple model scales

For **benchmarking**

Tuned implementations for optimal performance

To **develop** new models

Cerebras PyTorch & Layer APIs

File organization as a template to start your own models

Custom Training Loops with the cstorch API

- Scale out to multiple CS-3s with a one-argument configuration change
- Near-linear scaling happens automatically
- We don't need to change...
 - Model code
 - Training loop
 - Effective batch size
- It's easy to supply the value programmatically for edit-free scaling

Scale out {

```
1 import torch
2 import cerebras_pytorch as cstorch
3
4 model: torch.nn.Module = Model()
5 model = cstorch.compile(model, "CSX")
6 loss_fn = torch.nn.NLLLoss()
7 optimizer = cstorch.optim.SGD(model.parameters(), lr=0.01)
8 dataloader = cstorch.utils.data.DataLoader(get_train_dataloader)
9 executor = cstorch.utils.data.DataExecutor(
10     dataloader, cs_config=cstorch.utils.CSConfig(num_csx=16)
11 )
12
13 @cstorch.trace
14 def training_step(inputs, targets):
15     optimizer.zero_grad()
16     outputs = model(inputs)
17     loss = loss_fn(outputs, targets)
18     loss.backward()
19     optimizer.step()
20     return loss
21
22 @cstorch.step_closure
23 def print_loss(loss: torch.Tensor):
24     print(f"Loss: {loss.item()}")
25
```

Scaling from 8B model to 70B

Define the model

- Same code as for Llama-3 8B
- Different yaml parameterization
- 70B model code is still for a *single* device

params_llama3_70b.yaml

```
### LLaMA-3 70B  
  
hidden_size: 8192  
num_hidden_layers: 80  
num_heads: 64
```

Train the model

- Point to the upscaled model parameters
- The rest is the same as with Llama-3 8B
- Run!

training:

```
cszoo fit params_llama3_70b.yaml \  
--num_csx 1
```

Courtesy: Cerebras

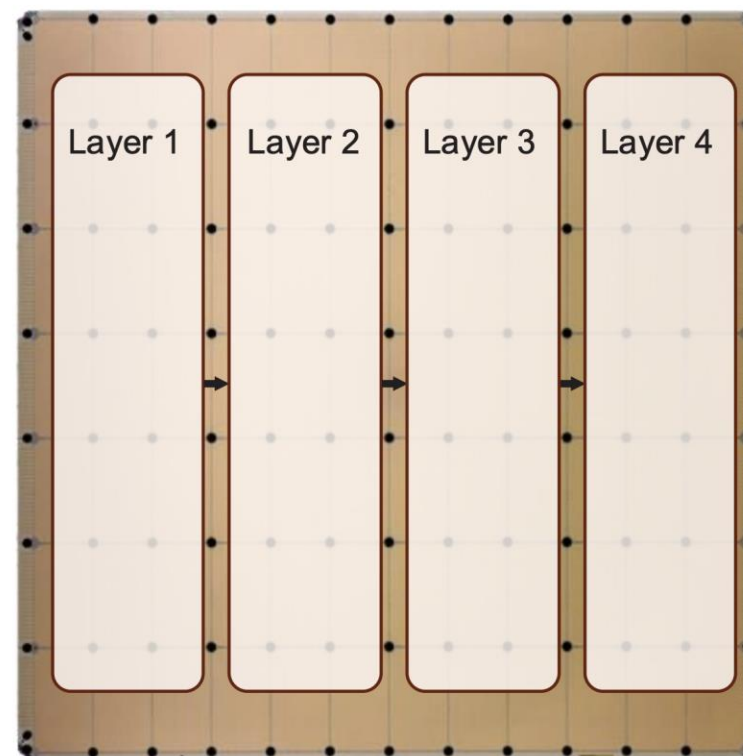
Pipeline execution on a single chip

Massive memory bandwidth enables opposite execution model

- GPU: multiple chips to run a single layer
- Cerebras: fraction of a chip to run a single layer

Pipeline mapping of model

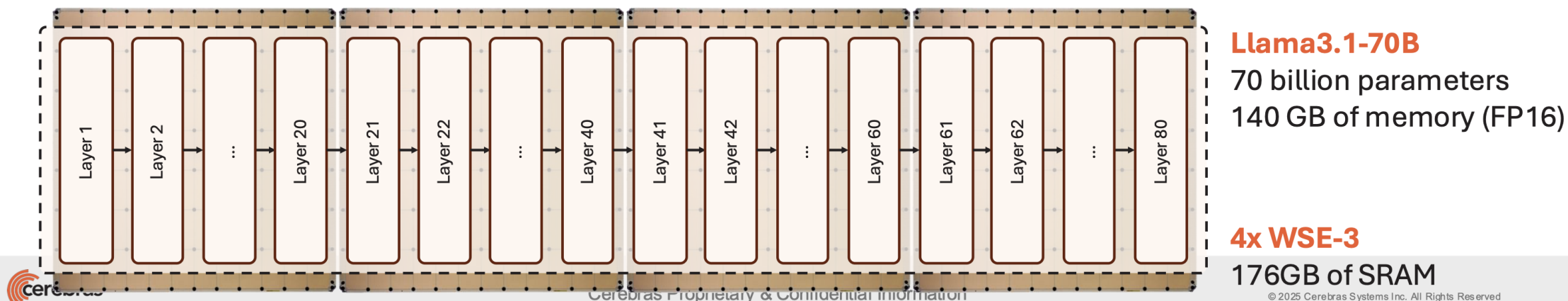
- Model layers are mapped to wafer regions
- Size of wafer region determined based on memory and compute requirements
- Model weights and KV cache stored locally in the region memory close to the compute



Courtesy: Cerebras

Cerebras

- Models that require more memory are mapped to multiple wafers
- Keep almost all high communication on wafer, using internal high bandwidth fabric
- Transfer only activations between wafers, requiring relatively lower bandwidth
- Using CS-3 low latency RDMA over Ethernet interconnect between systems



Cerebras CSL

<https://docs.alcf.anl.gov/ai-testbed/cerebras/csl/>

- Cerebras CSL (Cerebras System Language) is a low-level kernel programming language designed for the Cerebras system.
 - includes libraries for a variety of commonly used primitive operations, such as broadcasting, gathering, and scattering data across rows or columns of PEs.
- It enables users to write code that runs on individual Processing Elements (PEs) and to define the placement of programs and the routing of data on the Wafer-Scale Engine (WSE).
- 2 components: Device-code: runs directly on the wafer, host-code to facilitate data movement and execute functions on the wafer.
- Simulator mode: runs locally or Appliance mode: runs directly on CS-3.

Metis – SN40L Inference cluster

SambaNova SN40L



- 6x SN40L nodes each with 16x SN40L Accelerators
- 1.5TB per RDU – 48TB in aggregate
- Highly optimized for inference

Metis – SN40L

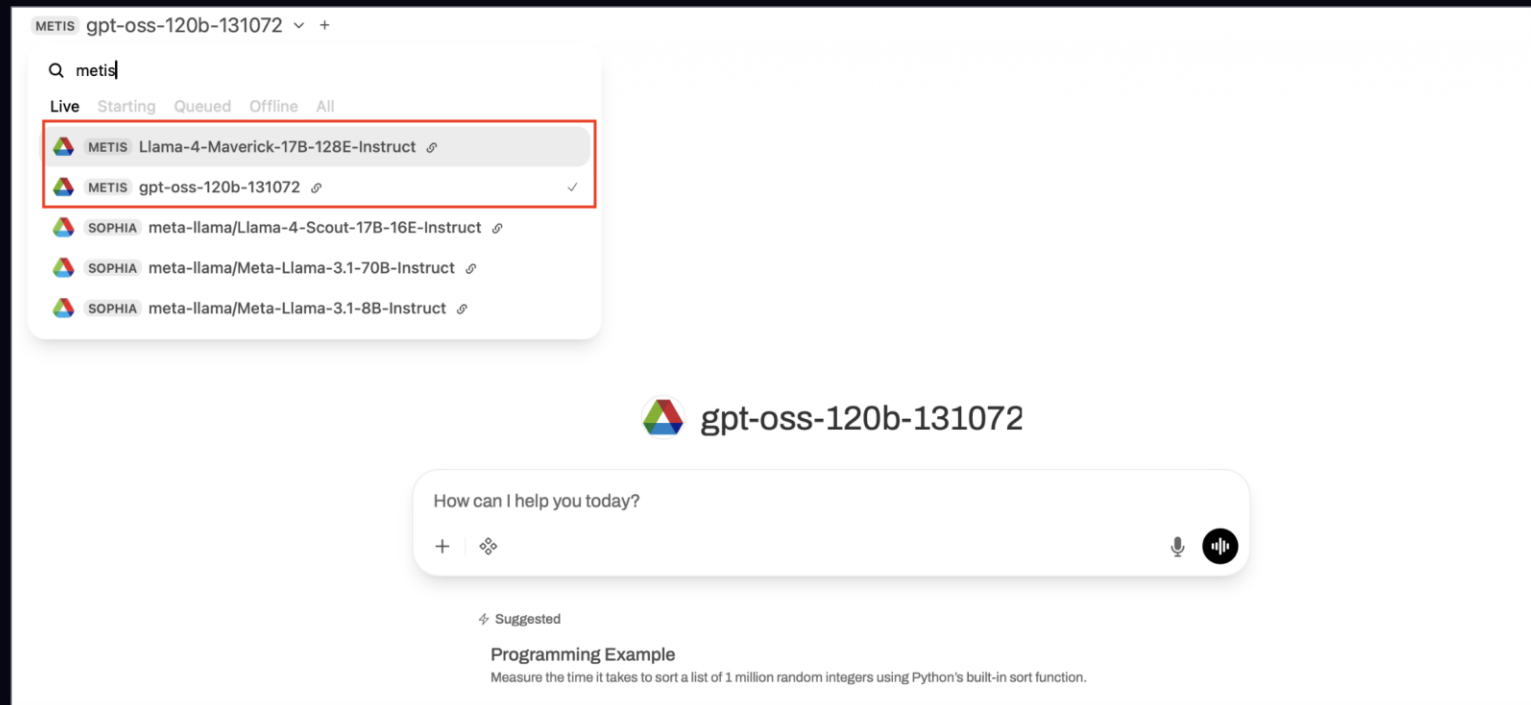
Accessing the endpoints using the Web UI.

The easiest way to get started is through the web interface, accessible at <https://inference.alcf.anl.gov/>

The UI is based on the popular Open WebUI platform. After logging in with your ANL or ALCF credentials, you can:

1. Select a model from the dropdown menu at the top of the screen.
2. Start a conversation directly in the chat interface.

In the model selection dropdown, you can see the status of each model:



Sambanova - SN40L

Choose sidebar display

 sambanova

SN40L: SambaNova's 4th Gen AI Chip

Reconfigurable Dataflow Unit (RDU)
Native multi-tenancy support with fast model switching
Ideal for production inference, multi-tenancy, agentic workflows



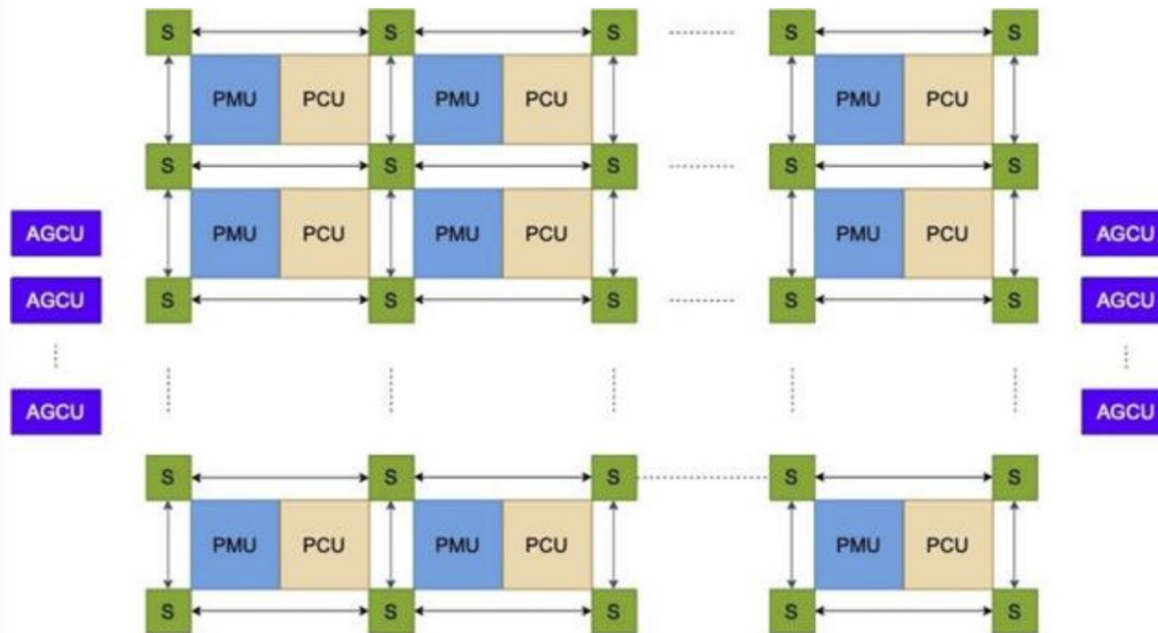
sambanova
SN40L RDU

- 3-tier Dataflow Memory
 - 520 MB On-Chip SRAM Memory → Very fast memory for high speed inference with caching
 - 64 GB High Bandwidth Memory → Switch between models in as little as 2 milliseconds
 - 768 GB High Capacity DDR Memory → Hold large number of models in memory

Copyright © 2025 SambaNova Inc. | sambanova.ai



SN40L: Tile Architecture



1040 PCUs and PMUs

PCU: Compute unit

PMU: Memory unit

S: Mesh switches

AGCU: Portal to off-chip
memory and IO

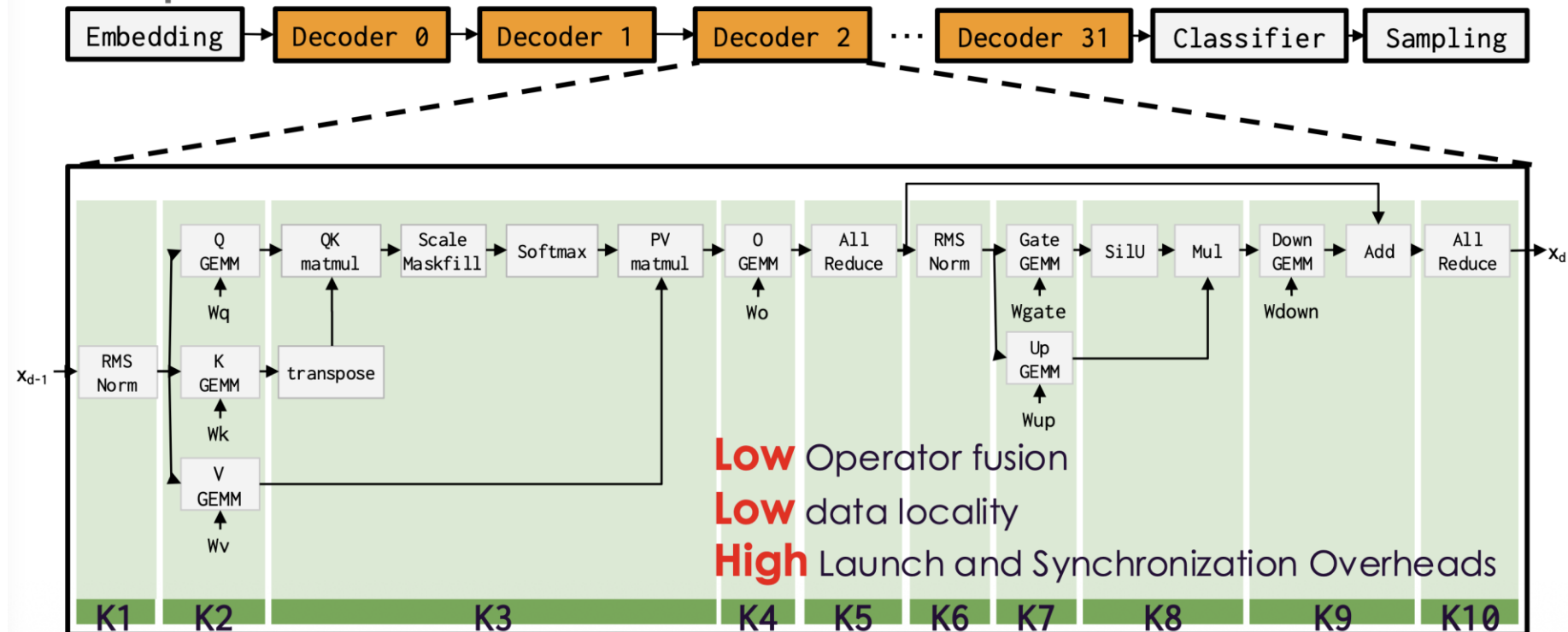
Copyright © 2025 SambaNova Inc. | sambanova.ai

Sambanova - SN40L



GPUs Incur Overhead with Low Operator Fusion

Example: Llama3.1 8B



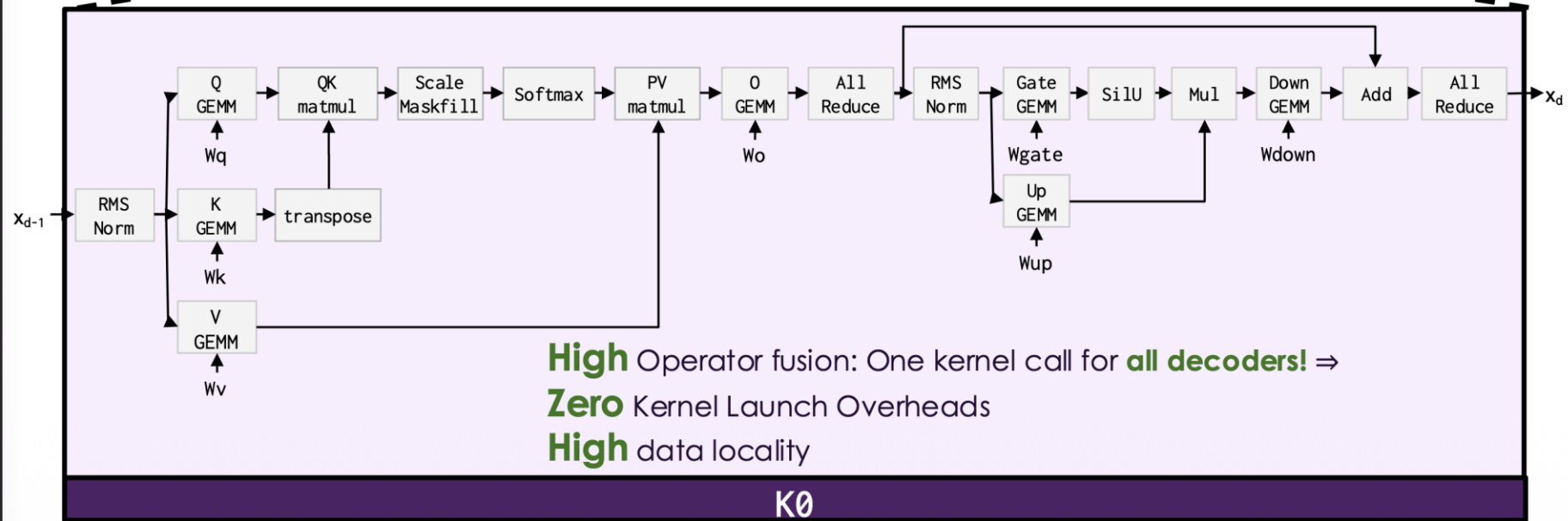
Copyright © 2025 SambaNova Inc. | sambanova.ai

Sambanova - SN40L



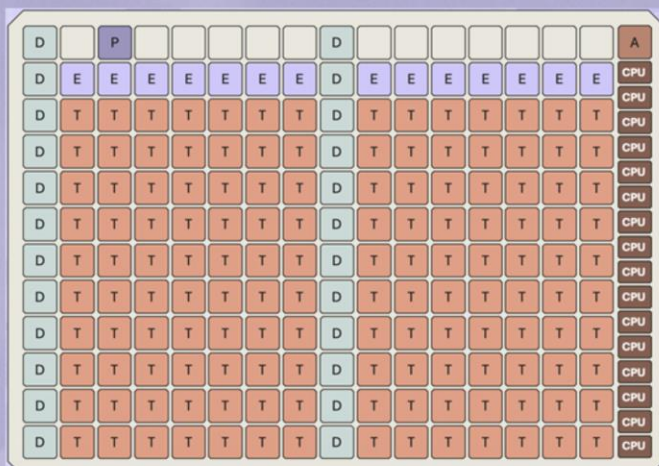
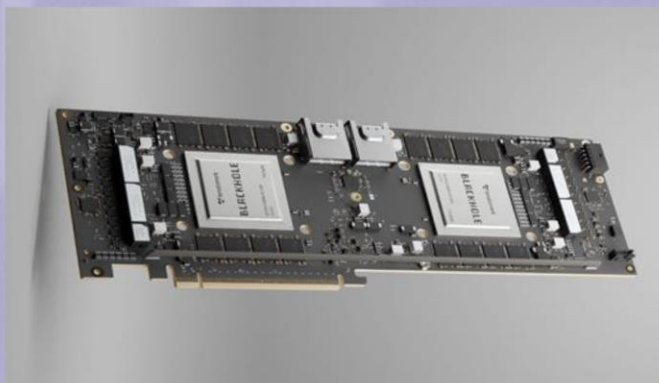
SN40L RDU Fuses the Entire Decoder into One Kernel!

Example: Llama3.1 8B



Copyright © 2025 SambaNova Inc. | sambanova.ai

Tenstorrent



- CPU** RiscV CPUs
- T** Tensor Cores
- D** DRAM Cores
- E** ETH Cores
- P** PCIe Core
- A** ARC Core

Blackhole™

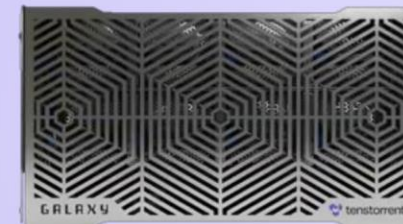
- 120 Tensix AI Processors
- 664 TFLOPs (8-bit)
- 32GB GDDR6 @ 512 GB/s
- 180MB SRAM
- 16 Risc-V Processors
- Dual 512-bit 2D Torus NOC
- PCIe Gen5 x 16
- Up to 10 x 400 Gbps Interconnect for Scale-out

12



Blackhole Galaxy

- 32 Blackhole chips
- GDDR6: 1 TB @ 16 TB/s
- 56 x 800 Gbps Ethernet
- Unified scale-up & scale-out
- Virtualize multiple Galaxies into Exaflop-scale systems

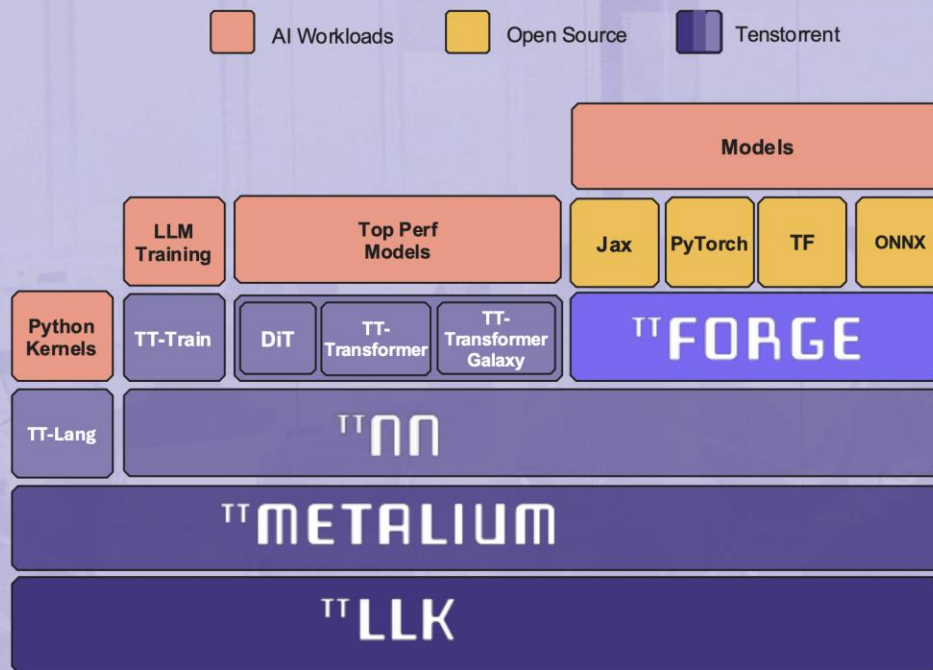


13

Courtesy: Tenstorrent

Building Block Approach

- Multiple entry points, from pytorch down to low-level kernel
- Fast growing ecosystem of partners & developers



Third-Party Developers



Model Development



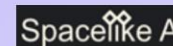
AI Kernel Generation



DX & Model Bring-up



Model Development



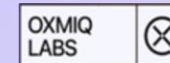
Model Development



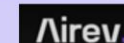
Triton compiler



In-Game AI



Cuda PyTorch Cross Compiler



AI Agents



Datacenter AI Visualization



AI Training



ML compiler



ML framework

16

Courtesy: Tenstorrent

ARGONNE ATPESC2026 EXTREME - SCALE COMPUTING

ARGONNE TRAINING PROGRAM ON EXTREME-SCALE COMPUTING

Produced by Argonne National Laboratory, a U.S. Department of Energy Laboratory managed by UChicagoArgonne, LLC under contract DE-AC02-06CH11357.

Special thanks to the National Energy Research Scientific Computing Center (NERSC) and Oak Ridge Leadership Computing Facility (OLCF) for the use of their resources during the training event.

The U.S. Government retains for itself and others acting on its behalf a nonexclusive, royalty-free license in this video, with the rights to reproduce, to prepare derivative works, and to display publicly.