SambaNova SYSTEMS

SC23
Denver, CO | i am hpc.

Programming New AI Accelerators for Scientific Computing

Tutorial at Supercomputing 2023
12 November 2023, Denver, CO, USA

Petro Junior Milan
petro-junior.milan@sambanova.ai

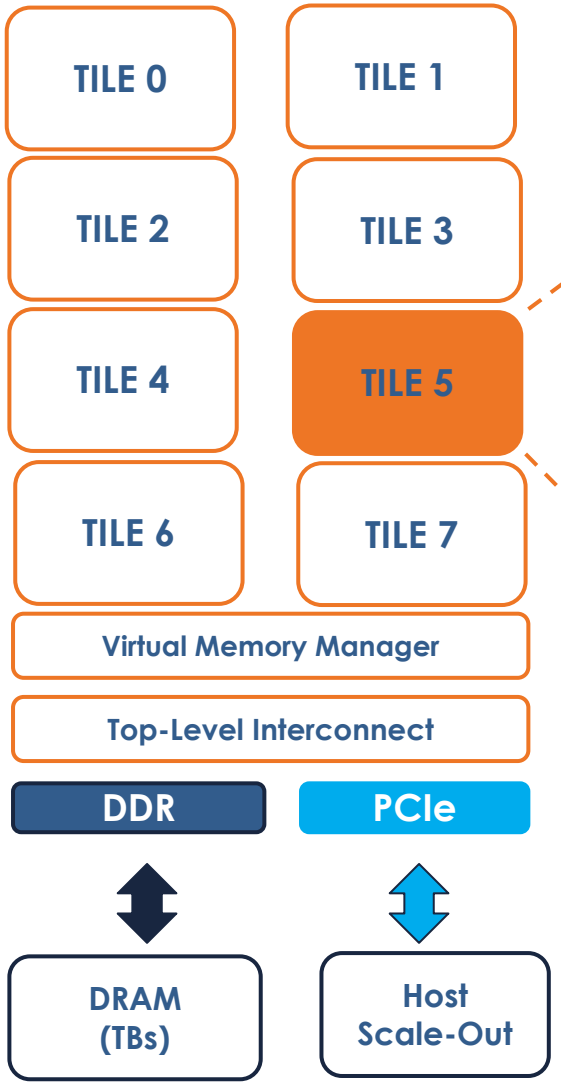
SambaNova Cardinal SN30 RDU

A reconfigurable dataflow processor

Cardinal SN30™
Reconfigurable Dataflow Unit™

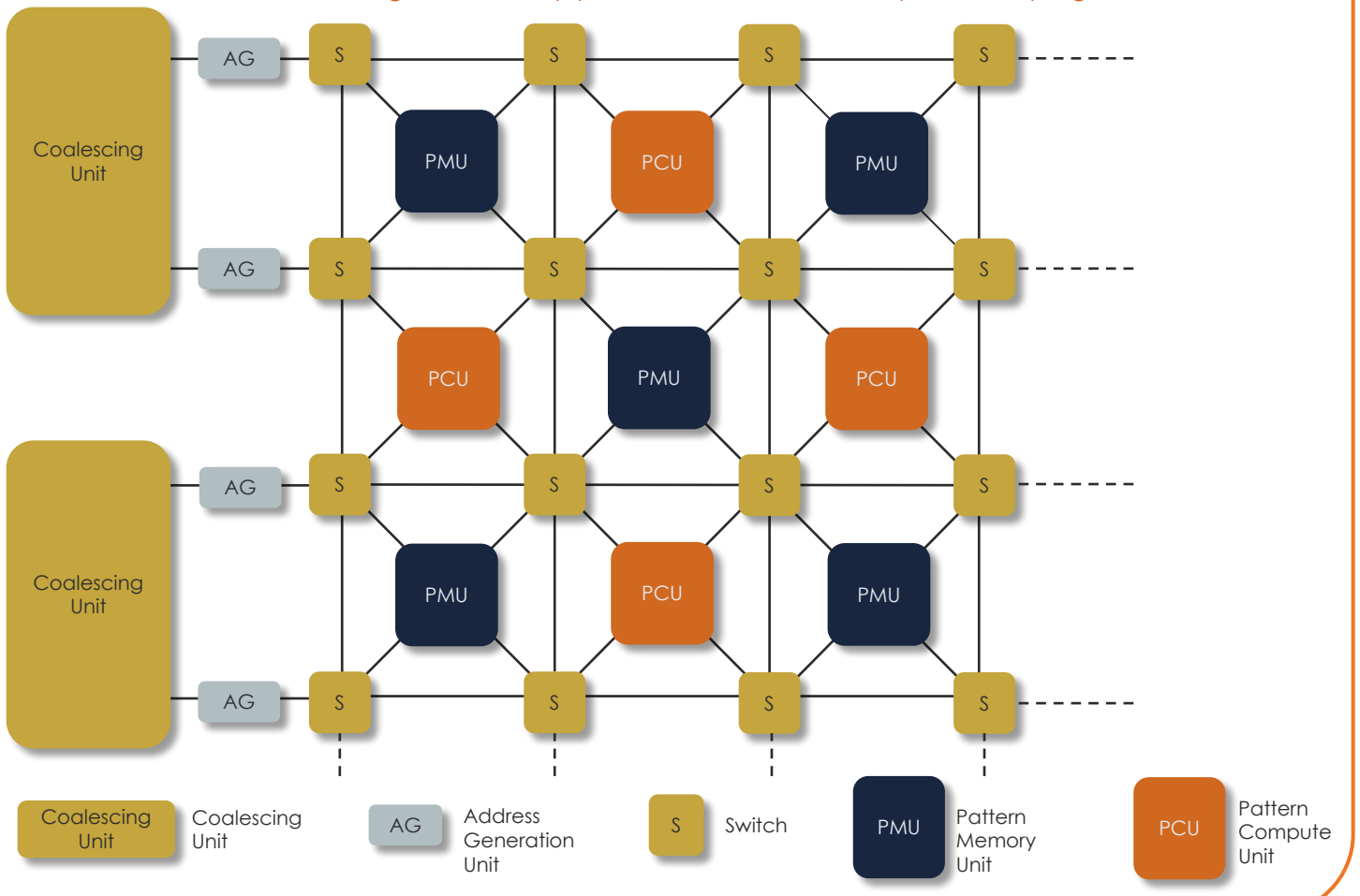
- 7nm TSMC, 86B transistors
- 102 km of wire
- 640 MB on-chip,
  1,024 GB external
- 688 TFLOPS (bf16)
- RDU-Connect™

Cardinal SN30: Chip Overview



Software-Driven Architecture

Tiled architecture with reconfigurable SIMD pipelines, distributed scratchpads, and programmed switches



SambaNova DataScale SN30

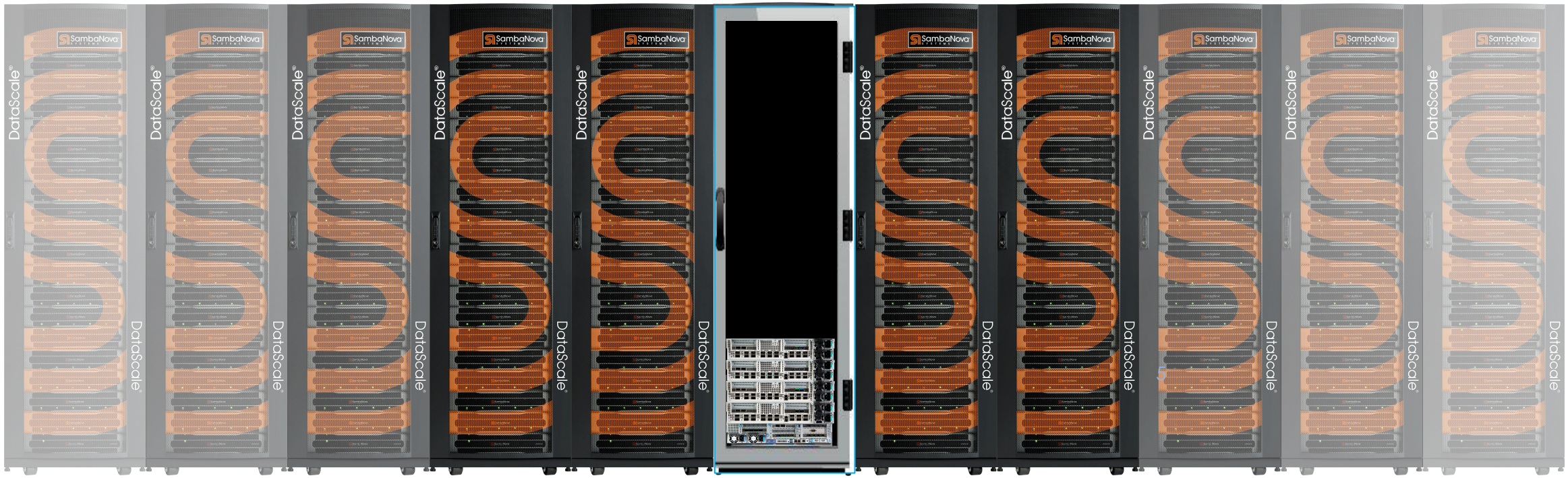
Scalable performance for training and inference

- **Rack optimized, integrated system**
  - 2x better performance over Gen 1
  - 2-node configuration
  - 400 GbE high-performance data switch, with 200 GbE device NICs

- **Each DataScale® SN30-8 node:**
  - 8 x Cardinal SN30™ RDUs
  - 8 TB DRAM
  - Host module

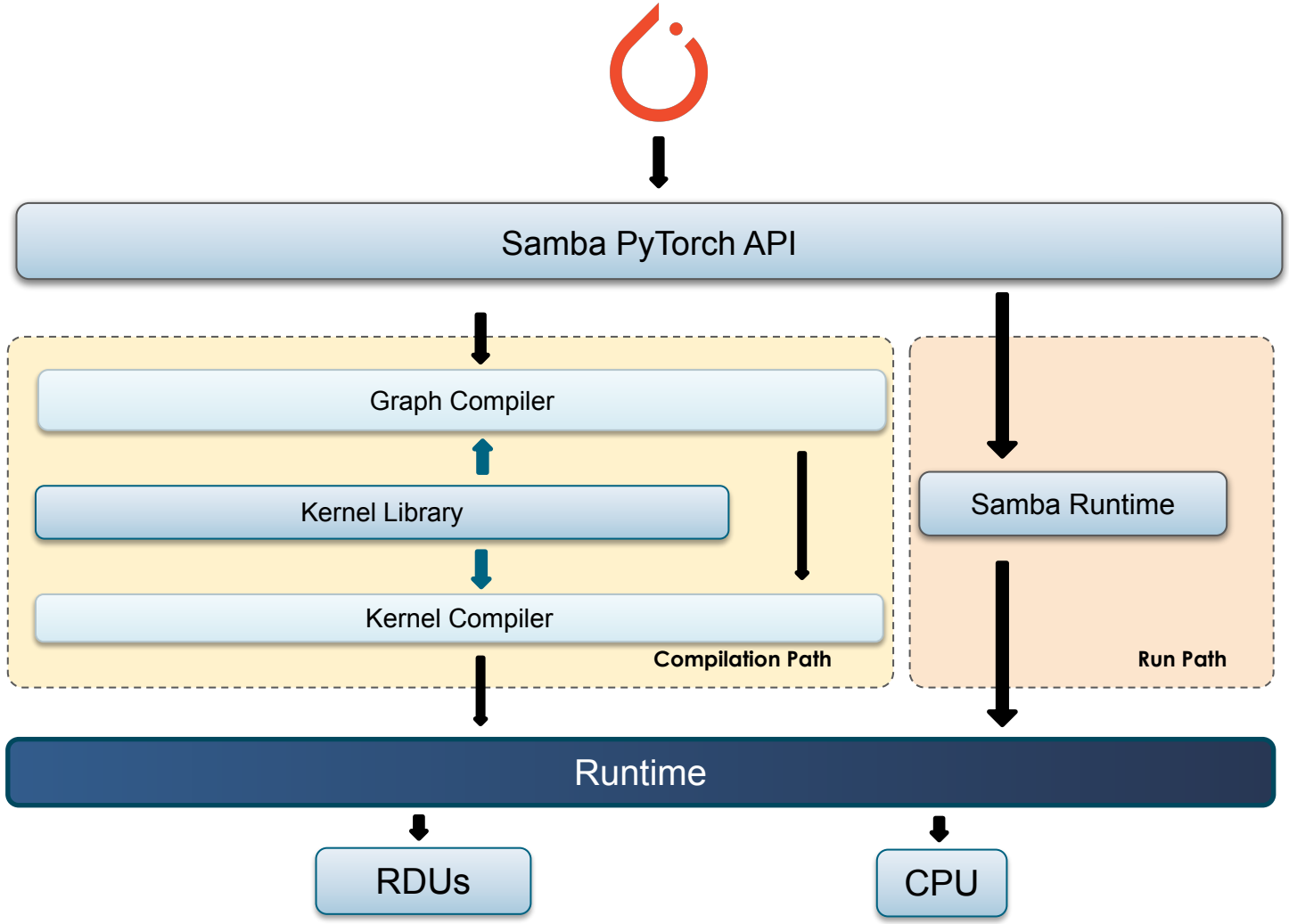
DataScale Systems Scale-Out

Scale to multiple racks with consistent rack-to-rack bandwidth and latency



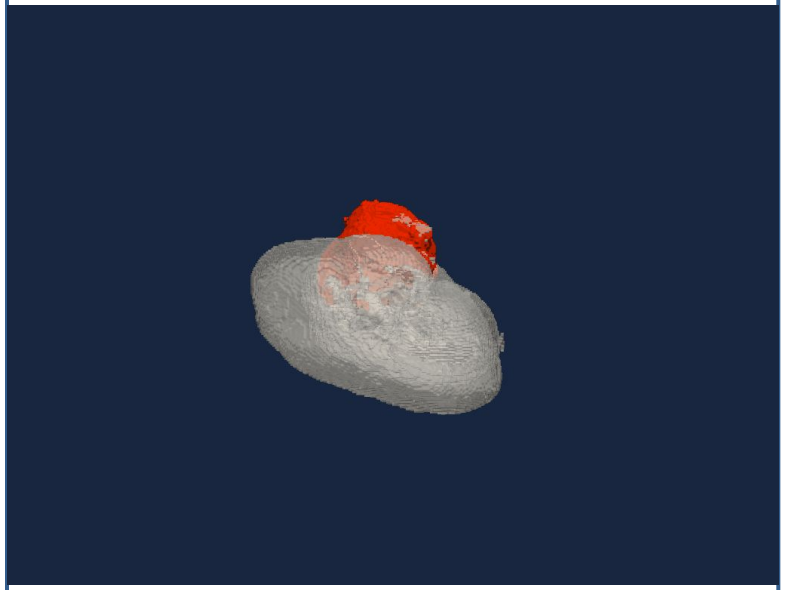
SambaFlow™ Software

- **Samba**
  - SambaNova PyTorch compilation & run APIs
- **Graph compiler**
  - High-level ML graph transformation & optimizations
- **Kernel compiler**
  - Low-level RDU operator kernel transformation & optimizations
- **Kernel library**
  - RDU operator implementations

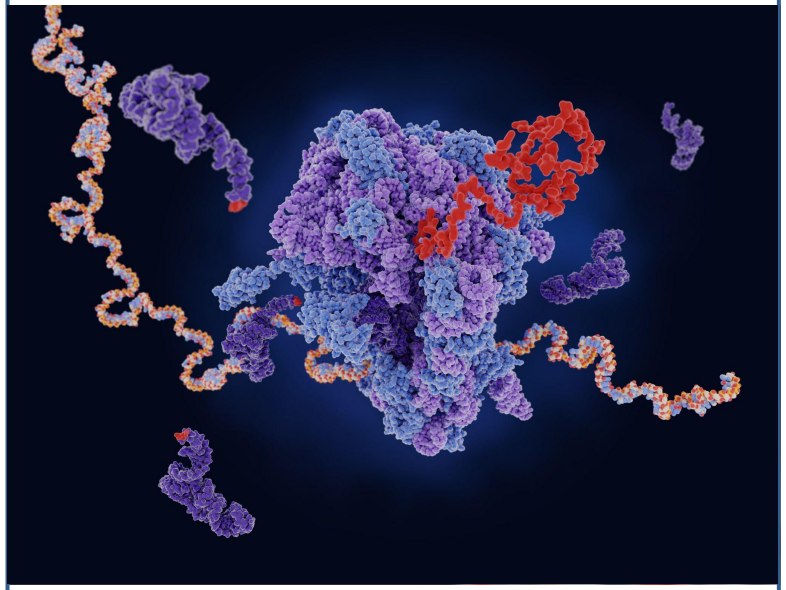


Powering the Next Wave of Discovery

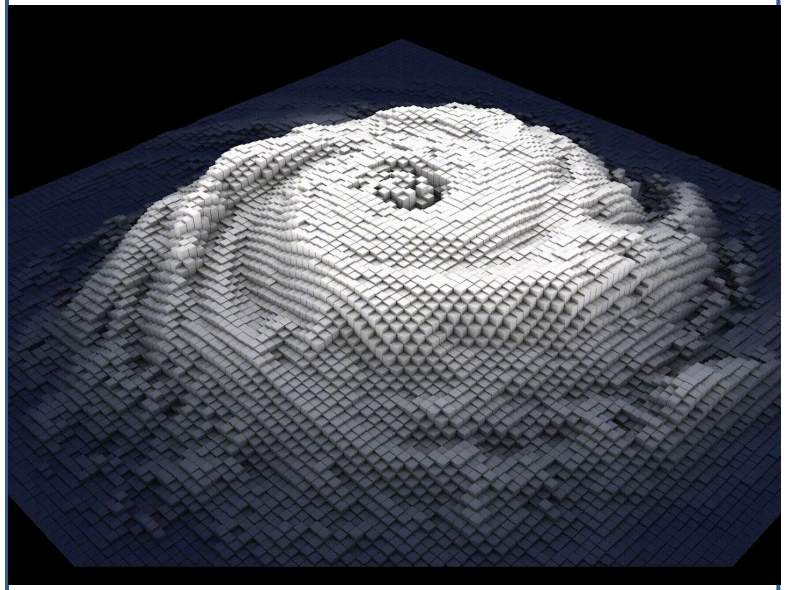
True resolution 3D imaging: 512^3 and beyond



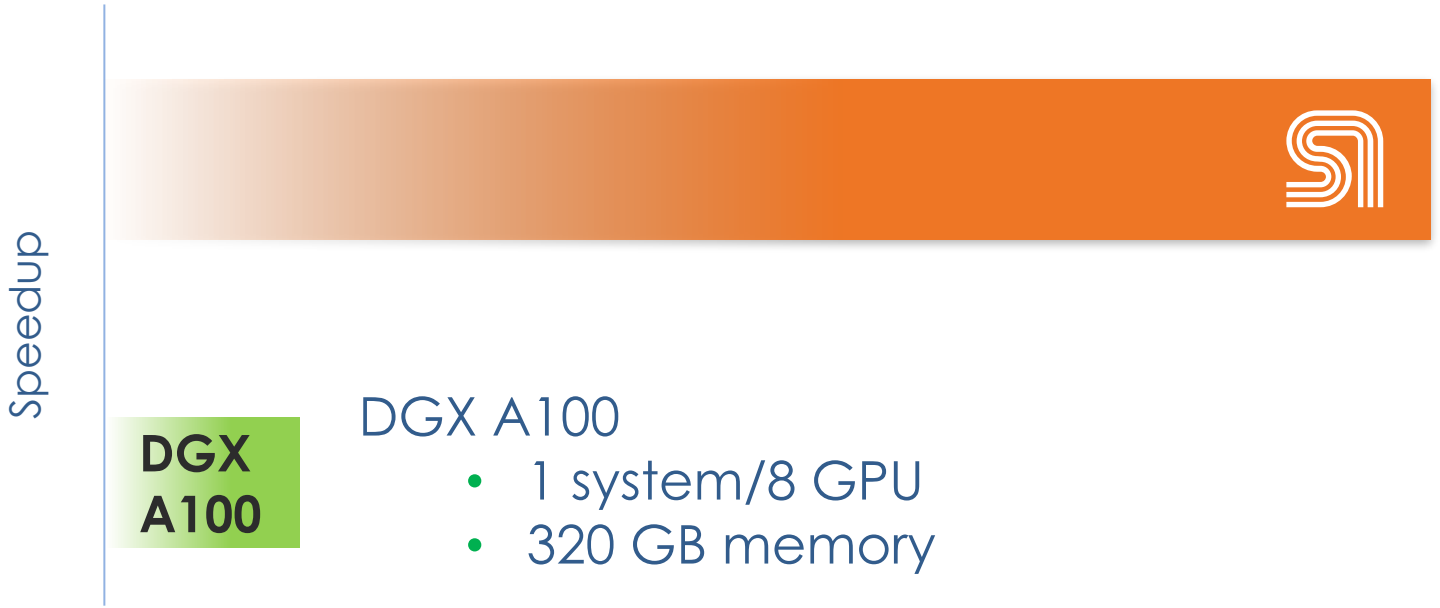
LLM and Transformers for Science



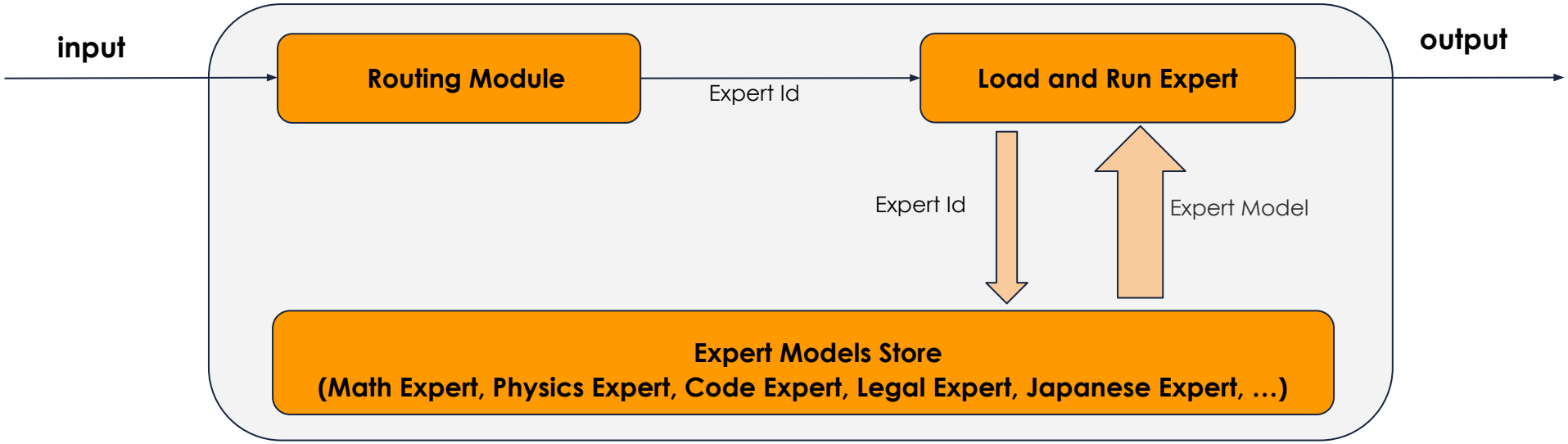
Surrogate and AI for Science Models



World Record Time-to-Accuracy on GPT 13B



Composition of Experts (CoE) with 1T Params – A New Way to Build Powerful LLMs



Simpler and faster to build:

- Classic ML based router
- Fine tuned small experts
- Leverage open source

Modular:

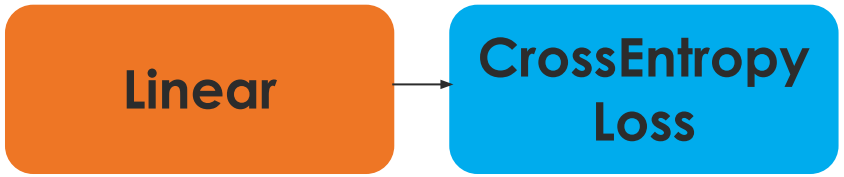
- Debuggable/Interpretability
- Improvements without regression

Suitable on RDU:

- Large memory
- Fast compute
- Dataflow

Main Steps to Integrate PyTorch with SambaFlow

Example on logistic regression




```
import samba

class Logreg(nn.Module):
    …
model = Logreg()

samba.from_torch_model_(model)

samba.session.compile(model, inputs, optimizer,
name='logreg_example')

samba.session.run(input_tensors=…, output_tensors=…,
                  hyperparam_dict=hyperparam_dict)
```

1. Define normal PyTorch model
2. Convert model parameters to RDU backend
3. Extract a static graph and compile
4. Run your graph

Steps to Compile and Run

Example on logistic regression

- **Method 1: using srun**

```
srun python logreg.py compile --pef-name="logreg" --output-folder="pef"
srun python logreg.py run --pef="pef/logreg/logreg.pef"
```

- **Method 2: using sbatch**
  - Create a bash script (submit-logreg-job.sh here as an example):

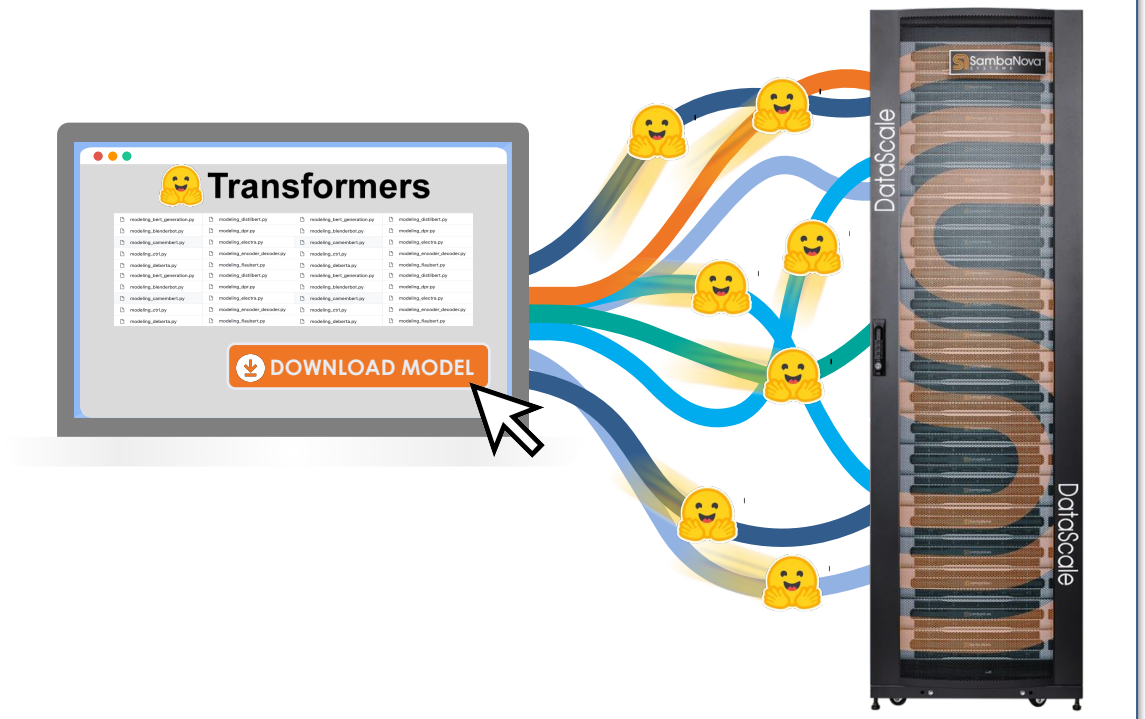
```
#!/bin/sh
source /opt/sambaflow/apps/starters/logreg/venv/bin/activate
python logreg.py compile --pef-name="logreg" --output-folder="pef"
python logreg.py run --pef="pef/logreg/logreg.pef"
```

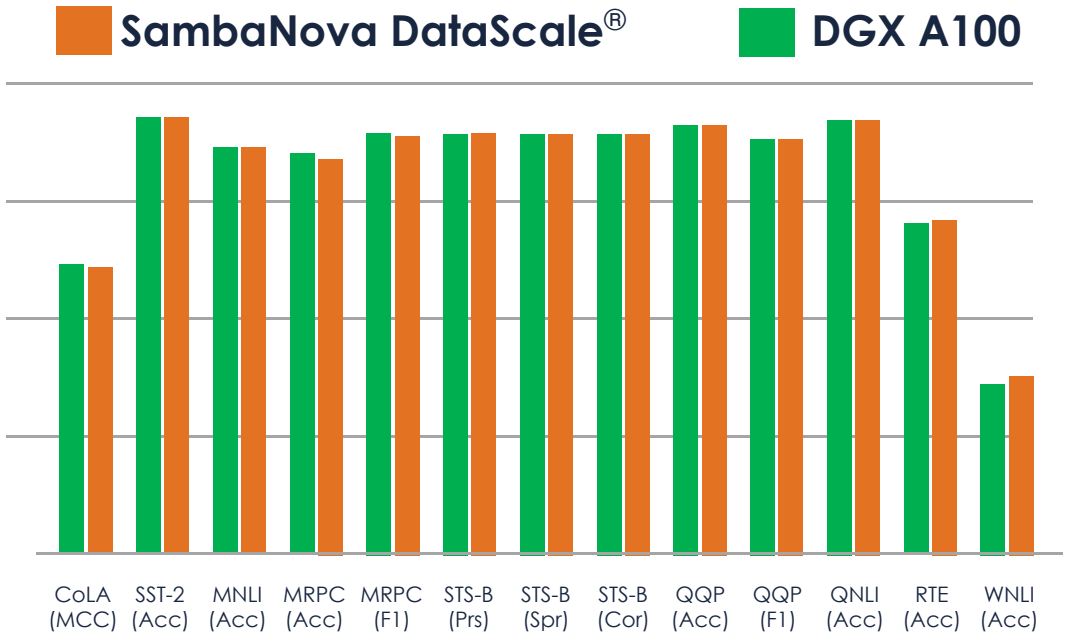
  - Then pass the bash script as an input to the `sbatch` command:

```
sbatch --output=output.log submit-logreg-job.sh
```

Run State of the Art Accuracy Transformers in Seconds

Instantly run many Hugging Face models with zero code change





Downloadable Models

Results in Seconds

Same Accuracy as on GPU

Learn More About Dataflow!

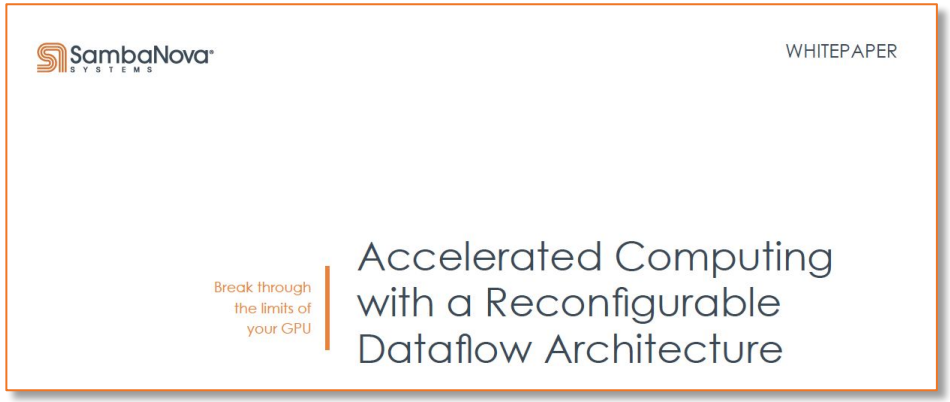
Connect with SambaNova's supercomputing experts at SC23
Booth #681

Visit the product pages

SambaNova Suite

DataScale

Download the RDA whitepaper



RDA whitepaper

Stay up to date on the latest

sambanova.ai

sambanova-systems

@SambaNovaAI

SambaNovaAI

SambaNova SYSTEMS

SambaNova SYSTEMS

Thank you

petro-junior.milan@sambanova.ai