SambaNova SN10 RDU: Accelerating Software 2.0 with Dataflow

Raghu Prabhakar, Sumti Jairath

8/24/2021

SambaNova SYSTEMS

Safe Harbor Statement

The following is intended to outline our general product direction at this time. There is no obligation to update this presentation and the Company's products and direction are always subject to change. This presentation is intended for information purposes only and may not be relied upon for any purchasing, partnership, or other decisions.

SambaNova SYSTEMS

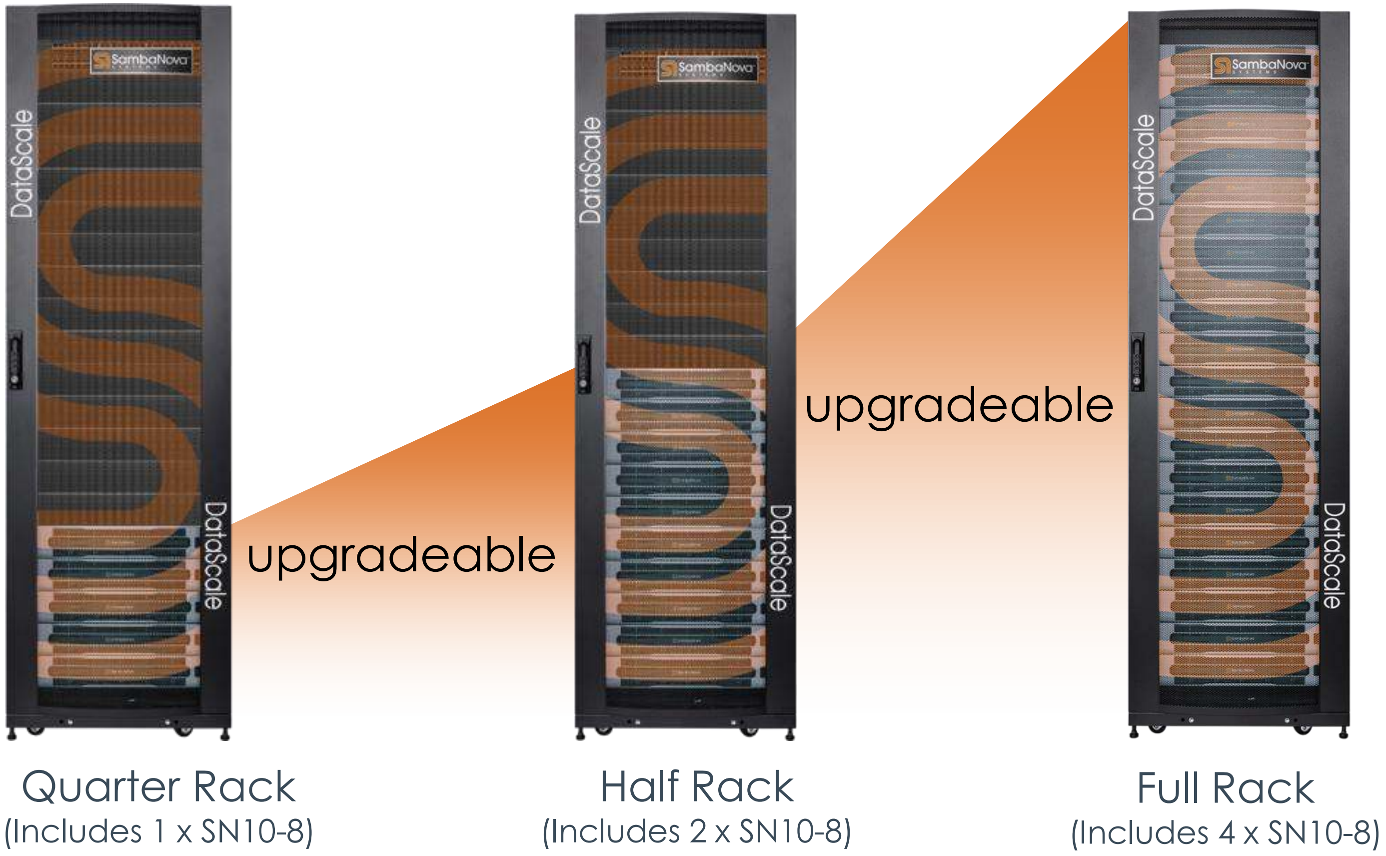
SambaNova Systems® Cardinal SN10 RDU

- First Reconfigurable Dataflow Unit (RDU)
- TSMC 7nm
  - Taped Out first half of 2019
  - 40B transistors, 50 Km of wire
- 640 Pattern Compute Units
  - >300 BF16 TFLOPs
  - BF16 with FP32 accumulation
  - Also supports FP32, Int32, Int16, Int8 data formats
- 640 Pattern Memory Units
  - >300 MB on-chip memory
  - 150 TB/s on-chip memory bandwidth
  - Memory transformation operations

SambaNova DataScale® SN10-8R Systems

Scalable performance for training and inference

- DataScale SN10-8R
  - 8x RDU in Quarter Rack
- 12 TB Memory
  - 48 DDR4-2667 Channels
- Host and RDU-RDU Communication
  - 32 PCIE-Gen4 x16 Links



Up to 38x more memory than conventional systems

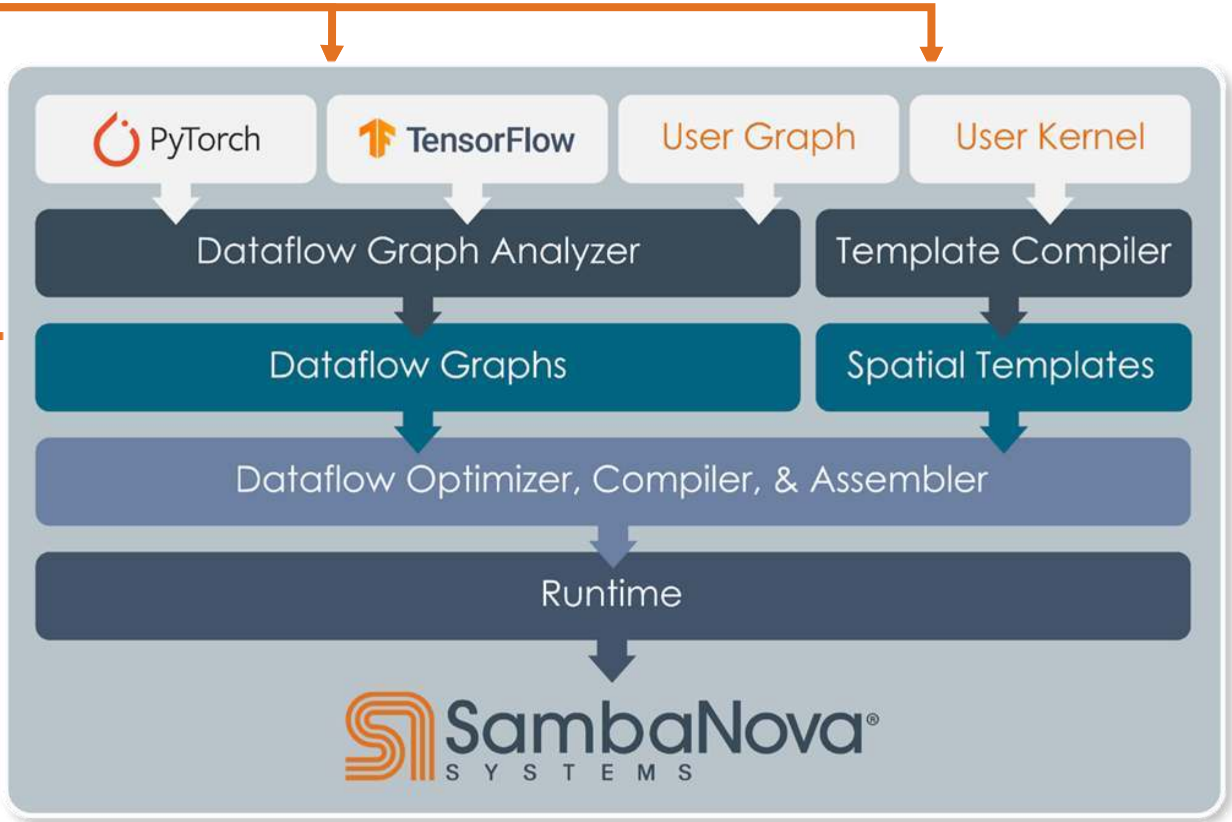
SambaFlow™ Software

Graph Entry Points

- Write to OSS ML frameworks or user's graph
- Push-button automation path

API Entry Point

- User programs to DSL
- Mix of manual and automatic



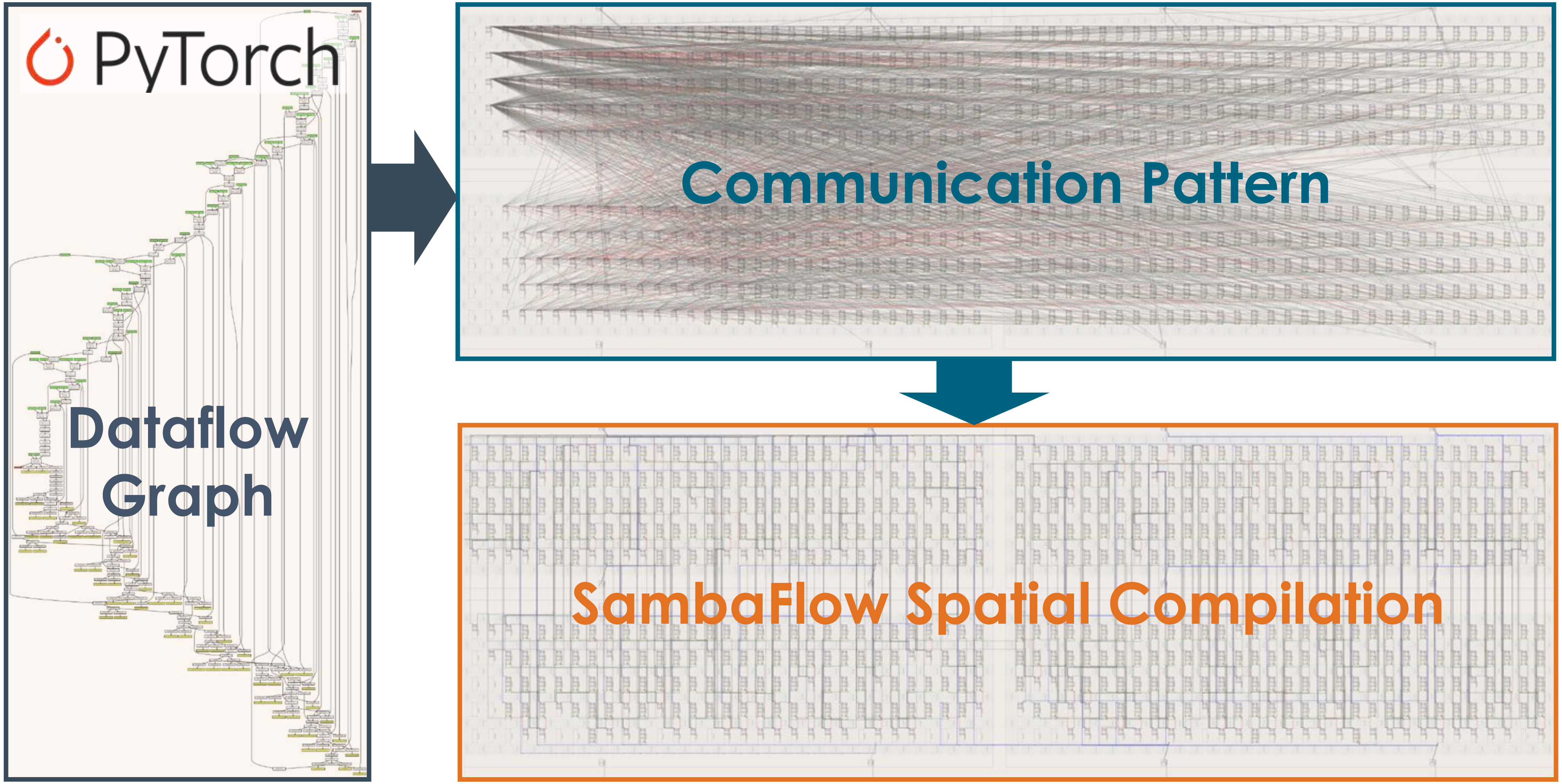
ML Optimizations

Model parallel
Data parallel

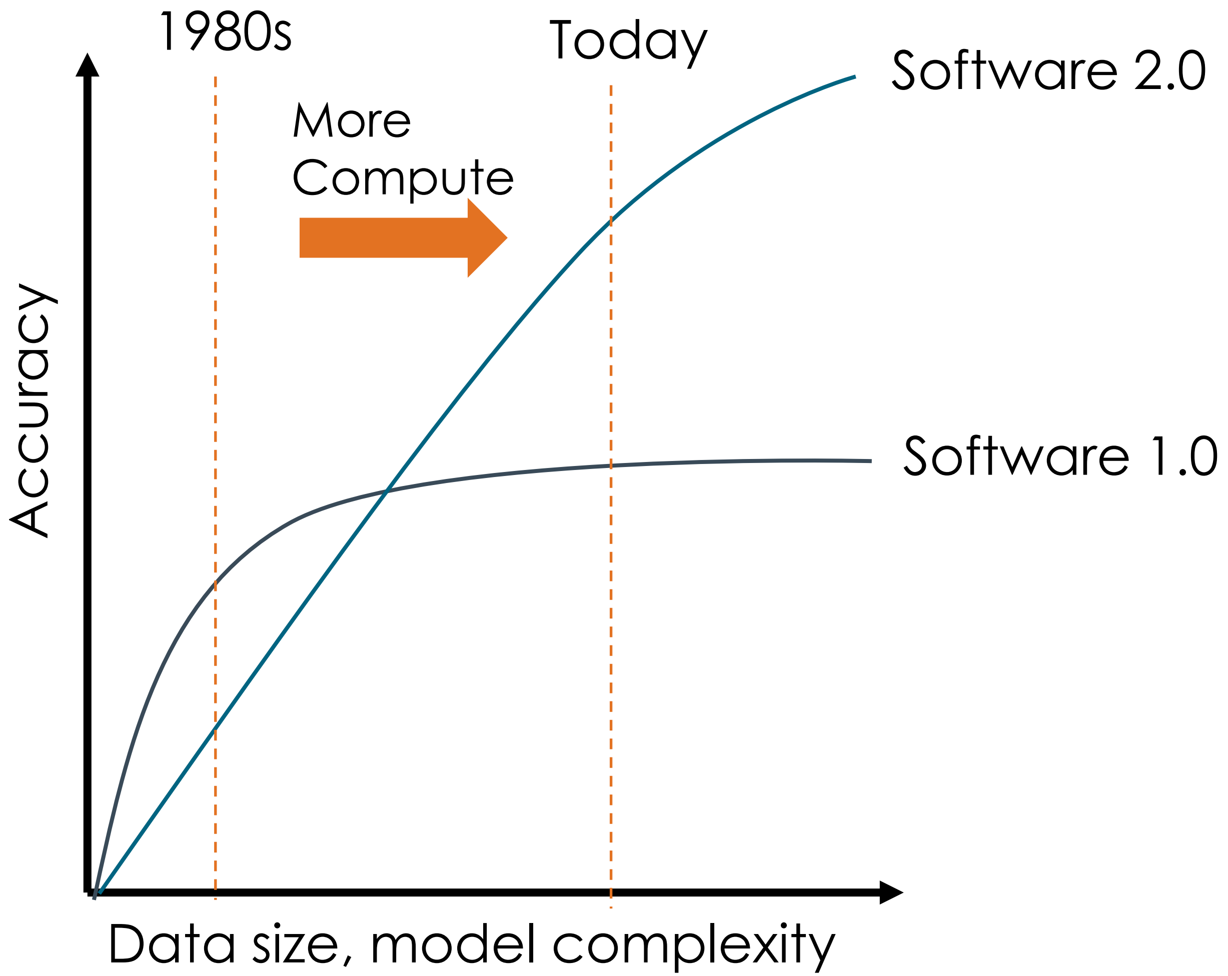
Dataflow Optimizations

Tiling
Op parallel
Streaming
Nested pipelining

SambaFlow Produces Highly Optimized Dataflow Mappings



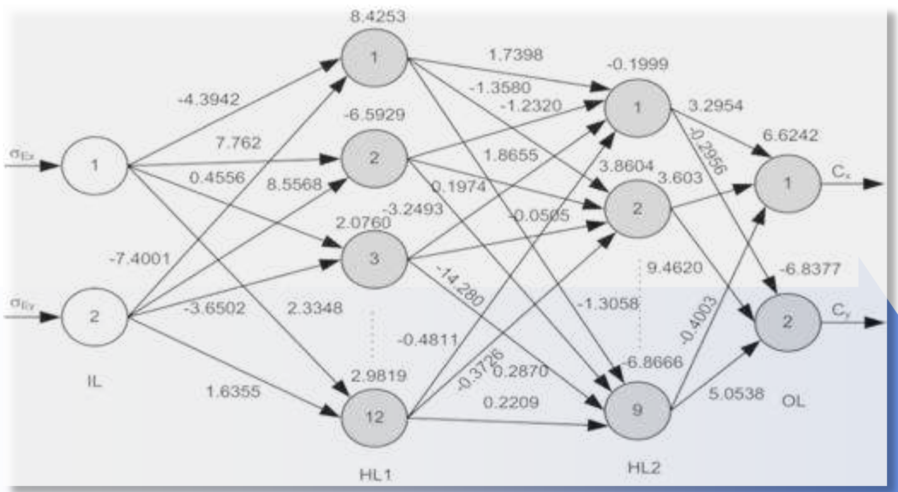
Machine Learning Today: Software 2.0



Adapted from Jeff Dean
HotChips 2017

Evolving Nature of Computational Models

```
template<typename InputIterator1, typename InputIterator2, typename NumericT>
NumericT  inner_product(InputIterator1 start1,
                        InputIterator1 end1,
                        InputIterator2 start2,
                        NumericT       startval) const {
  for (; start1 != end1; ++start1, ++start2)
    startval += ( (*start1) * (*start2) );
  return startval;
}
// ..
vector_type vec1, vec2;
vector_type::numeric_t val;
val = inner_product(vec1.begin(), vec1.end(), vec2.begin(),
                    vector_type::numeric_t(0));
```



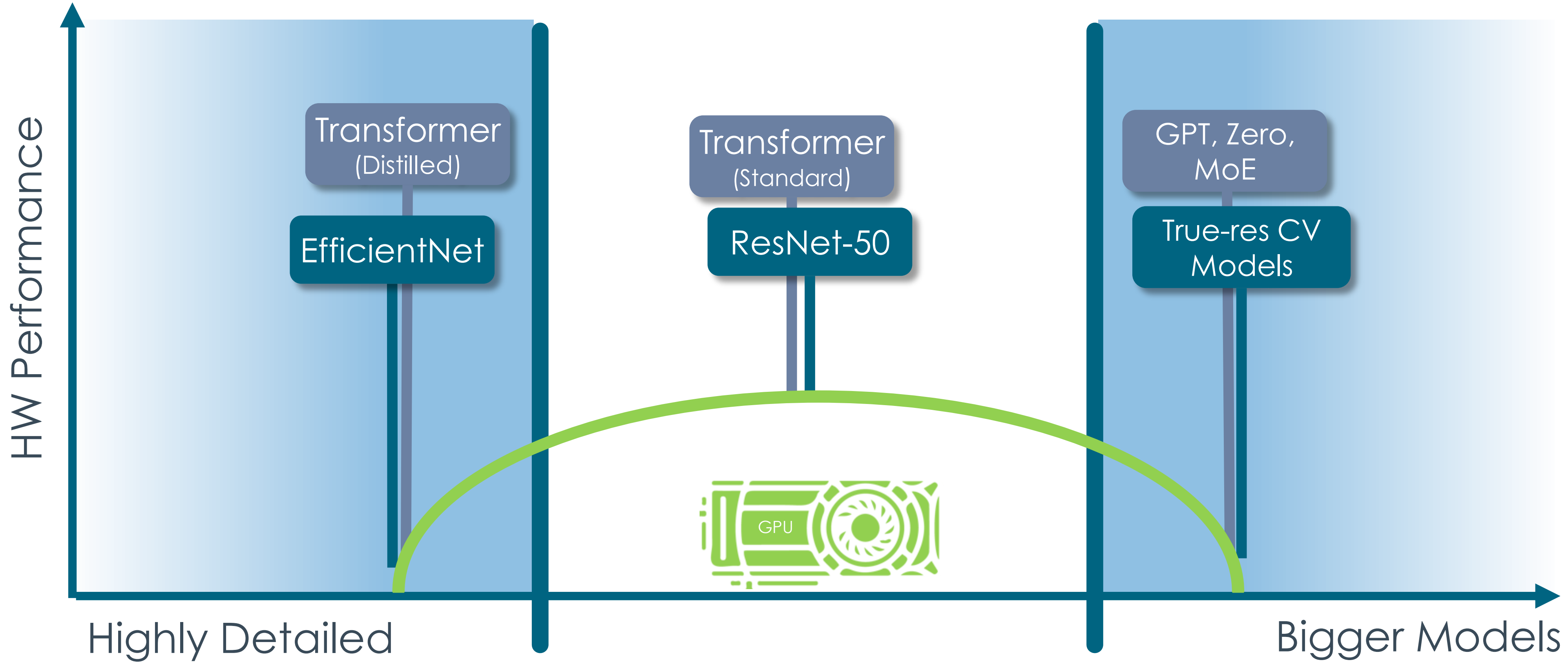
Software 1.0

- Programmer input: code (C++, etc.)
- Solution encoded in composed algorithms
- Deterministic computations
- Only has a single correct result

Software 2.0

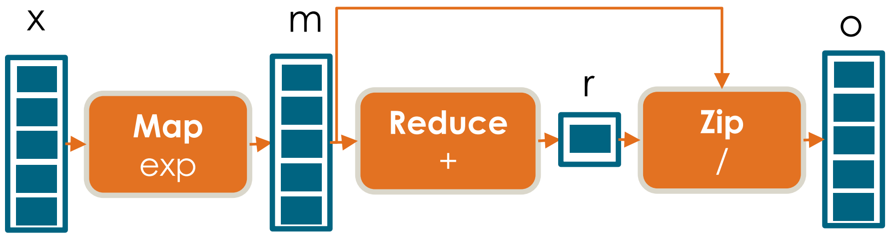
- Programmer input: training data
- Solution encoded in trained weights
- Probabilistic models
- Results need only be statistically correct

Yesterday's Goldilocks Zone is Constraining Progress

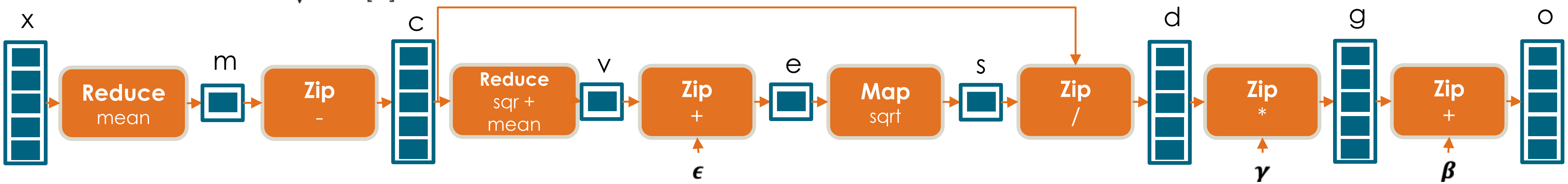


Dataflow Exploits Data Locality and Parallelism

SOFTMAX: $\mathrm{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$

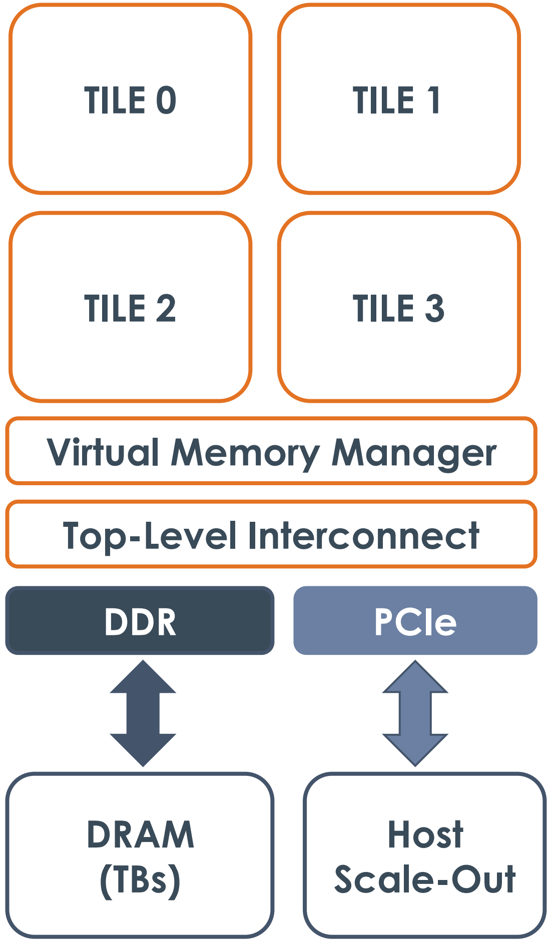


LAYERNORM: $y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$



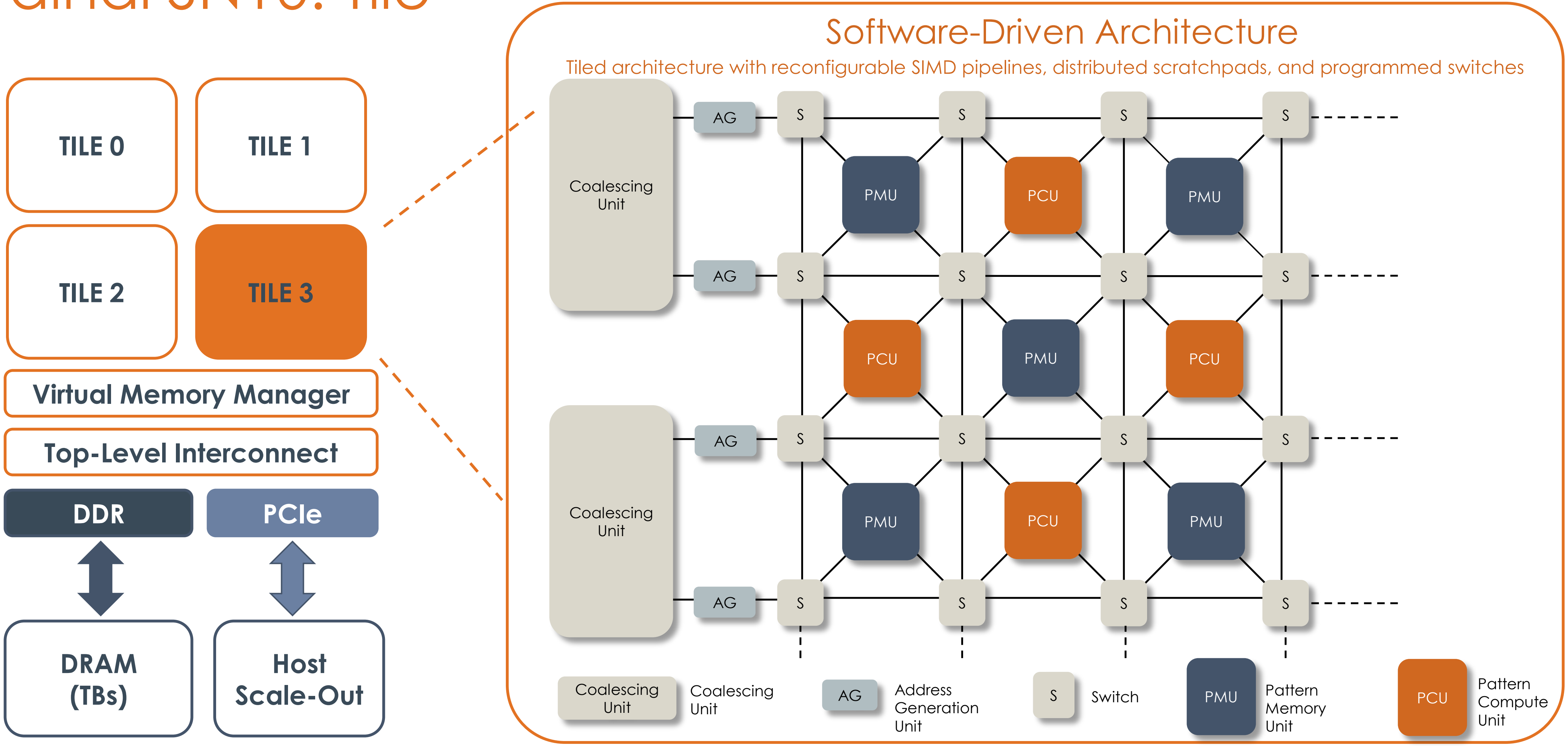
- Dataflow captures data locality and parallelism abundant in Software 2.0
- Flexible scheduling in space and time for higher utilization
- Flexible memory system and interconnect to sustain high compute throughput
- Build custom dataflow pipeline

Cardinal SN10: Chip and Architecture Overview

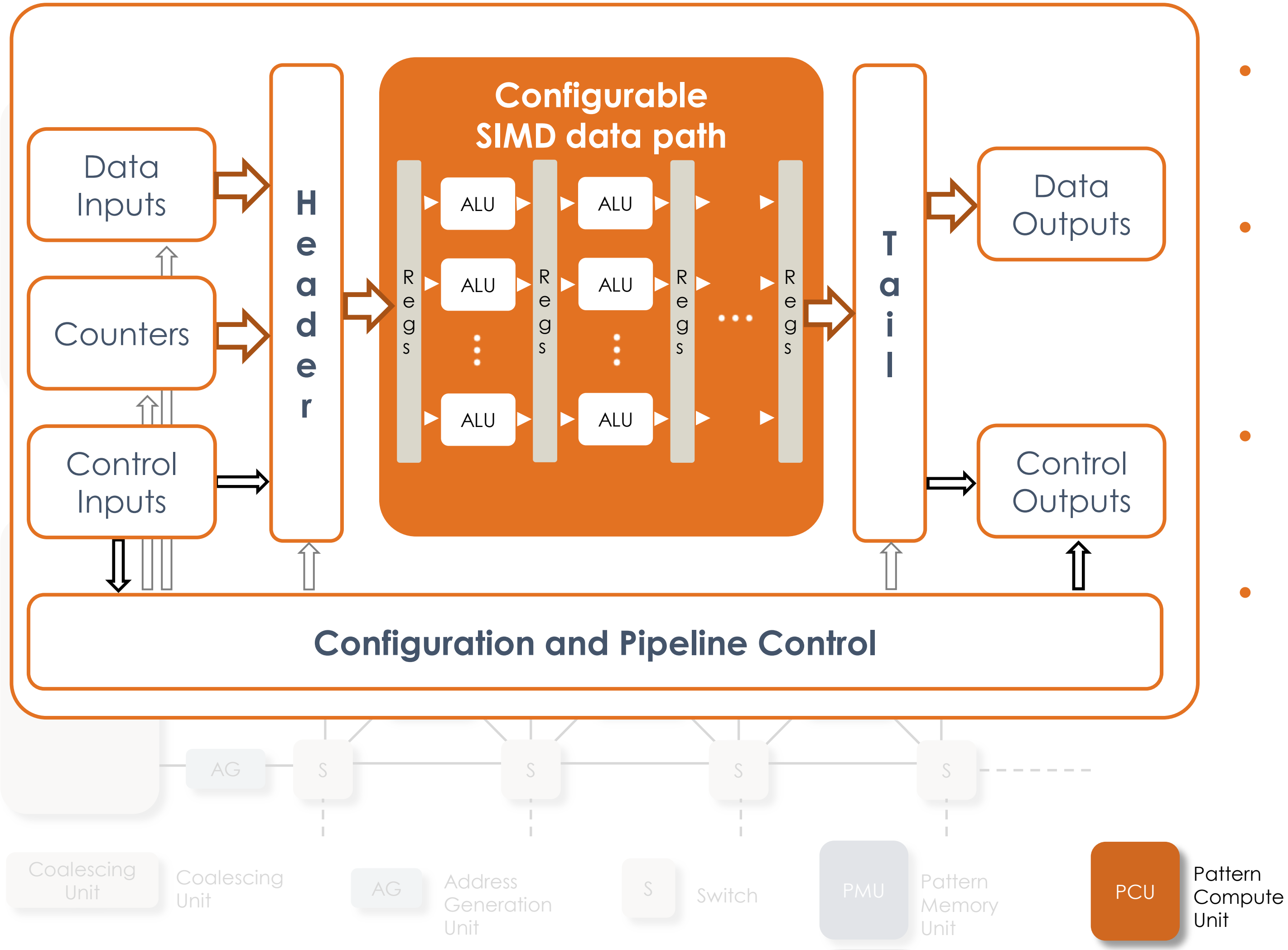


- TILE: Sea of programmable compute and memory components in a programmable interconnect
- Tile resource management: Combined or independent mode
  - Combined: Combine adjacent to form a larger logical tile for one application
  - Independent: Each tile controlled independently, allows running different applications on separate tiles concurrently.
- Direct access to TBs of DDR4 off-chip memory
- Memory-mapped access to host memory
- Scale-out communication support

Cardinal SN10: Tile

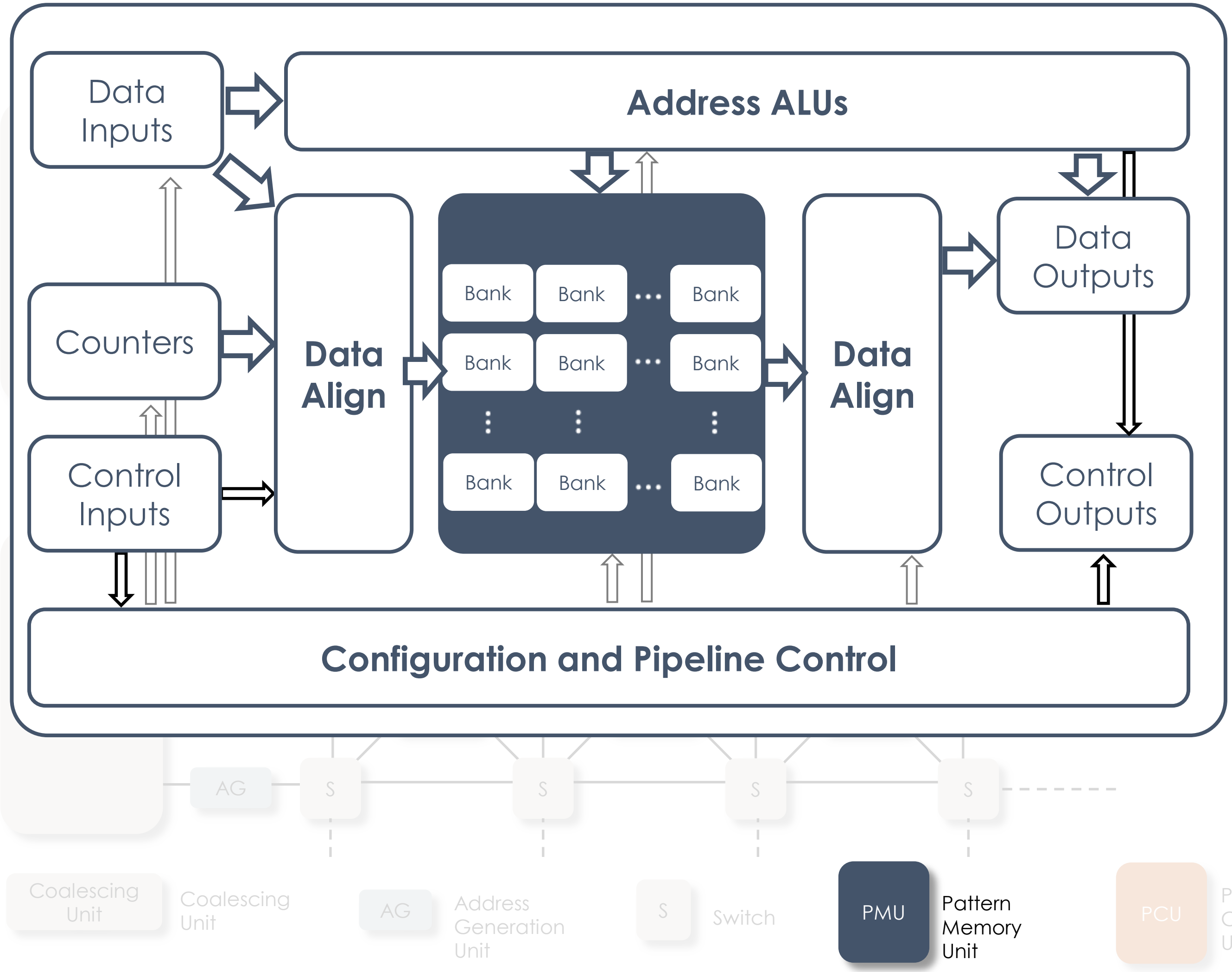


Cardinal SN10: PCU



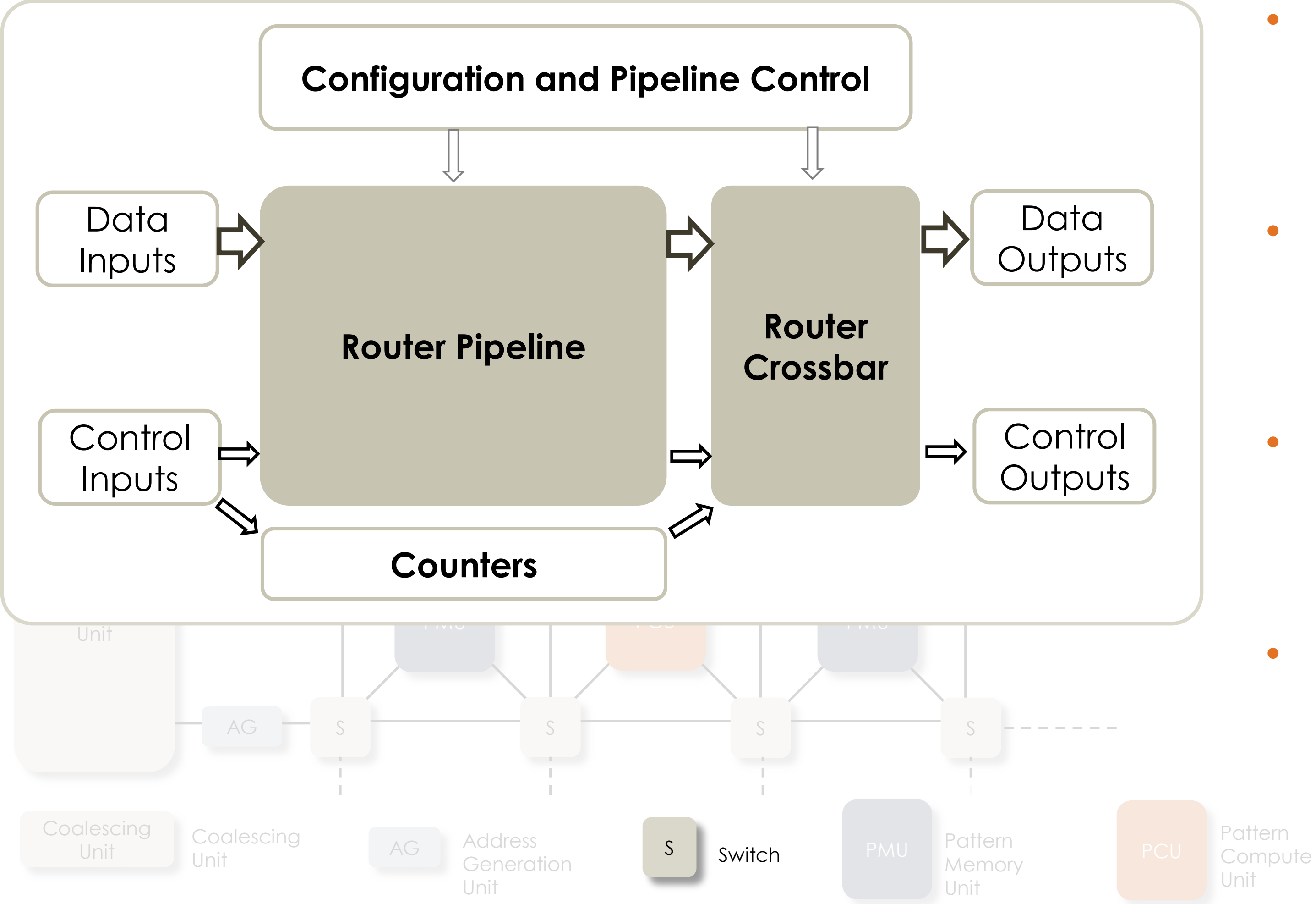
- **Pattern Compute Unit:** SN10's compute engines
- **Reconfigurable SIMD data path** for efficient dense and sparse tensor algebra in FP32, BF16, and integer formats
- **Programmable Counters** to efficiently program loop iterators
- **Tail unit** accelerates common functions like exp, sigmoid

Cardinal SN10: PMU



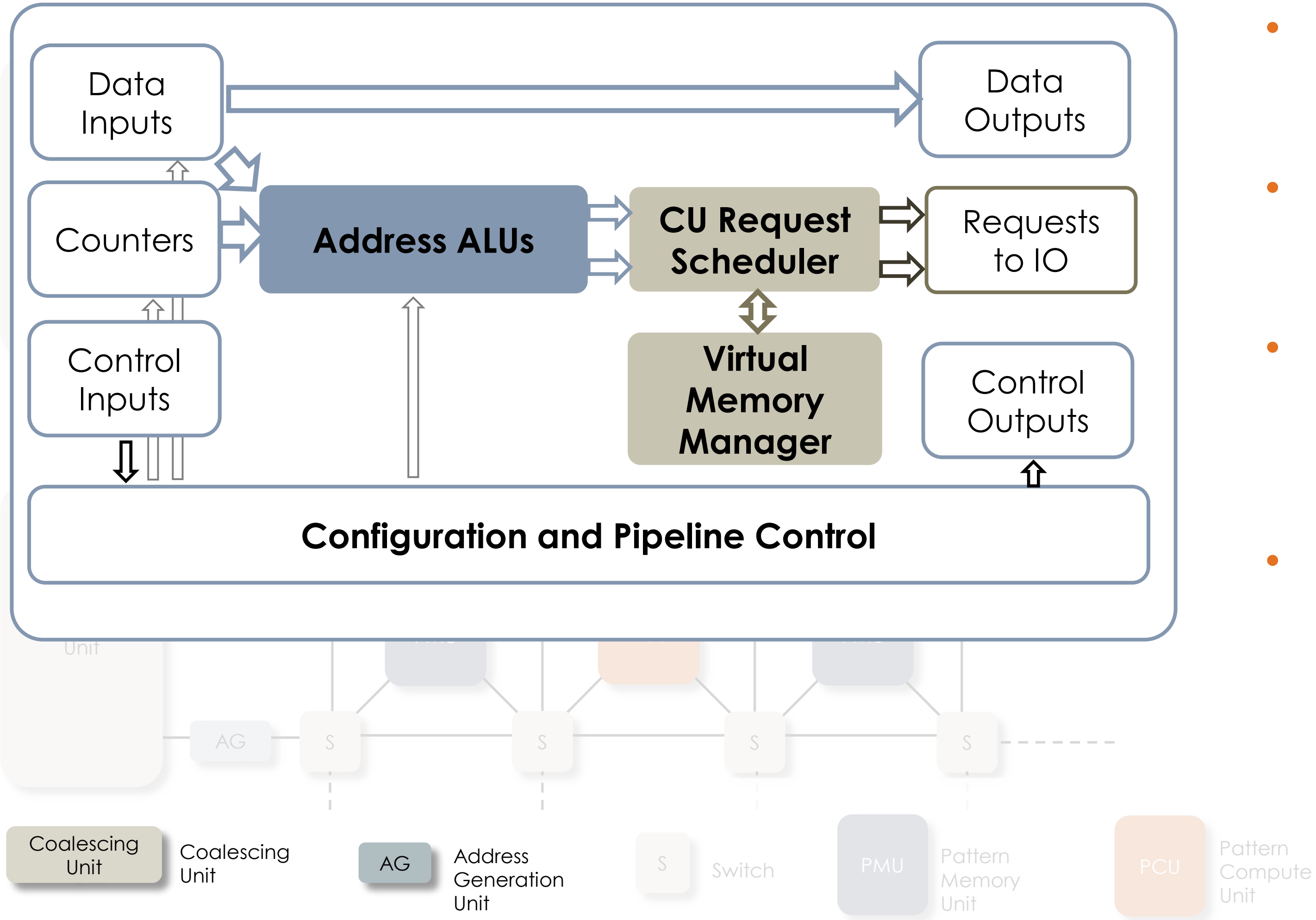
- **Pattern Memory Unit:** SN10's on-chip memory system
- **Banked SRAM arrays** to write and read multiple high-bandwidth SIMD data streams concurrently
- **Address ALUs** for high throughput address calculation for arbitrarily complex accesses
- **Data Align** units for high throughput Tensor layout transformations like transpose

Cardinal SN10: Switch and On-chip Interconnect



- **Switch:** SN10's programmable packet-switched interconnect fabric
- **Independent Data and Control** buses to suit different traffic classes inherent in real applications
- **Programmable Routing** for flexible on-chip bandwidth allocation to concurrent streams
- **Programmable Counters** to implement outer loop iterators, on-chip metric collection

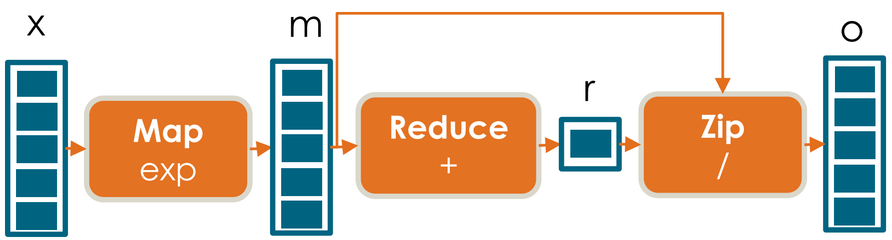
Cardinal SN10: AG & CU



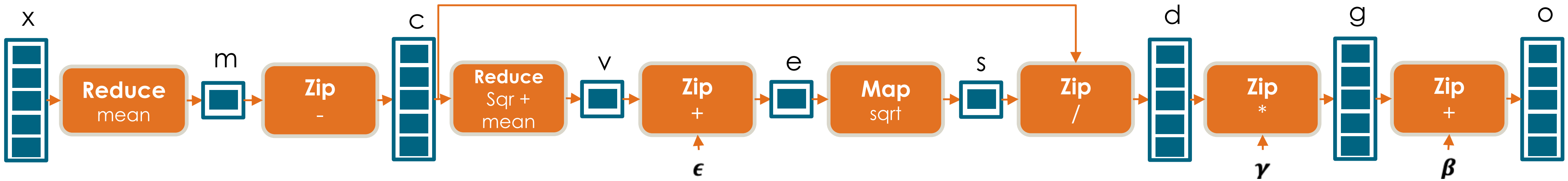
- **Address Generation and Coalescing Units:** SN10's interface to IO subsystem
- **Address ALUs** for high throughput address calculation for arbitrarily complex accesses
- **Coalescing Units** to enable transparent access to memories across RDUs and host memory
- Address space management using programmable, variable length **segments**

Programming Model

SOFTMAX: $$\mathrm{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

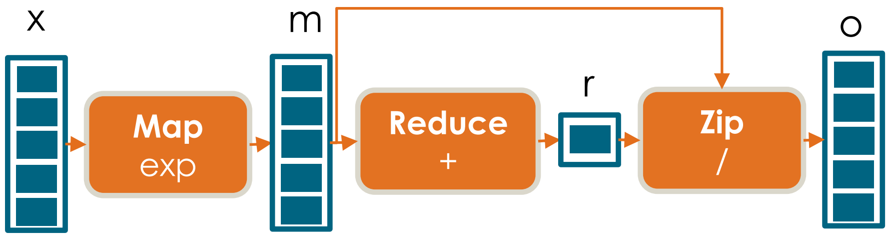


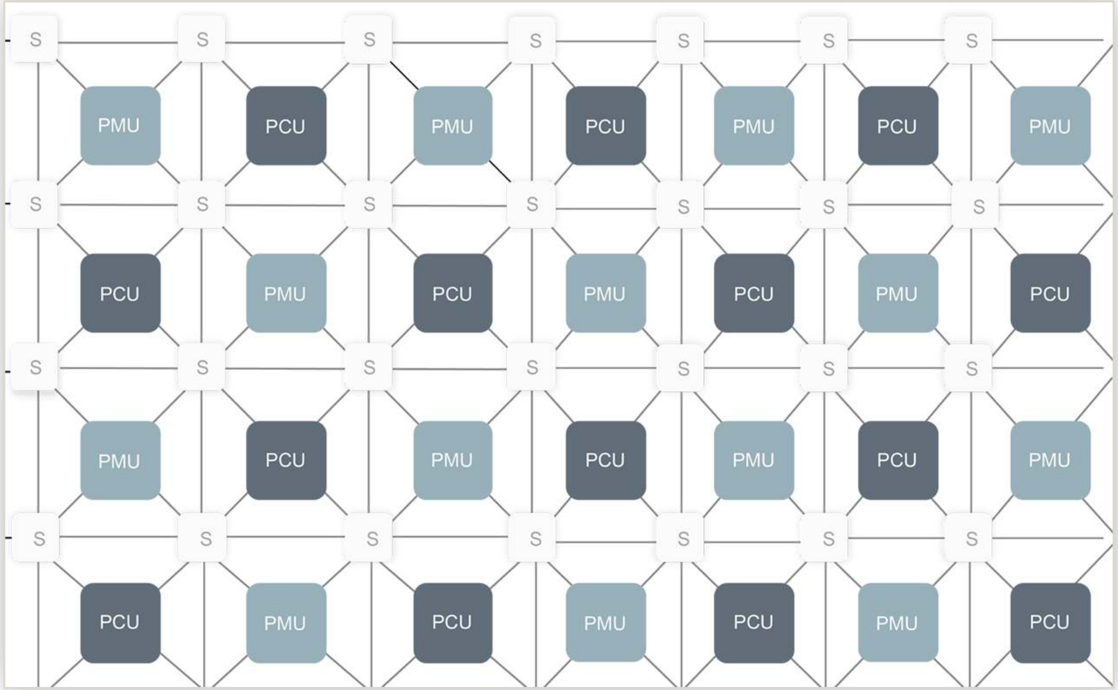
LAYERNORM: $$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$



Example: Softmax

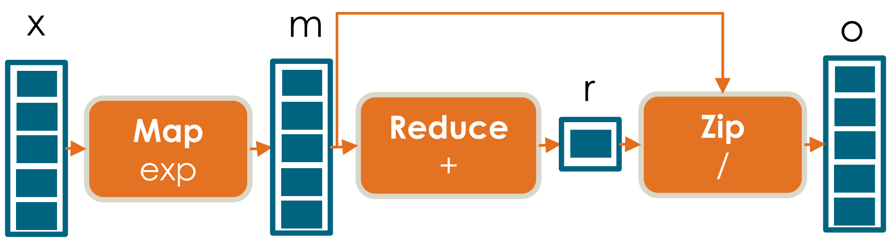
SOFTMAX: $$\mathrm{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

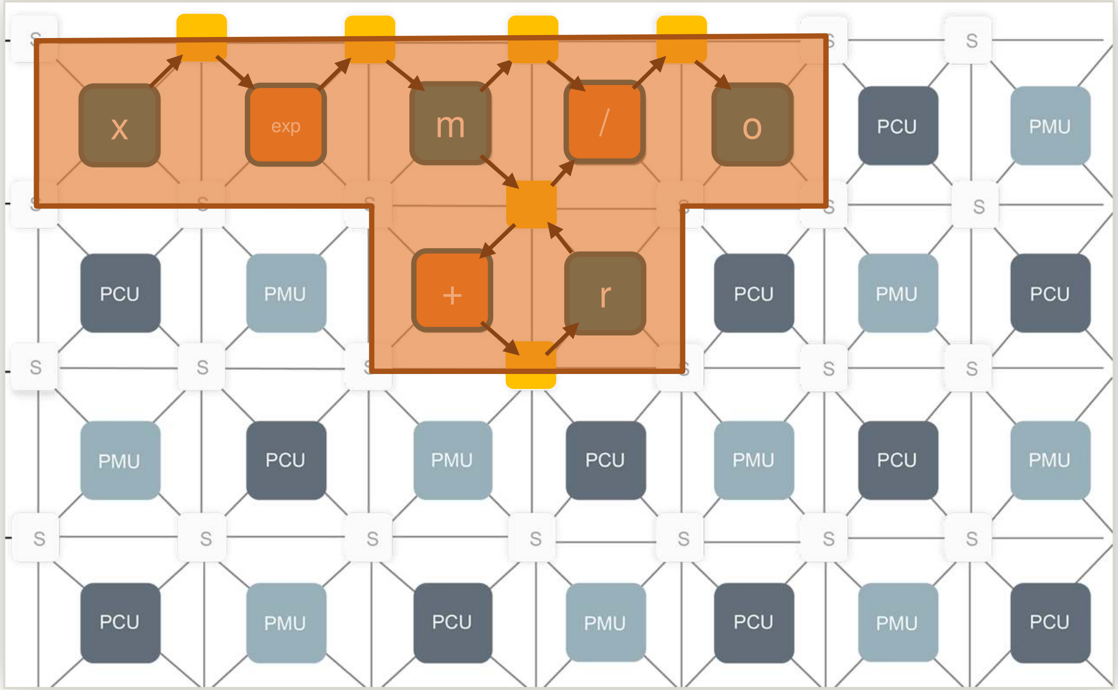




Example: Softmax

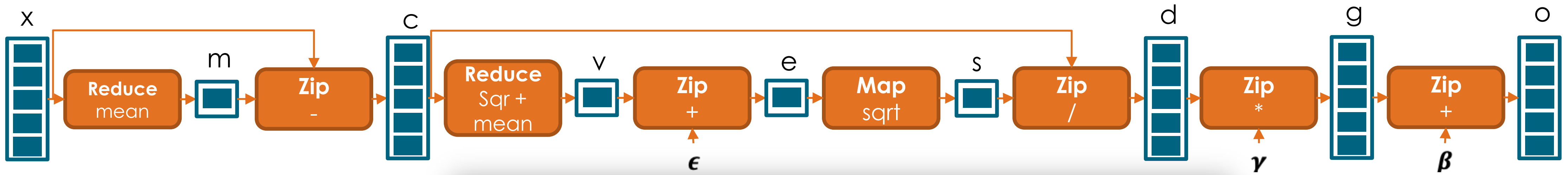
SOFTMAX: $$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

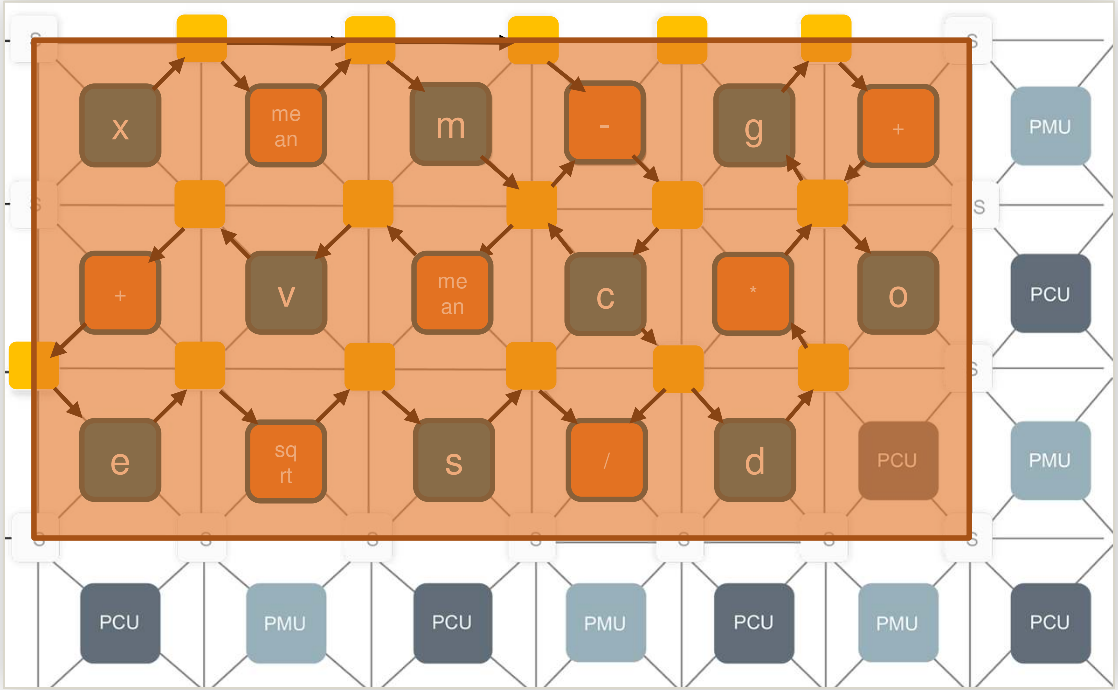




Example: LayerNorm, Pipelined in Space

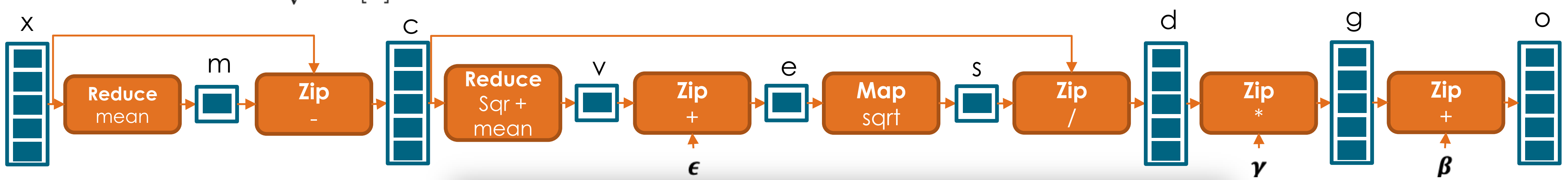
LAYERNORM: $$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

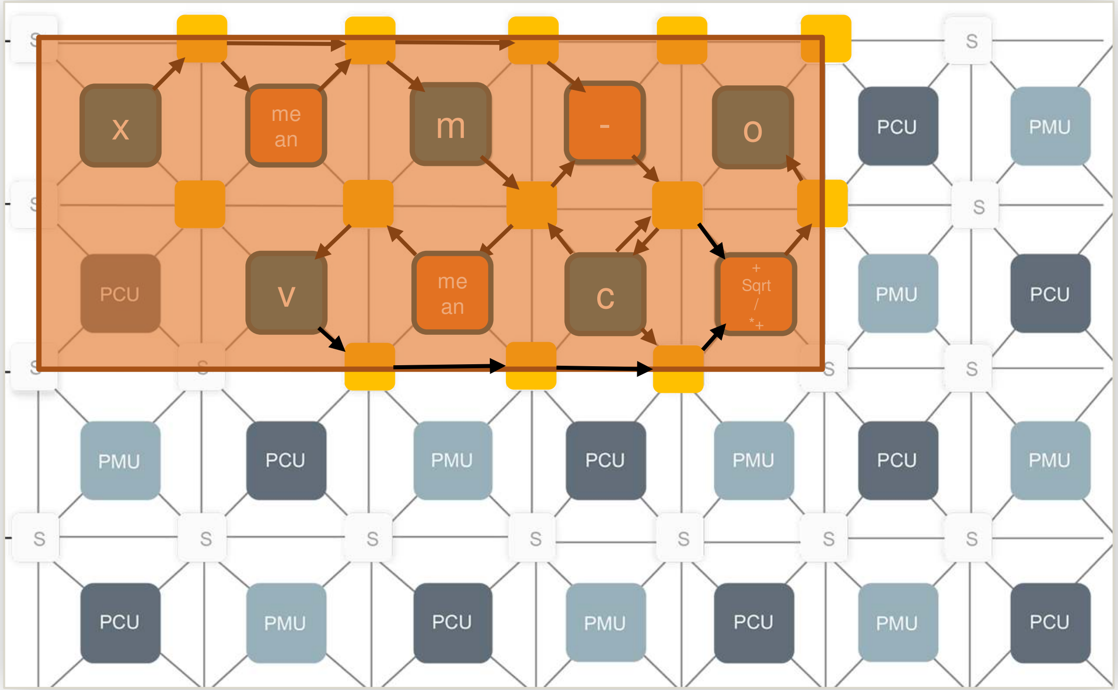




Example: LayerNorm, Pipelined in Space + Fused

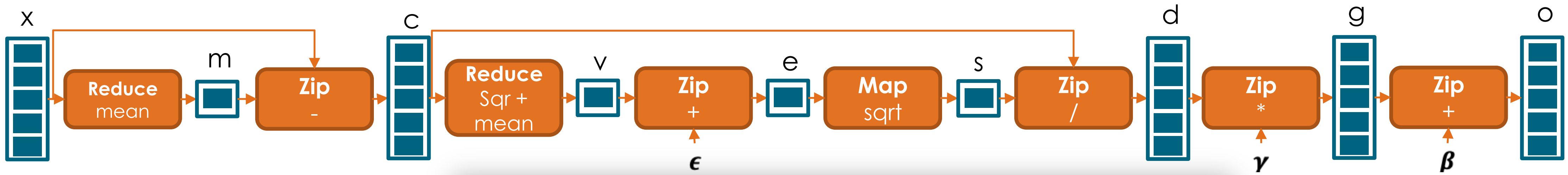
LAYERNORM: $$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

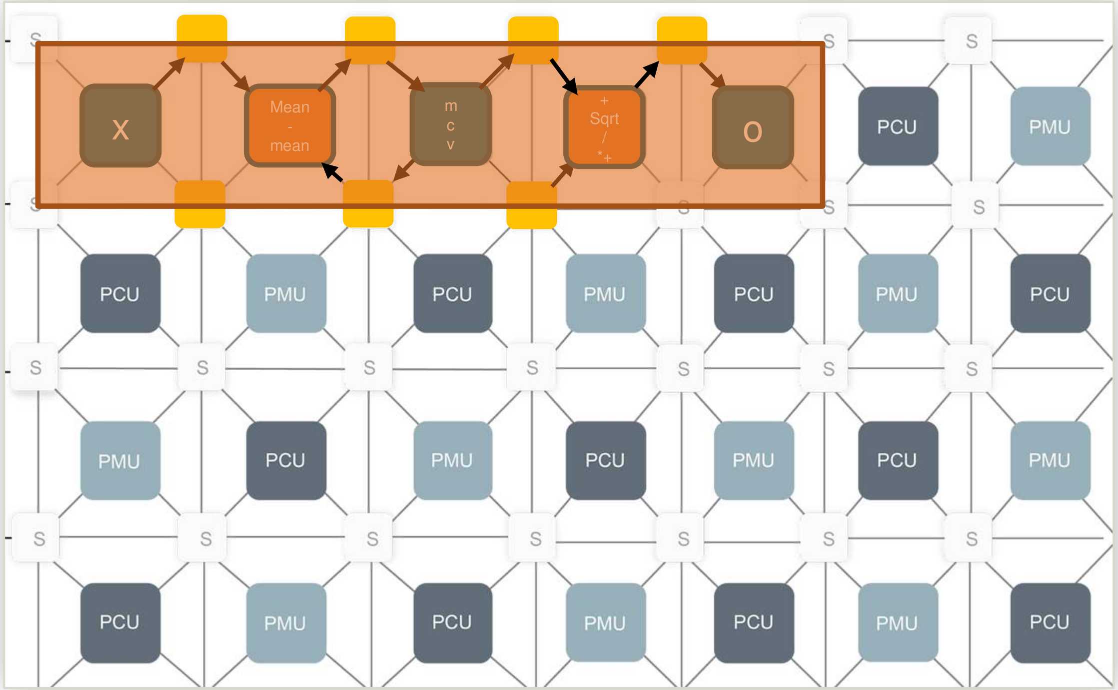




Example: LayerNorm, Hybrid Space + Time Execution

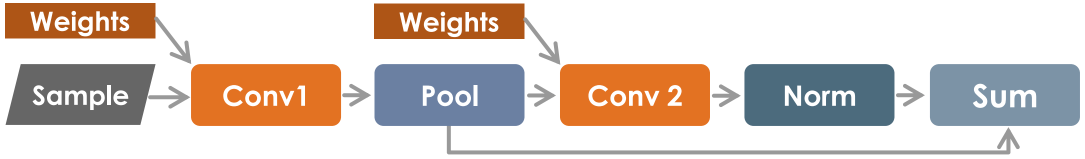
LAYERNORM: $$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

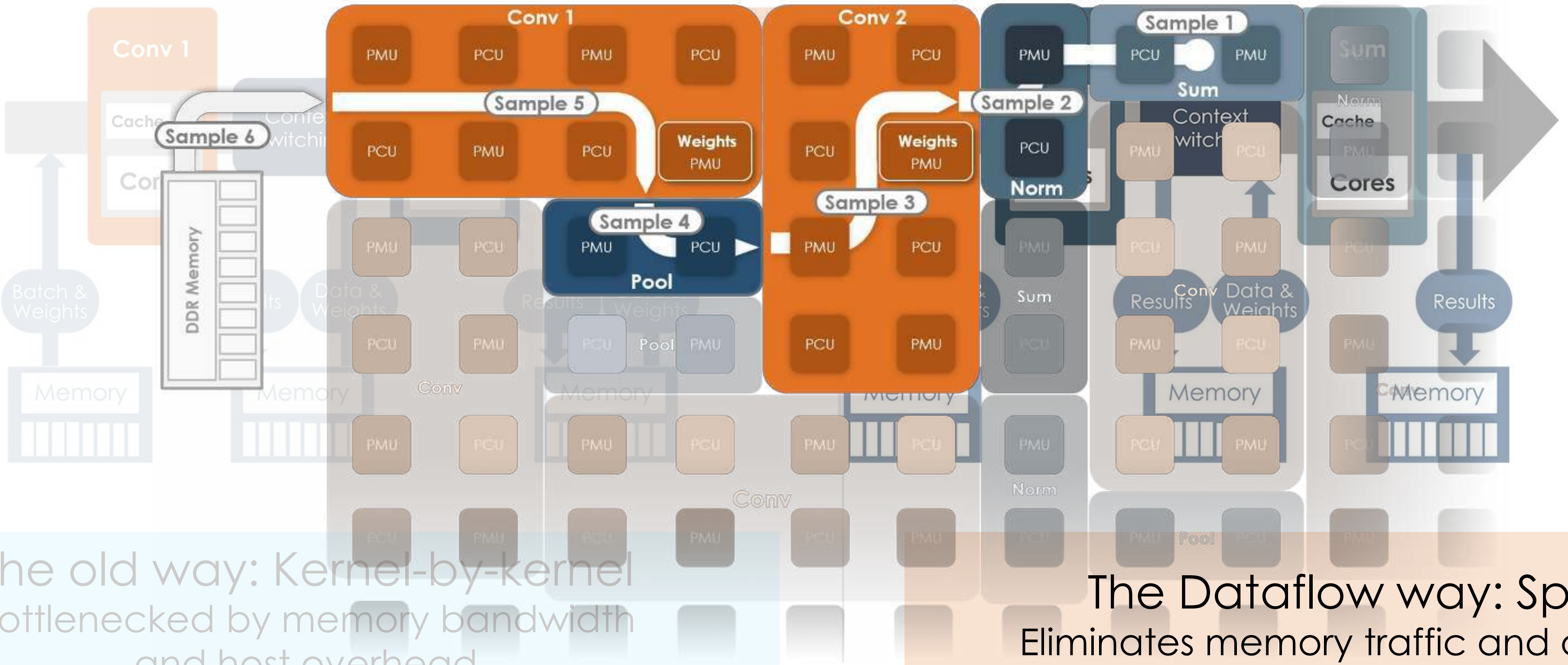




Spatial Dataflow Within an RDU

CONVOLUTION GRAPH

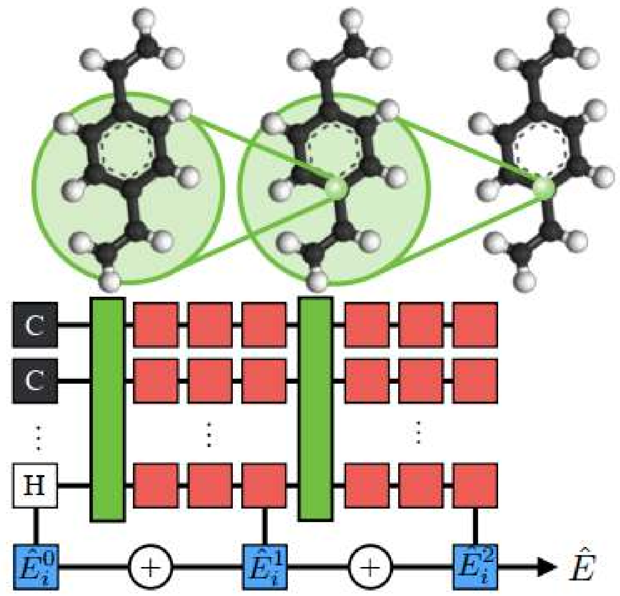




The old way: Kernel-by-kernel
Bottlenecked by memory bandwidth and host overhead

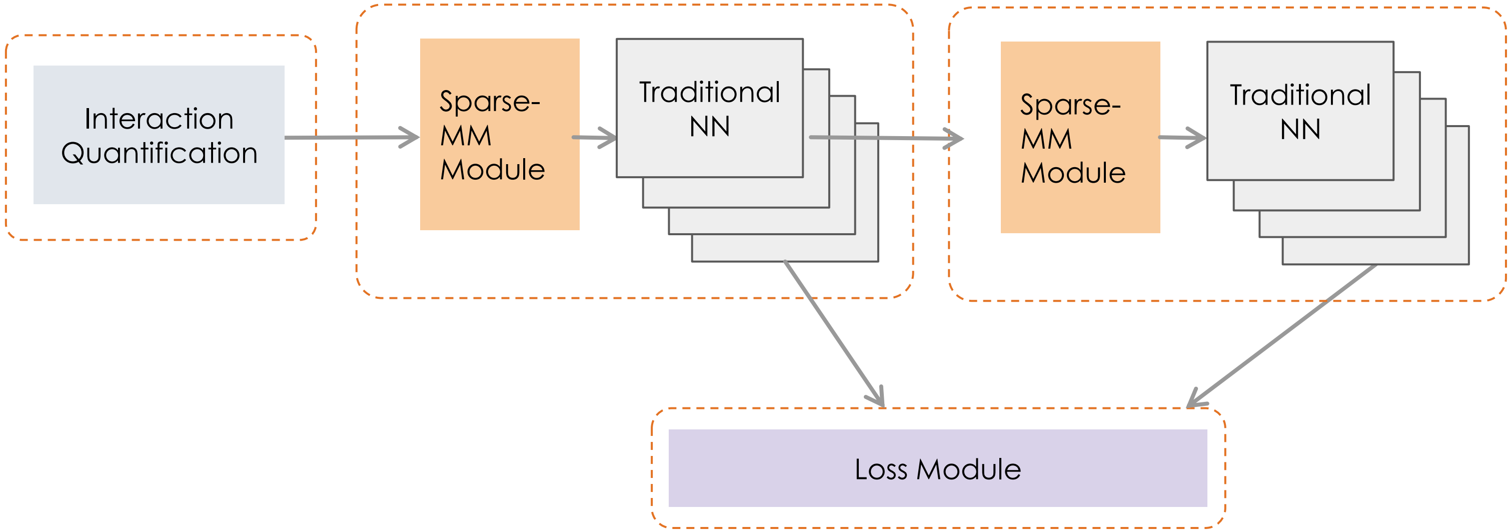
The Dataflow way: Spatial
Eliminates memory traffic and overhead

SpMM Within Application – Kernel fusion



- Accurate modeling of intra-molecular interactions using SpMM + NN

- Representation of atom interactions: Sparse CSR-like format



Sparse Matrix Multiply on RDU

Logical

$$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -2 \\ 0 & -3 & -4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} -3 & -4 \\ -10 & -12 \\ -29 & -36 \end{bmatrix}$$

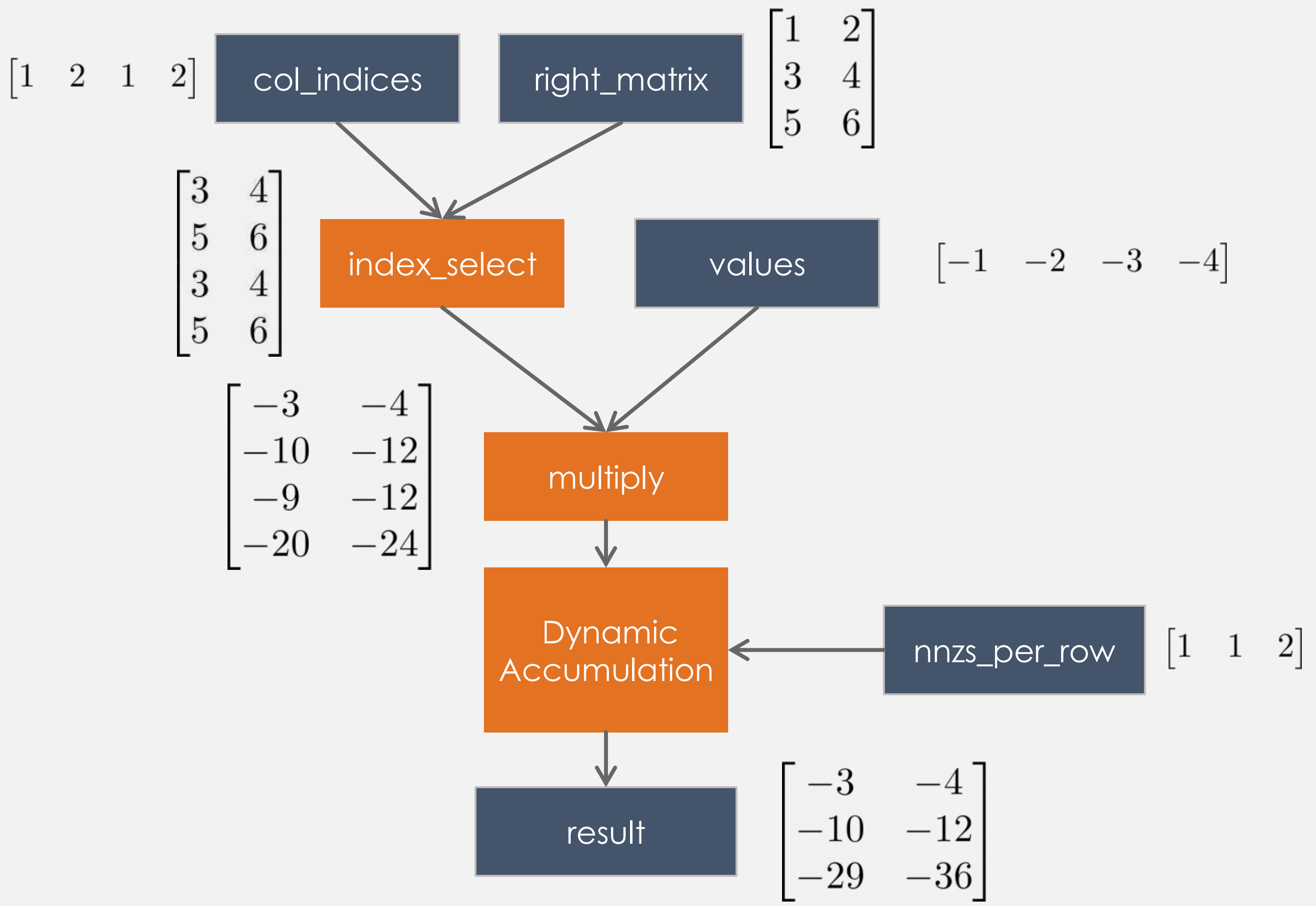
$\downarrow$ CSR-like Representation

$$col\_indices = \begin{bmatrix} 1 & 2 & 1 & 2 \end{bmatrix}$$

$$values = \begin{bmatrix} -1 & -2 & -3 & -4 \end{bmatrix}$$

$$nnzs\_per\_row = \begin{bmatrix} 1 & 1 & 2 \end{bmatrix}$$

Spatial



Spatial Dataflow Architecture

Hierarchical parallel pattern Dataflow
Natural ML execution model, Communication-efficiency, Ease-of-use

Terabyte sized models
Large embeddings, True-resolution, Flexible batch-size

Sparsity
Graph based neural networks

Flexible mapping
Model and data parallelism

Data processing
SQL in inner loop of ML training

Dataflow Architecture for Terabyte Sized Models

DataScale SN10-8R
¼ Rack System

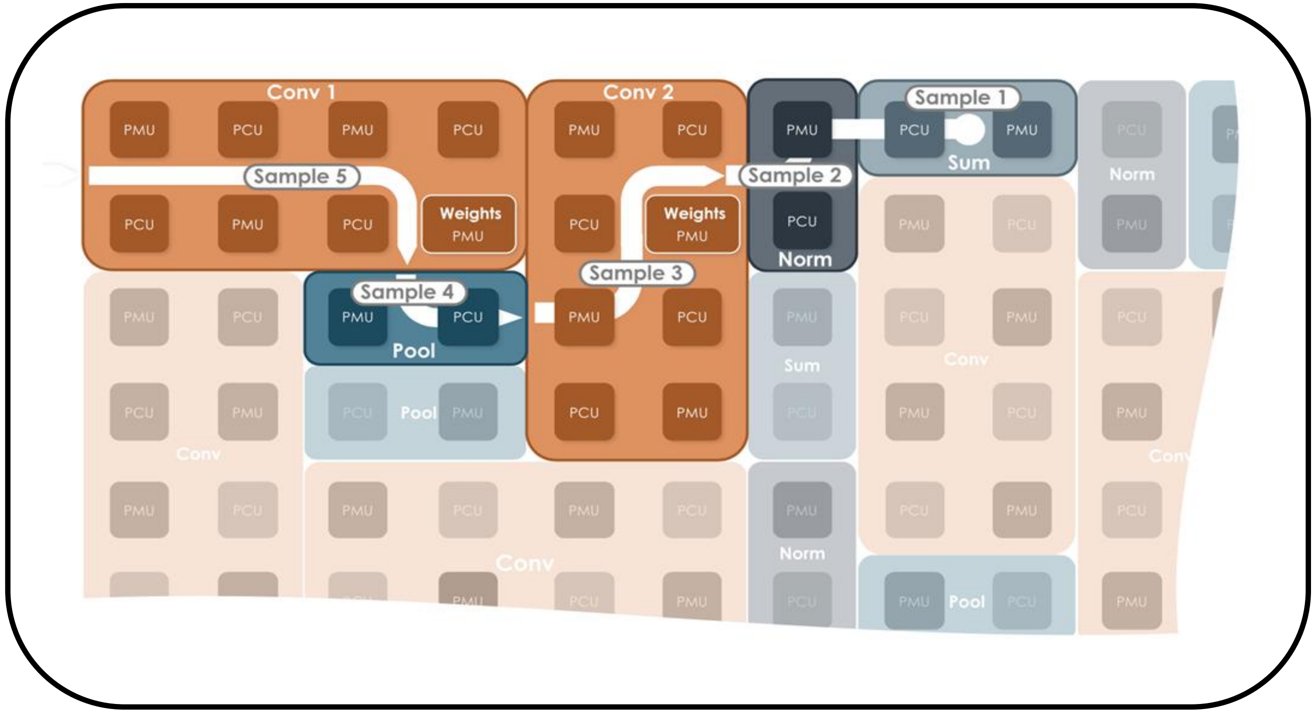
Dataflow Efficiency

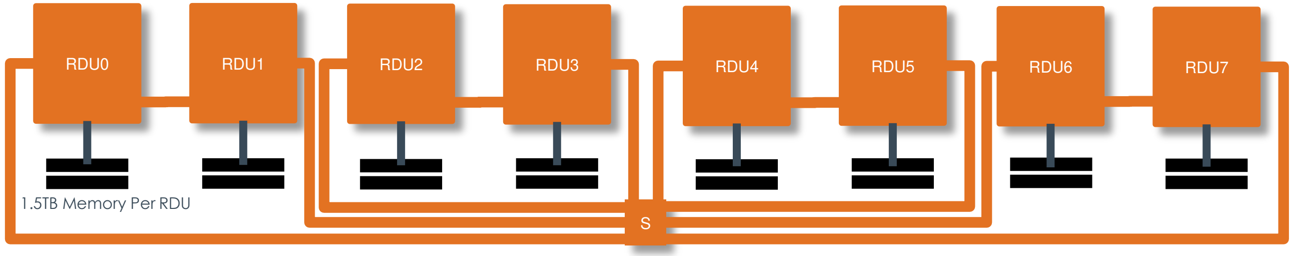
+

Compute Capability

+

Large Memory Capacity

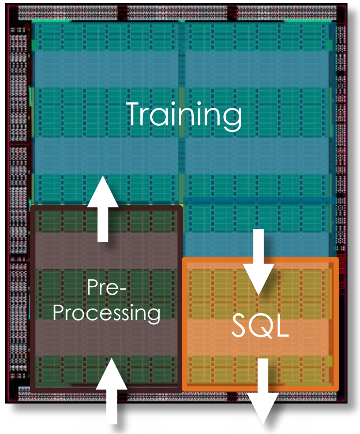




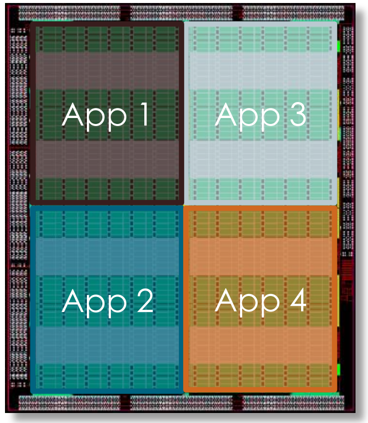
SambaNova Systems Flexibility to Support Key Scenarios

4 RDU deployment examples

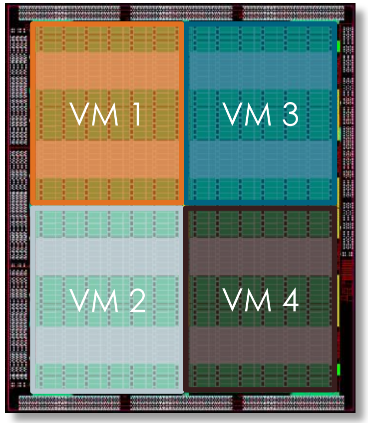
1) High Performance Mixed Workloads



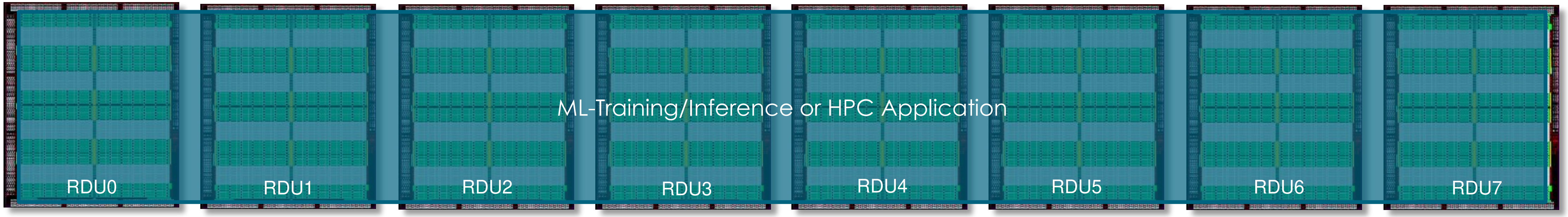
2) Efficient Concurrent Applications



3) Secure Multi-Tenancy

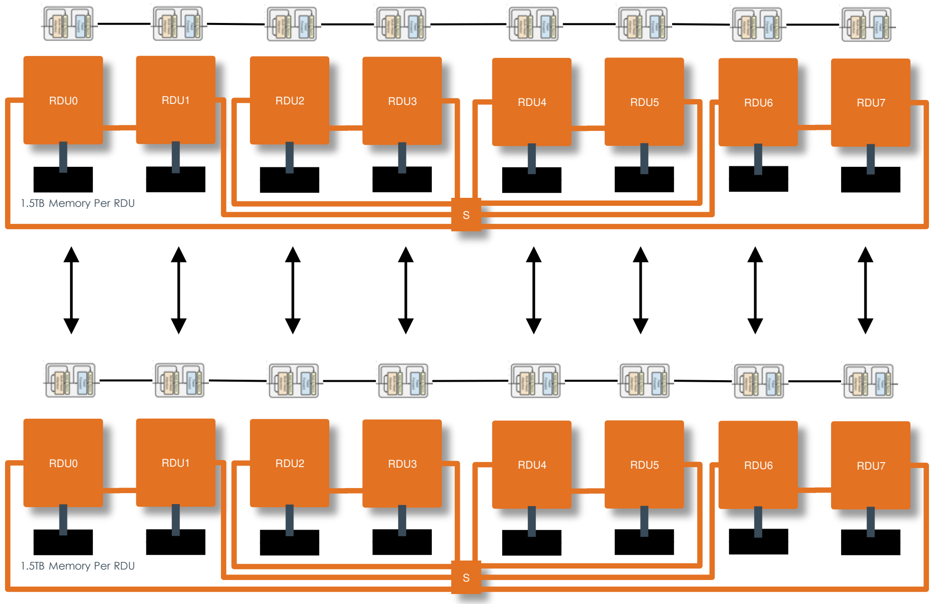


4) Compiler Driven Application Scale-Up



DataScale Systems Scale-Out

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



DataScale MP/DP Scale-Out

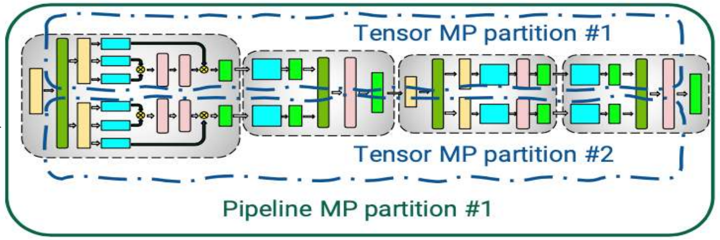
Train Large Language Models, Without Code Changes

1T parameter NLP training with a small footprint and programming ease

Model size	Attention heads	Hidden size	Number of layers	Number of parameters (billion)	Model-parallel size
1.7B	24	2304	24	1.7	1
3.6B	32	3072	30	3.6	2
7.5B	32	4096	36	7.5	4
18B	48	6144	40	18.4	8
39B	64	8192	48	39.1	16
76B	80	10240	60	76.1	32
145B	96	12288	80	145.6	64
310B	128	16384	96	310.1	128
530B	128	20480	105	529.6	280
1T	160	25600	128	1008.0	512

Large Kernels
- To Hide Communication Costs
- Statistically Nonoptimal

Complex System Engineering To Enable Model Architecture Exploration



PyTorch

Out-of-Box Models
- Huggingface Models
- Write yours in Pytorch

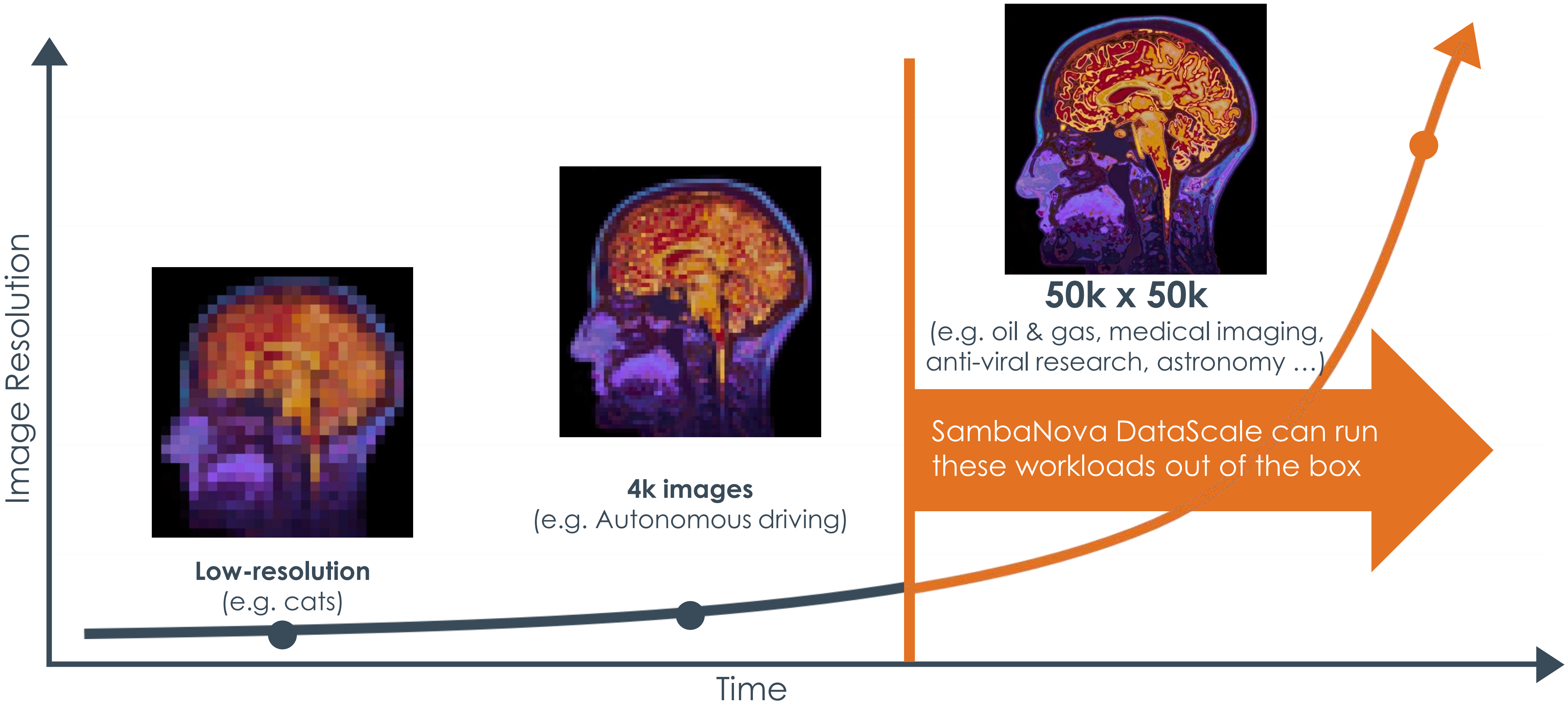
Developer Efficiency
- Focus on ML problems instead of System Engineering

High Accuracy Models
- No Compromise on Model Architecture required to hide System Deficiencies

https://arxiv.org/pdf/2104.04473.pdf

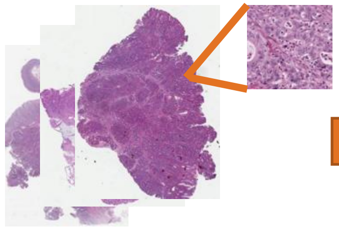
Train with 4k to 50k Convolutions, Without Code Changes

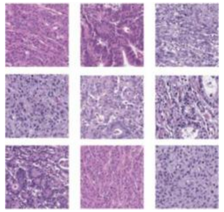
SambaNova trains true-resolution Computer Vision models effortlessly

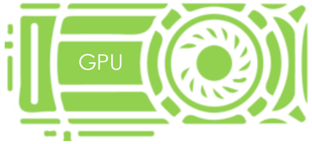


Enabling High-Resolution Full-Image Pathology

Info loss

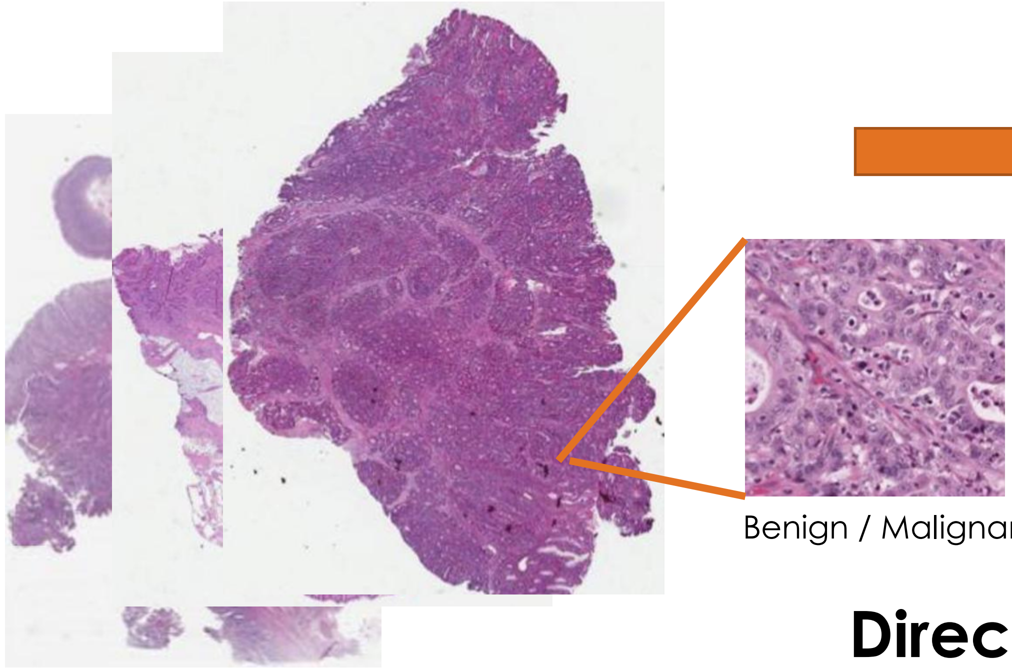






Downsample

Tiling



Benign / Malignant

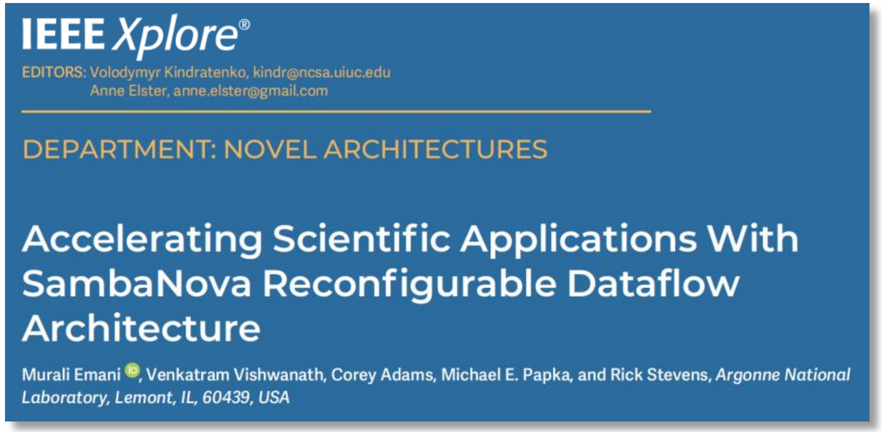
Direct analysis with 100X higher resolution

World Record Accuracy High-Res Convolution Training

Out of the box world record accuracy

90.23%

Accuracy



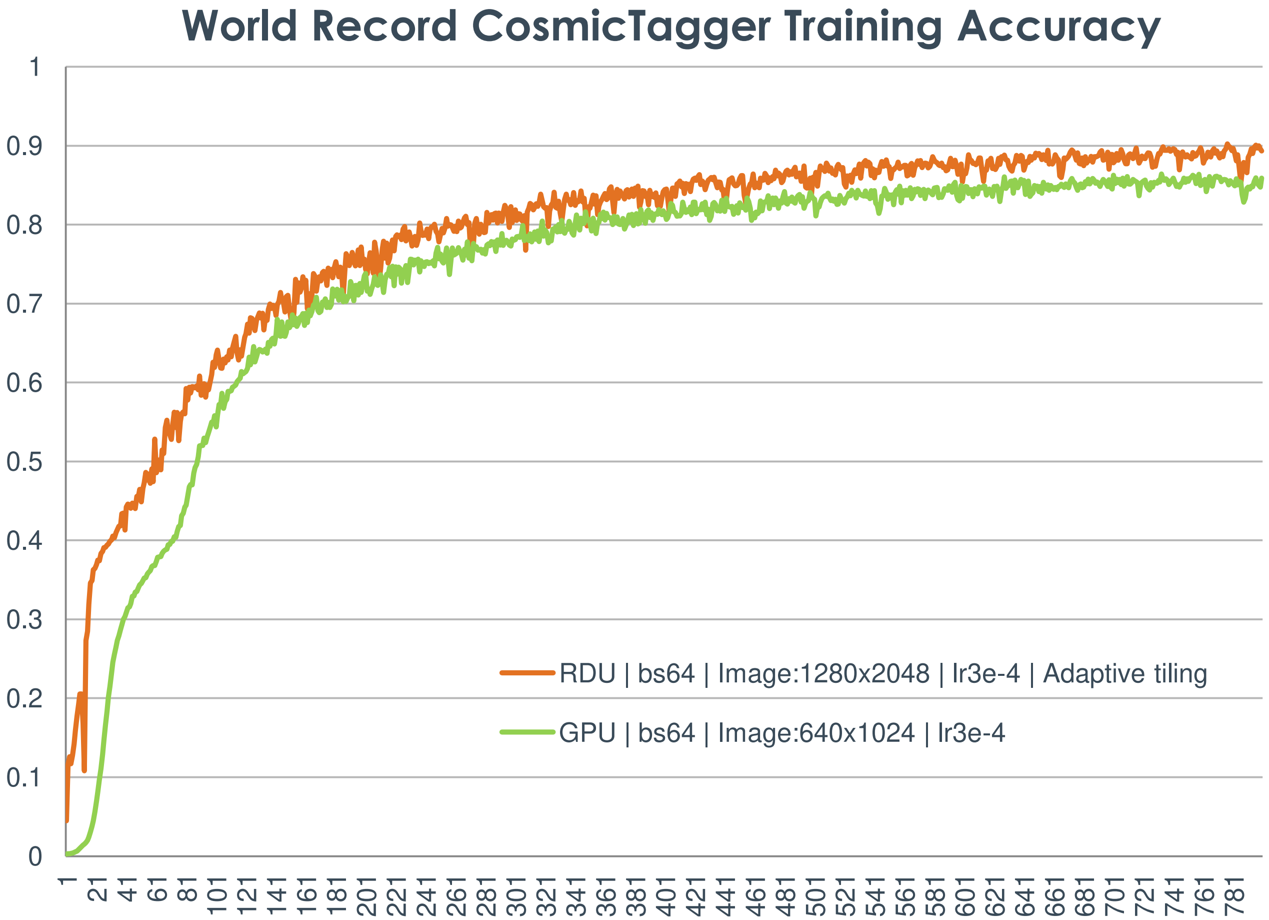
IEEE Xplore®

EDITORS: Volodymyr Kindratenko, kindr@ncsa.uiuc.edu
Anne Elster, anne.elster@gmail.com

DEPARTMENT: NOVEL ARCHITECTURES

Accelerating Scientific Applications With SambaNova Reconfigurable Dataflow Architecture

Murali Emani, Venkatram Vishwanath, Corey Adams, Michael E. Papka, and Rick Stevens, *Argonne National Laboratory, Lemont, IL, 60439, USA*



SambaNova Systems: AI for Science

Accelerating scientific applications

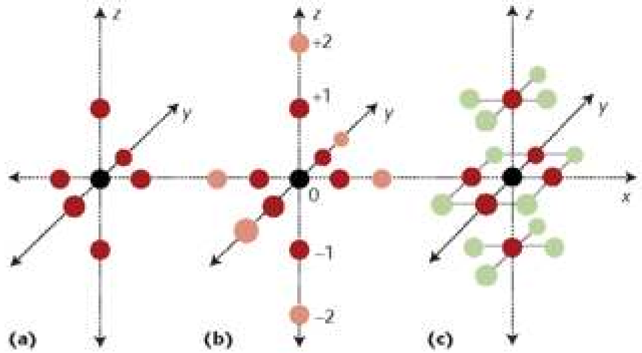
DFT

$$H \cdot X = -\frac{1}{2}\nabla^2 \cdot X + FFT_{3D} \cdot (V + \Phi_R \cdot \Phi_R^T) \cdot iFFT_{3D} \cdot X$$

= Pointwise Computations + * Sparse Tensor Contractions * Padded/Pruned FFT Computations

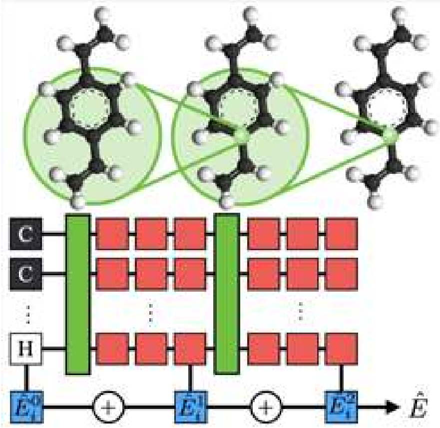
The base algorithm for HPC
Public sector, energy, BFSI, manufacturing, automotive, pharma, biomed

Finite Difference



Differential equation approximation
Heat transfer, stress/strain mechanics, fluid dynamics

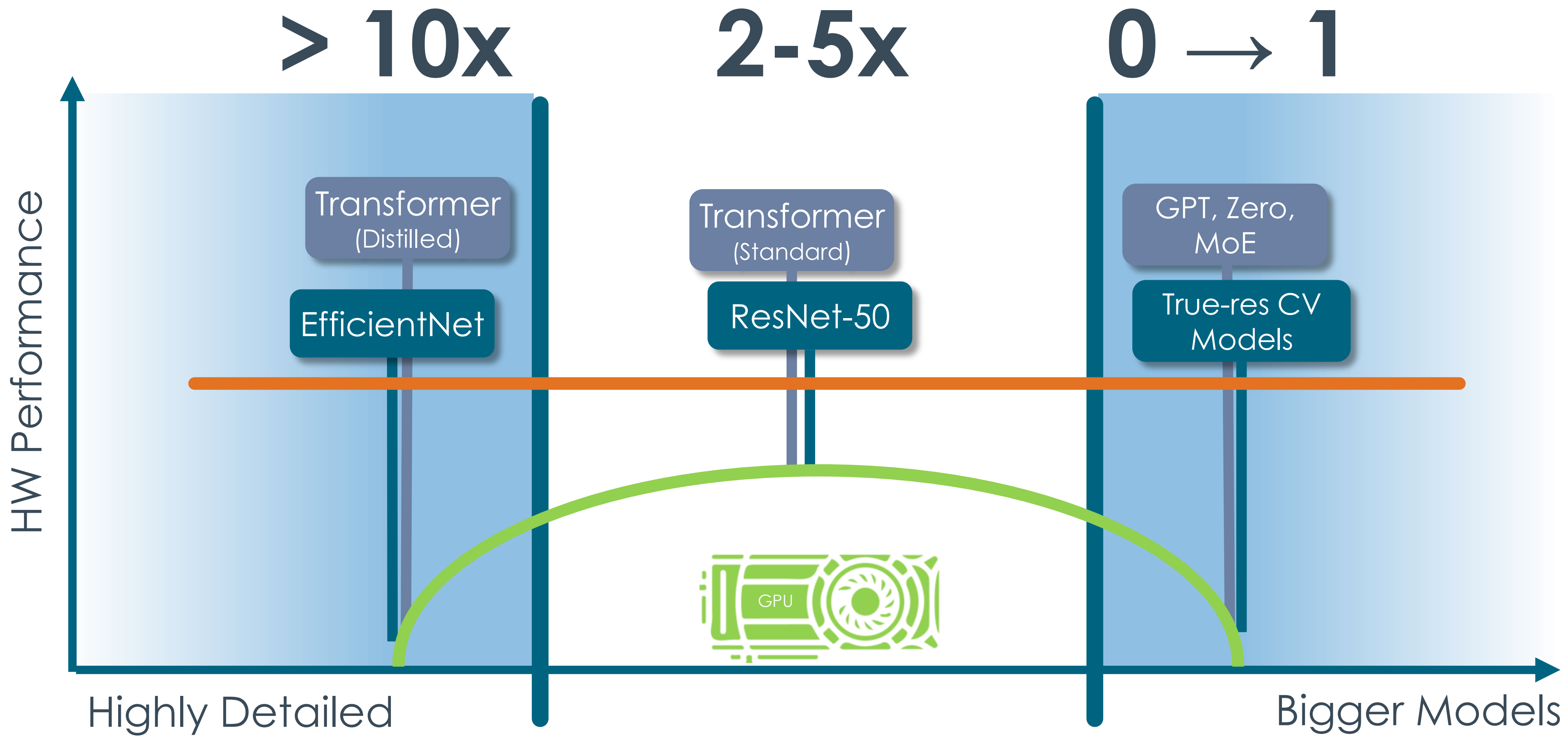
Molecular Modeling



Molecular energies modeling
Materials science, chemistry, molecular biology, drug design

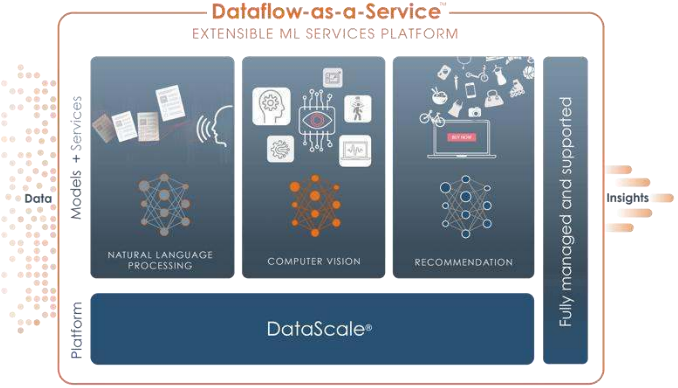
These Problems Are All Dataflow

Surpassing State-of-the-Art Accuracy and Performance with Dataflow



Learn More About Dataflow!

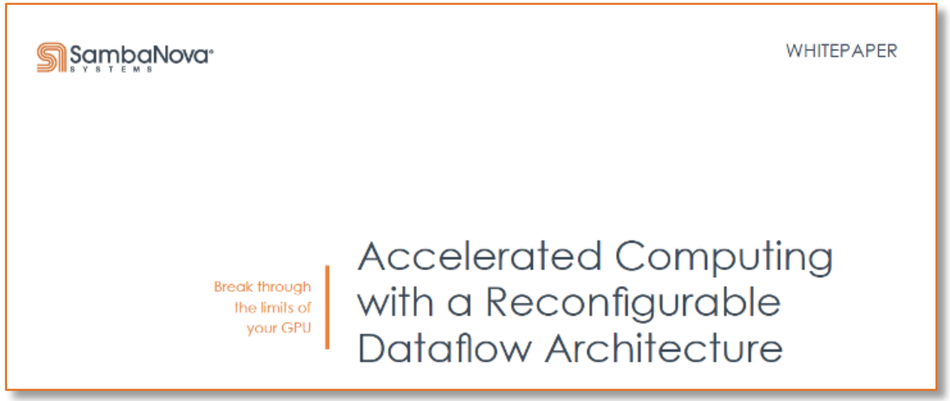
Visit the product pages



https://sambanova.ai/products/dataflow-as-a-service/

https://sambanova.ai/products/datascale/

Download the RDA whitepaper


SambaNova SYSTEMS

WHITEPAPER

Break through the limits of your GPU

Accelerated Computing with a Reconfigurable Dataflow Architecture

https://sambanova.ai/wp-content/uploads/2021/06/SambaNova_RDA_Whitepaper_English.pdf

We are hiring!

Stay up to date on the latest

sambanova.ai

sambanova-systems

@SambaNovaAI

SambaNovaAI

SambaNova®
SYSTEMS