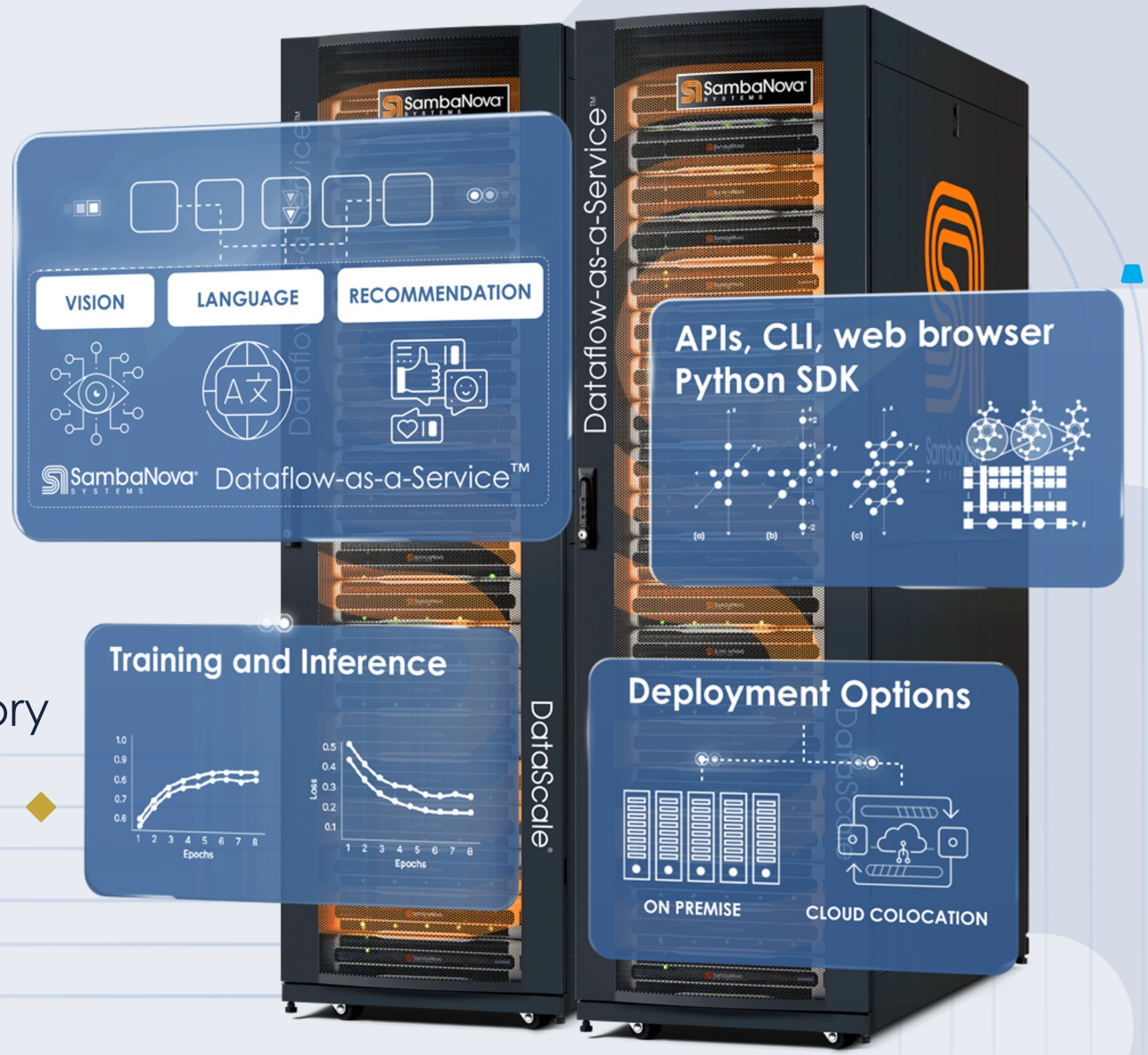

SambaNova®
SYSTEMS

Reconfigurable Dataflow Architecture

Jennifer Glore, Vice President
Martin Mueller, Customer Engineer

Murali Emadi, Argonne National Laboratiory

May 2023



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

Accelerate time to value with production quality, enterprise-scale AI solutions.

Industry leading AI-as-a-Service

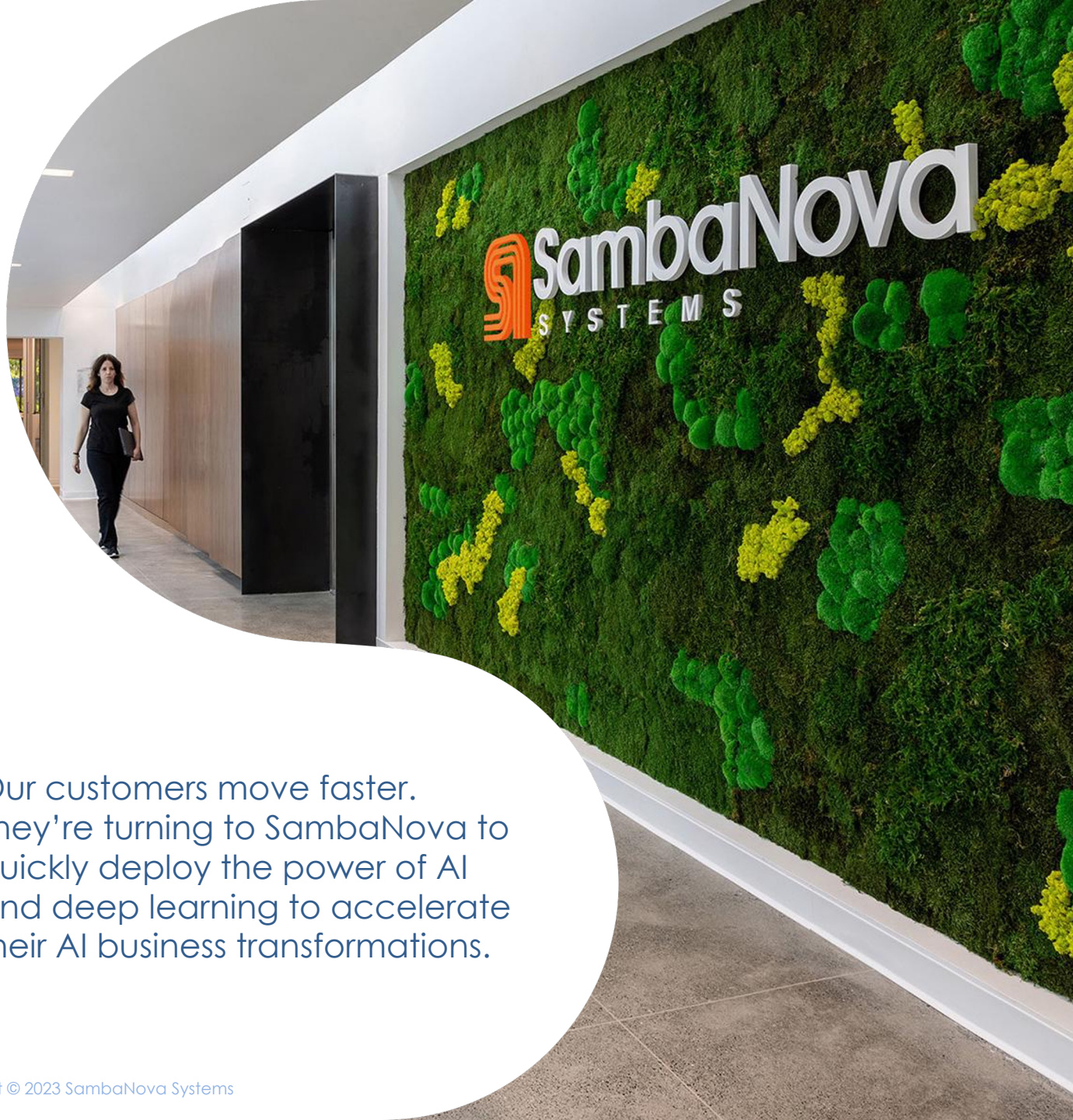
- Purpose-built enterprise-scale AI platform, from silicon to software
- Deployed in minutes

Pretrained Foundation Models

- Delivering the largest models
- Domain specific models
- Record setting accuracy and performance

Next generation architecture

- Dataflow, designed for AI
- Unshackled from limitations of legacy designs



Our customers move faster. They're turning to SambaNova to quickly deploy the power of AI and deep learning to accelerate their AI business transformations.

SambaNova SYSTEMS

The SambaNova Approach to AI

Innovation at every layer of the stack

SambaSuite®

as-a-SERVICE
Pre-trained Foundation Models

SYSTEMS
DataScale

SOFTWARE
SambaFlow™

SILICON
RDU

DataScale®

LLNL Scales Up SambaNova to Accelerate AI for Science

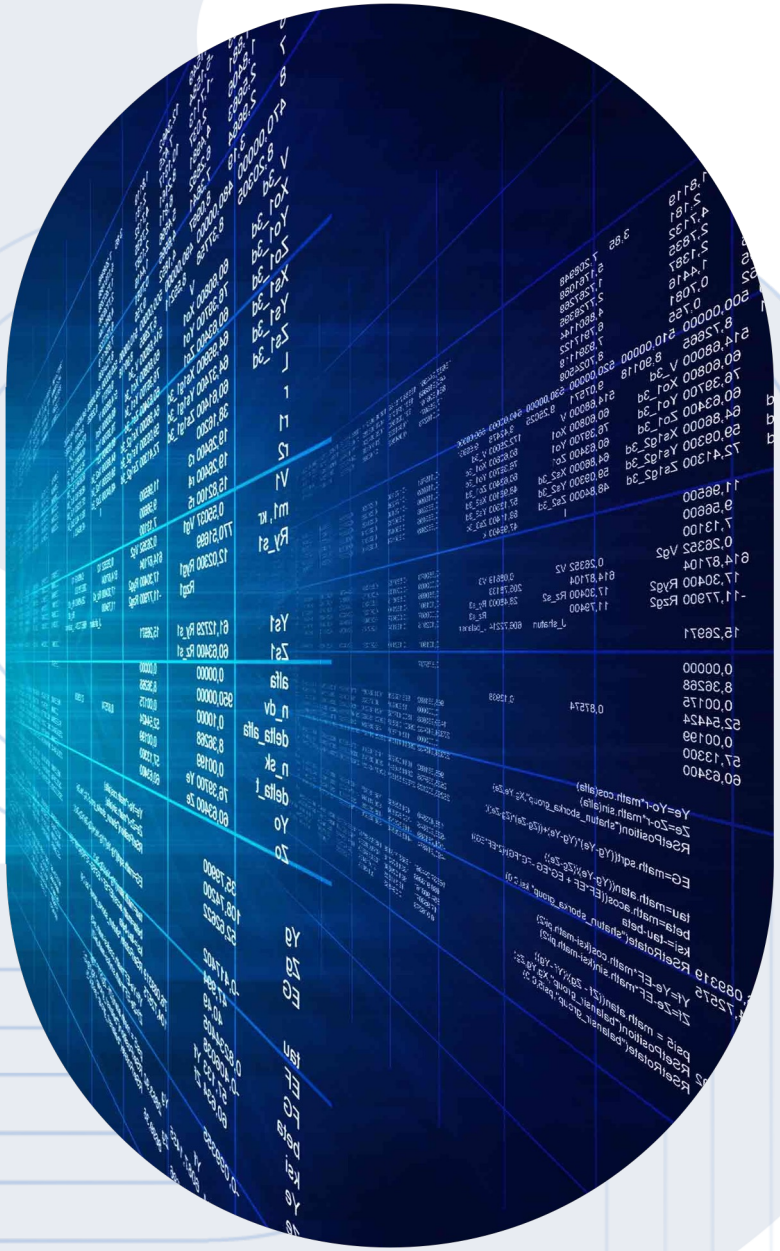
Complex physics, ICF reactions, small molecule drug design

"SambaNova has a different architecture than CPU or GPU-based systems, which we are leveraging to create an enhanced approach for CogSim that leverages a heterogeneous system combining the SambaNova DataScale with our supercomputing clusters."

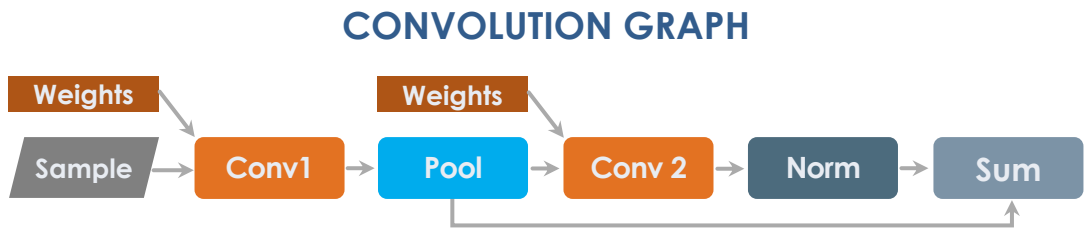
SambaNova SYSTEMS

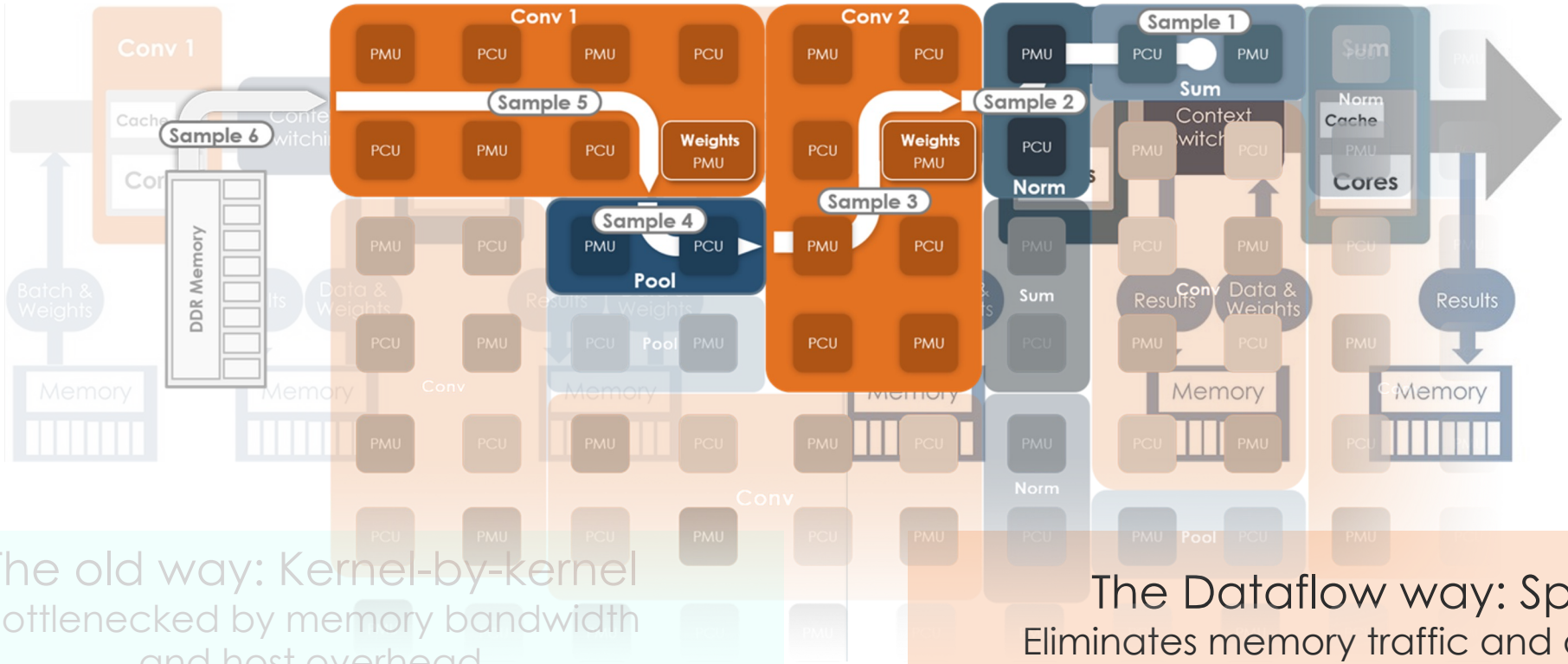
Architecture

SambaNova
Reconfigurable Dataflow Architecture



Spatial Dataflow Within an RDU





The old way: Kernel-by-kernel

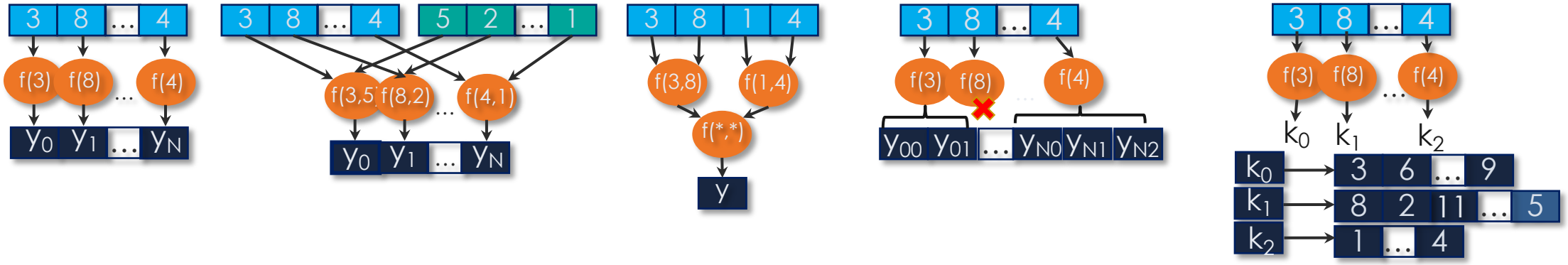
Bottlenecked by memory bandwidth and host overhead

The Dataflow way: Spatial

Eliminates memory traffic and overhead

Reconfigurable DataFlow Architecture: Parallel Patterns

https://stanford-ppl.github.io/website/papers/isca17-raghu-plasticine.pdf



Map
element-wise function f

y = vector + 4
y = vector * 10
y = sigmoid(vector)

Zip
element-wise function f (multi-collection)

y = vecA + vecB
y = vecA / vecB
y = max(vecA,vecB)

Reduce
combine all elements with f (f is associative)

y = vector.sum
y = vector.product
y = max(vector)

FlatMap
element-wise function ≥0 values out per element

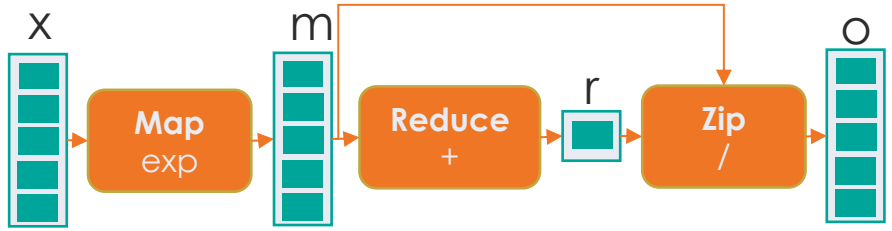
SELECT * FROM vector
WHERE elem < 5

GroupBy
group elements into buckets based on key

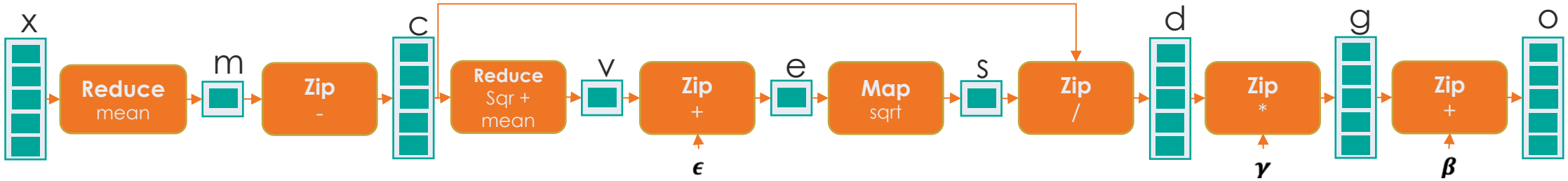
vector.groupBy{e => e % 3}

Programming Model

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



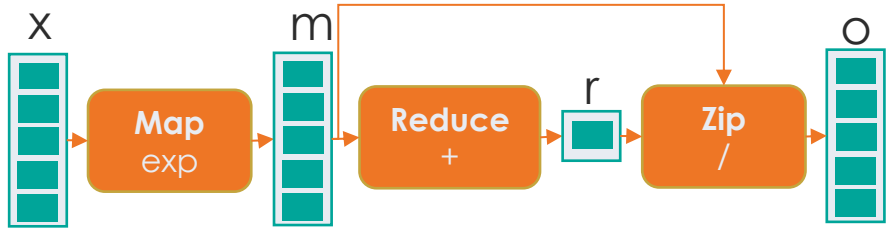
LAYERNORM: $y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$

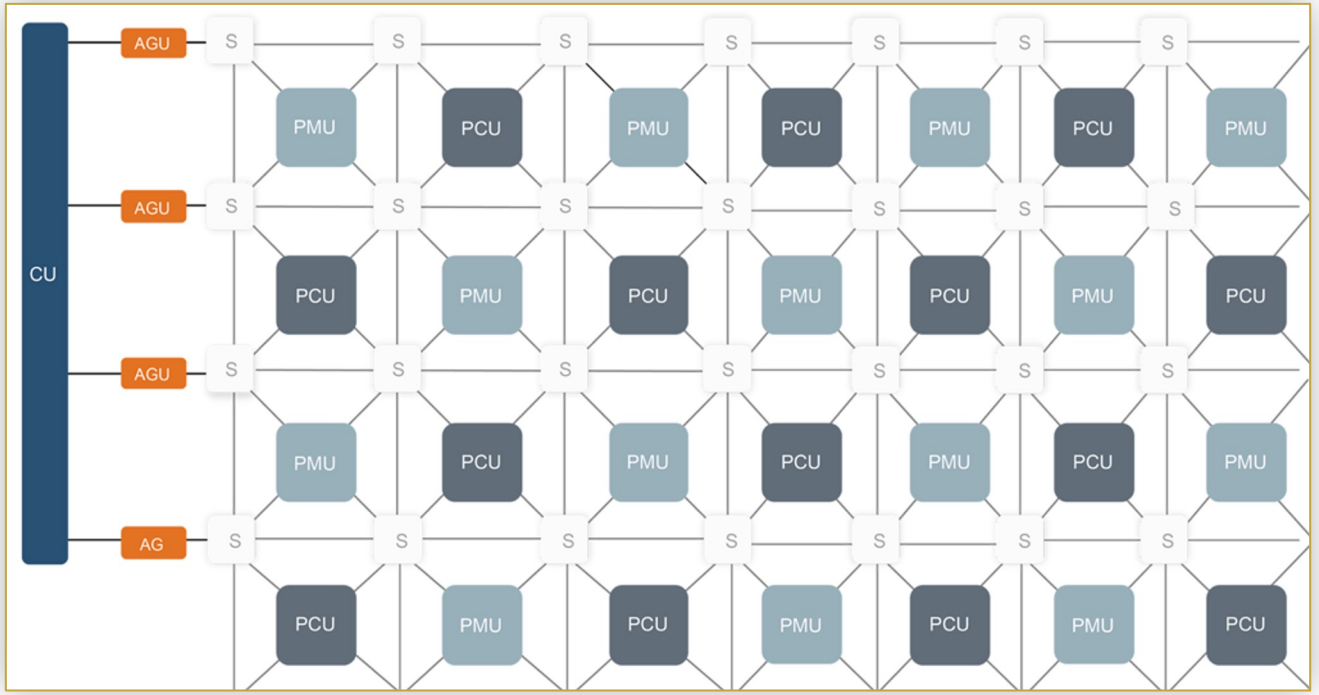


Example: Softmax

SOFTMAX :

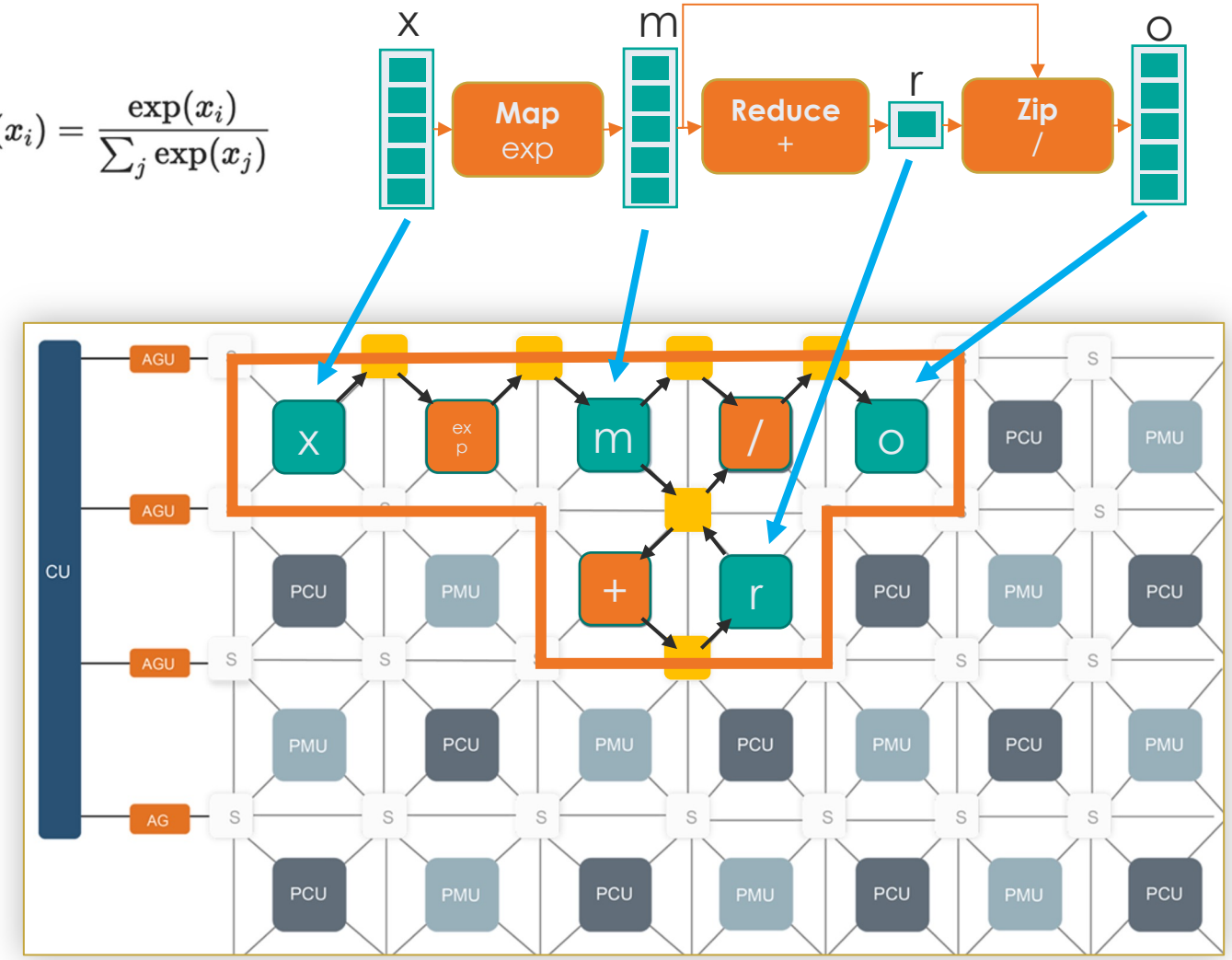
$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$





Example: Softmax

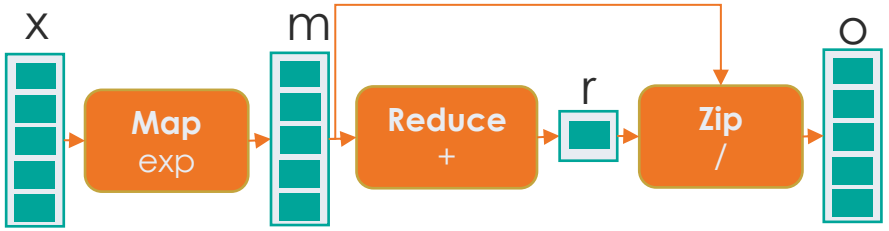
SOFTMAX: $\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$



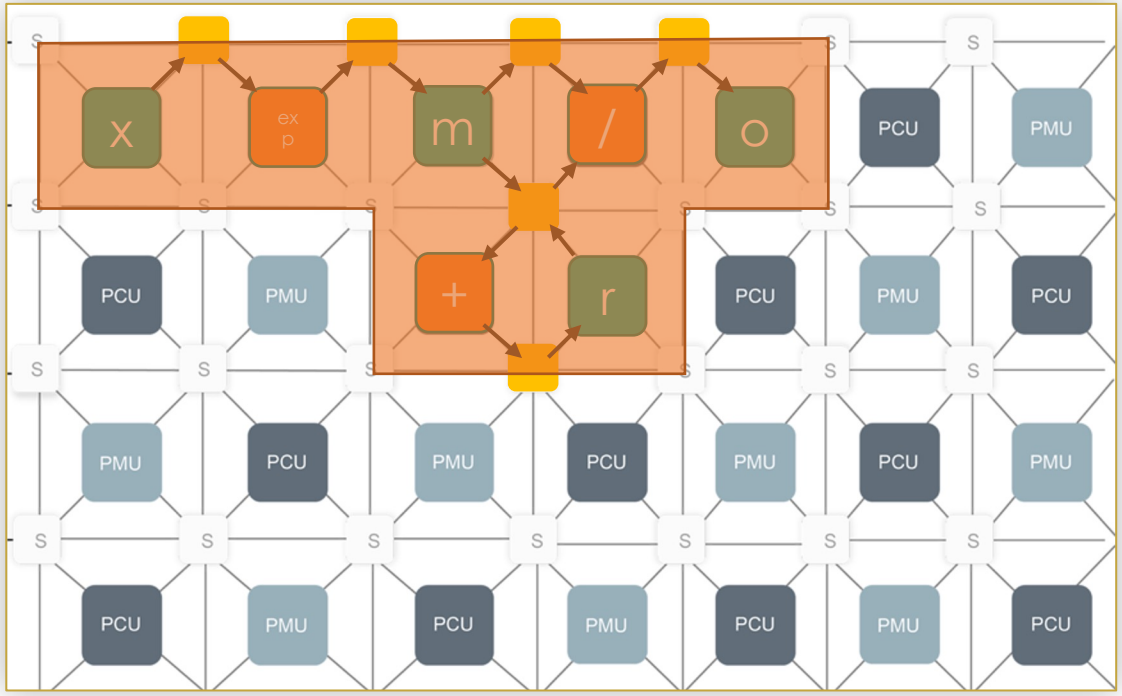
Example: Softmax

SOFTMAX :

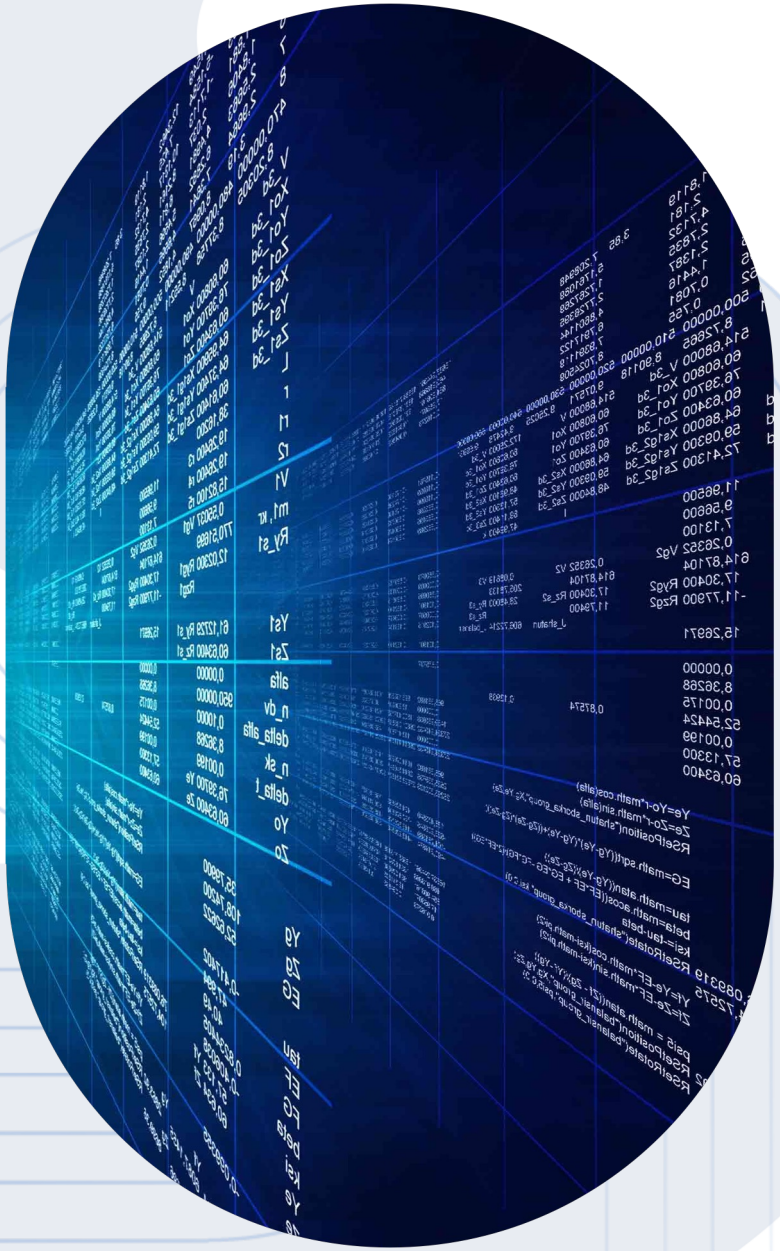
$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$




```
def softmax(inp, dim=1):
    exp_inp = sn.Map(inp, MAP_EXP)
    exp_sum = sn.Reduce(exp_inp, dim, REDUCE_SUM)
    sft_out = sn.Zip(exp_inp, exp_sum, ZIP_DIV)
    return sft_out
```



SambaNova Systems Platform



SambaNova DataScale SN30

DataScale SN30

- Rack optimized, integrated system
- Each System is 10 RU
  - 8 x SN30™ RDU
  - 8 TB DRAM
- Sambaflow Software Stack
- Can be installed in minutes

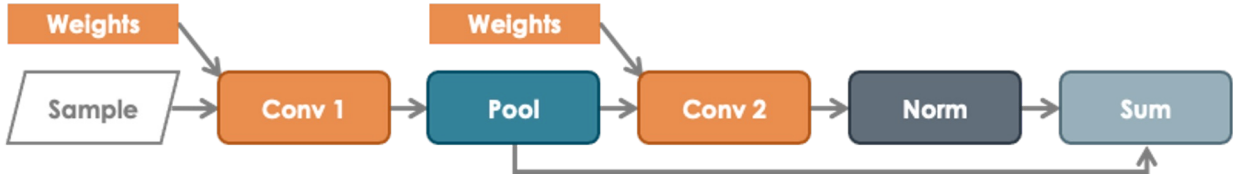
as-a-SERVICE
Pre-trained Foundation Models

SYSTEMS
DataScale®

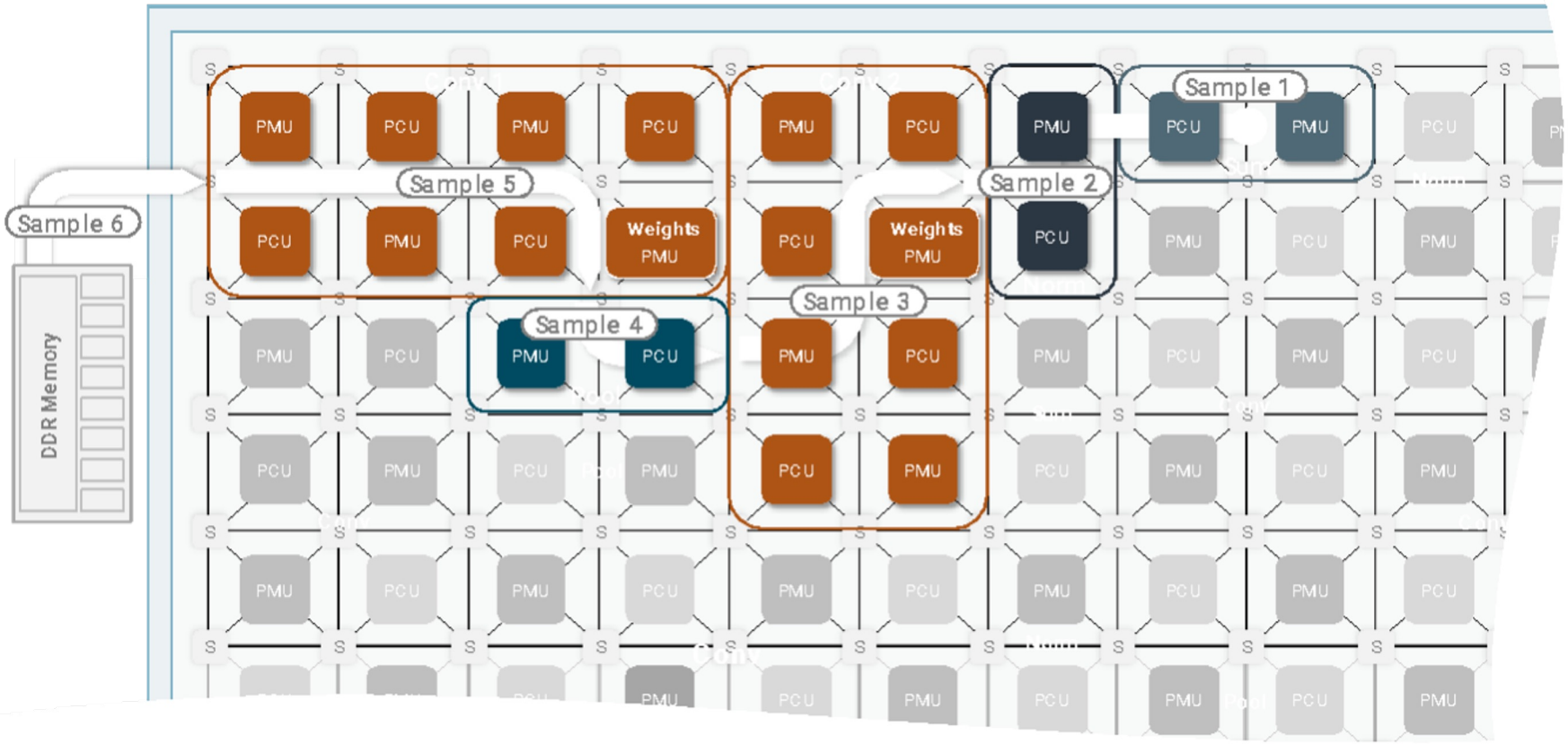
SOFTWARE
SambaFlow™

SILICON
RDU

SambaFlow Software Stack

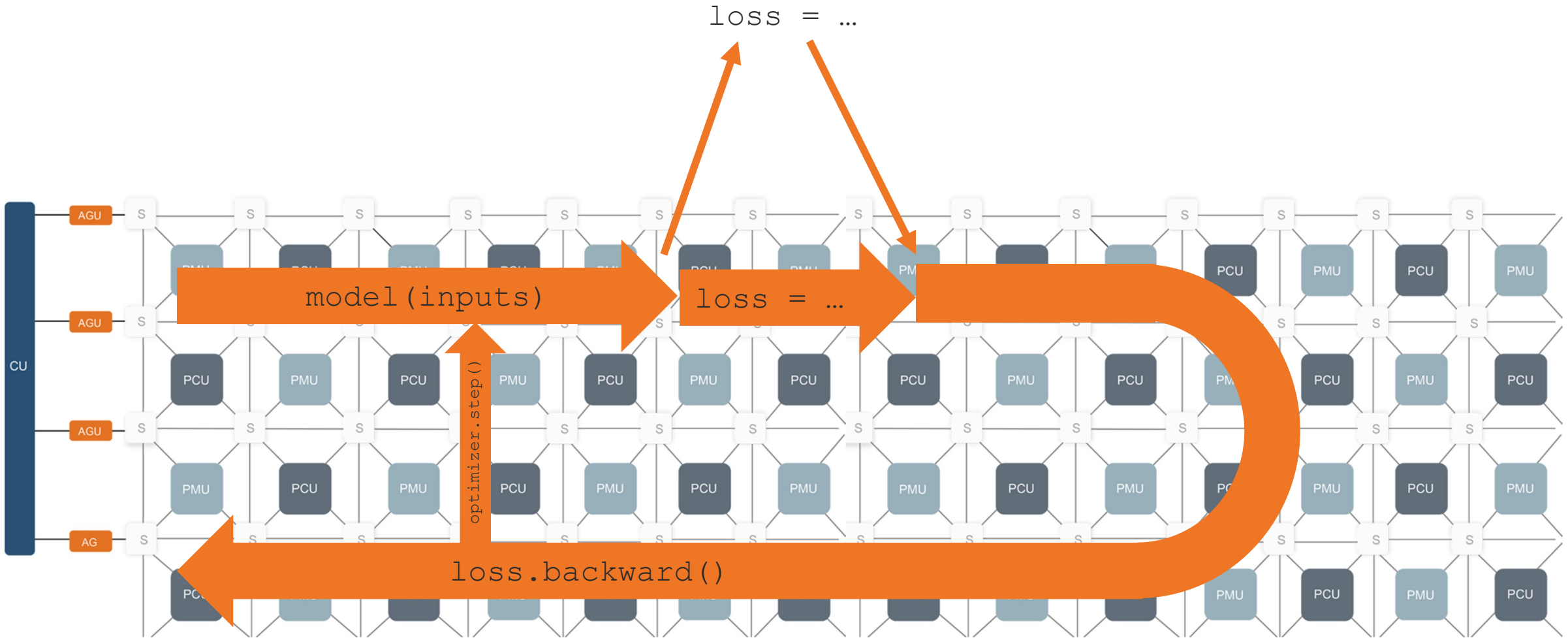



```
def forward(self, inputs, labels):
        x = self.conv1(inputs).relu()
        x = self.maxpool1(x)
        x = self.conv2(x).relu()
        x = self.maxpool2(x)
        x = torch.reshape(x,[x.shape[0],-1])
        x = self.fc1(x).relu()
        x = self.fc2(x).relu()
        out = self.fc3(x)
        loss = self.criterion(out, labels)
        return loss, out
```

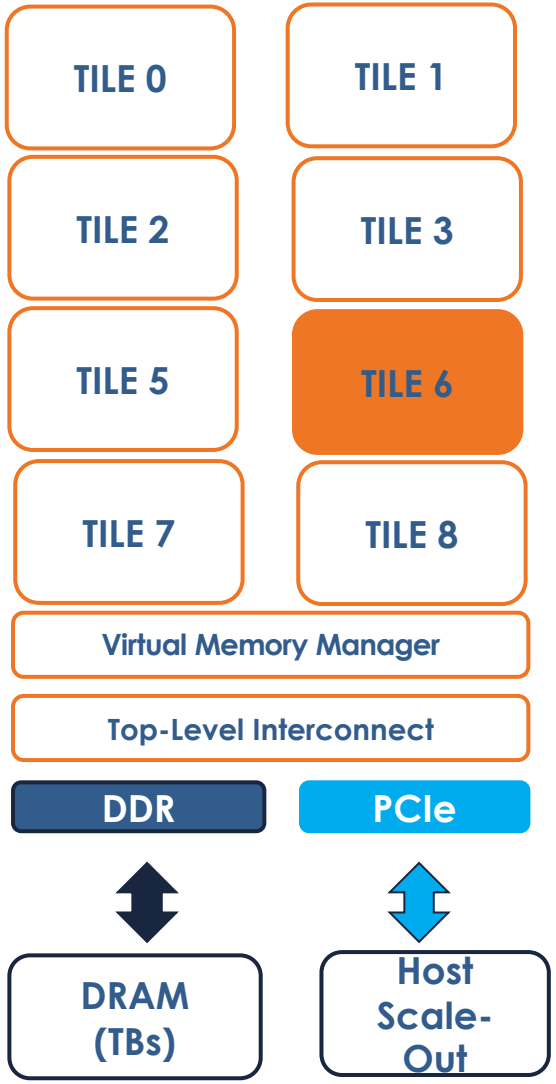


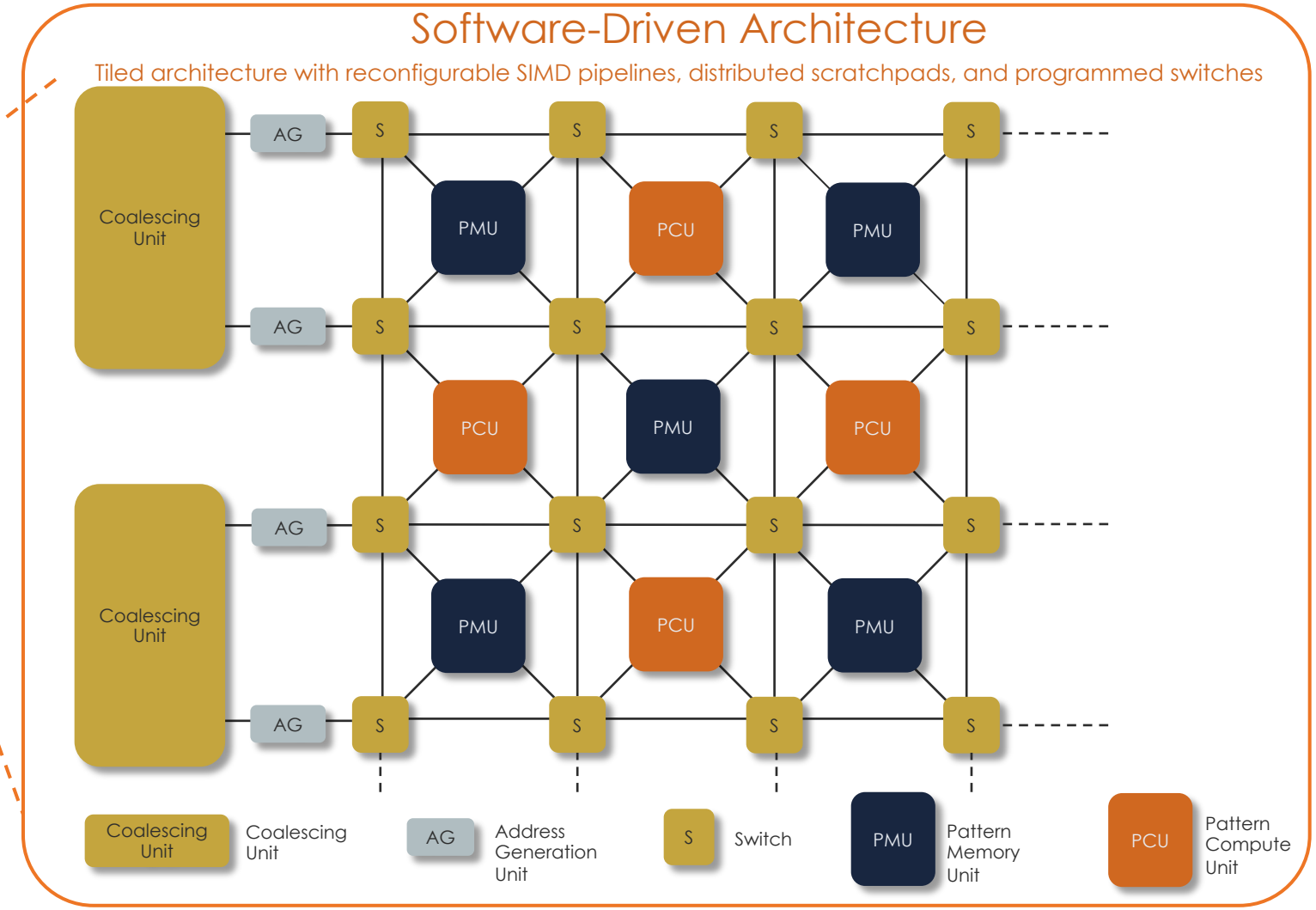
Mapping of a Typical Training Loop

10000ft view



Cardinal SN30: Reconfigurable Dataflow Unit (RDU)





SambaNova Cardinal SN30 RDU

The world's most advanced Dataflow processor powered by SambaFlow

Cardinal SN30
Reconfigurable Dataflow Unit

- 86B Transistors
- 102 Km of wire
- 640 MB on-chip, 1,024 GB External
- 688 TFLOPS (bf16)
- 7nm TSMC

DataScale SN30

2-Node Configuration Details

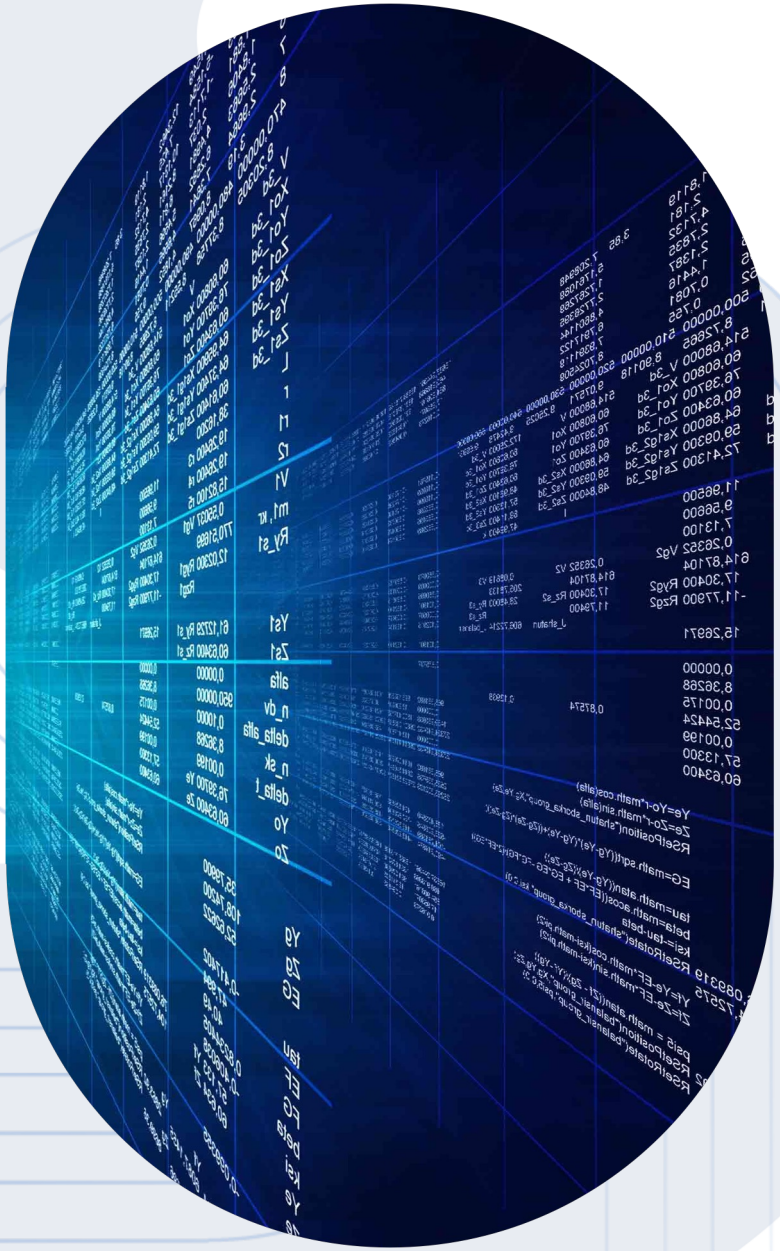
- 2 x SambaNova SN30-8 compute nodes
- 400 GbE high-performance data switch, with 200GbE device NICs, 32-port
- 1 GbE management switch, 48-port
- Serial console manager
- PDUs and networking cable included

- Each DataScale SN30-8 Node:
  - 8 x SN30 RDUs
  - 64 x 128 GB DDR4 DIMMs (8 TB total)

SambaNova SYSTEMS

Argonne National Laboratory

Murali Emadi



SambaNova SYSTEMS

Thank you

Martin Mueller, Customer Engineer

martin.mueller@sambanova.ai


SambaNova®
SYSTEMS