SambaNova SYSTEMS

Accelerating Scientific Applications with SambaNova Reconfigurable Architecture

Vijay Tatkar
Director, SambaNova Systems

BoF at Supercomputing 2024
20 November 2024, Atlanta, GA

Agenda

1. SambaNova: Who We Are
2. Hardware Architecture and Compilation flow
3. Project Redwood: C++ SDK

SambaNova: Who We Are

SambaNova SYSTEMS

Who We Are

Snapshot

- Founded in 2017 by industry luminaries and originated at Stanford University
- Fully integrated generative AI platform, from 4th generation hardware to pre-trained models
- $1B+ funding raised

Founded by pioneers in AI

Lip-Bu Tan
Executive Chairman

Rodrigo Liang
Co-founder & CEO

Kunle Olukotun
Co-founder & Chief Technologist & Stanford Professor

Christopher Ré
Co-founder & Stanford Professor

Sophisticated, long-term investors

BlackRock Capital Investment Corporation™ · SoftBank Investment Advisers · TEMASEK · GV · intel Capital · SAMSUNG CATALYST FUND · GIC · Micron · SK telecom · WALDEN INTERNATIONAL

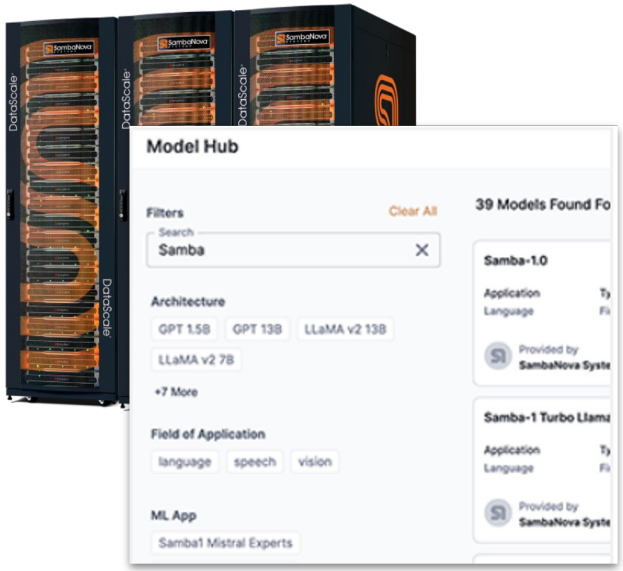
SambaNova SYSTEMS

SambaNova Products

SambaNova Cloud

API service for inference at record-breaking speeds (link)

SambaNova Suite



Secure, on-premises AI platform for training and inference (link)

SambaNova DataScale

Fully integrated hardware-software AI system (link)

SambaNova SYSTEMS

Hardware and Software Architecture

SambaNova SYSTEMS

SN40L: SambaNova's 4th Gen AI Chip

"Cerulean" Architecture-based Reconfigurable Dataflow Unit

5nm TSMC

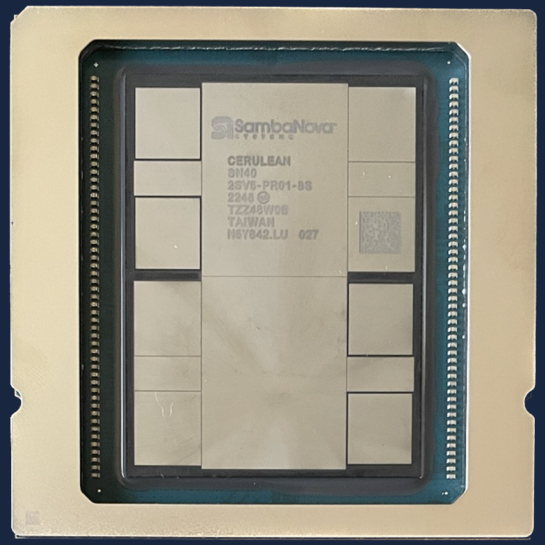
102B Transistors

1,040 RDU Cores

638 TFLOPS (bf16)

SambaNova SYSTEMS

Cerulean SN40L RDU



3-tier Dataflow Memory

520 MB
On-Chip Memory

64 GB
High Bandwidth Memory

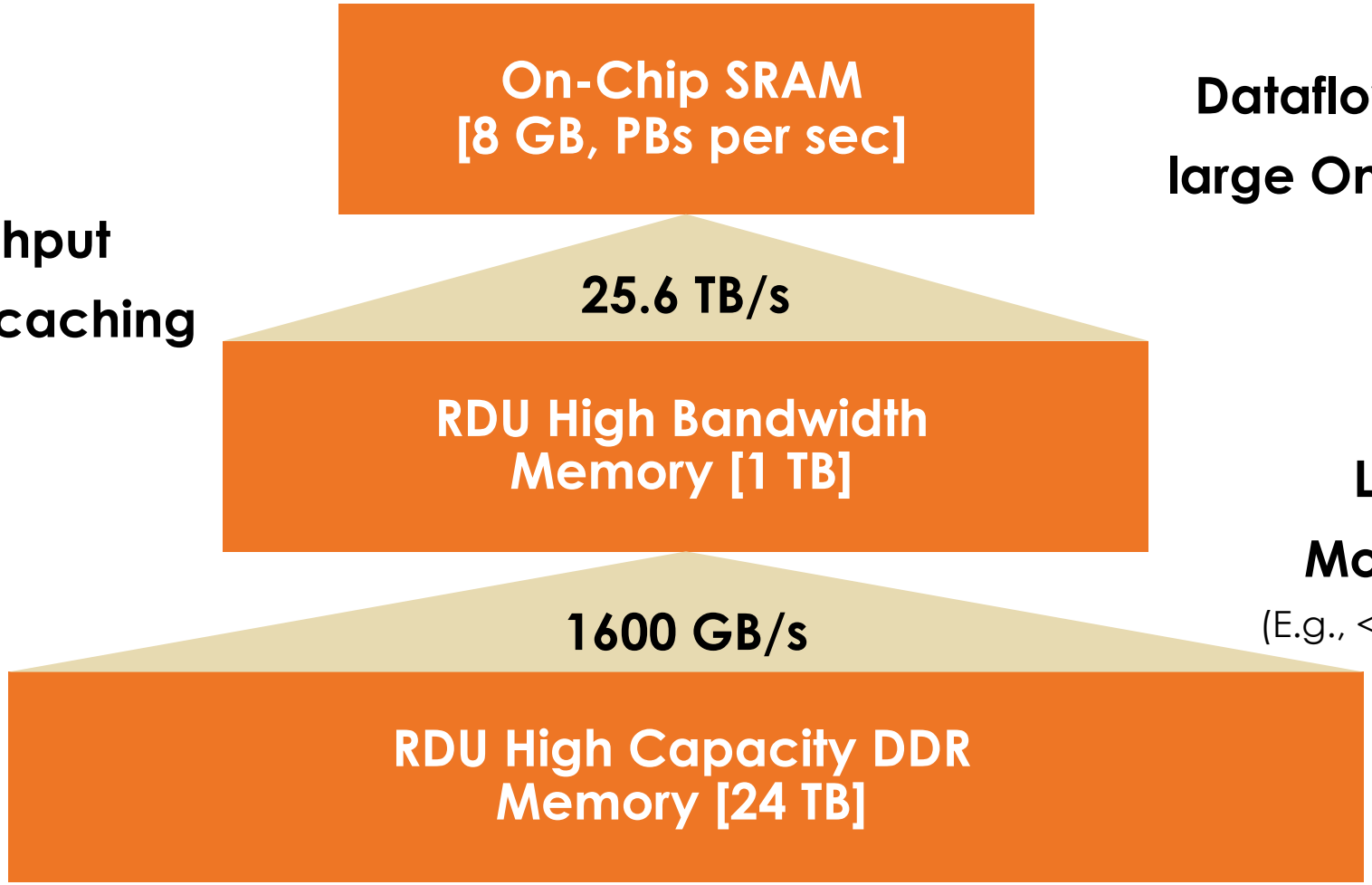
1.5 TB
High Capacity Memory

Generative AI Training and Inference

SambaNova SYSTEMS

SN40L: Three Tier Memory Architecture

3-tier Memory System with SRAM, HBM, and DDR



High throughput inference with caching

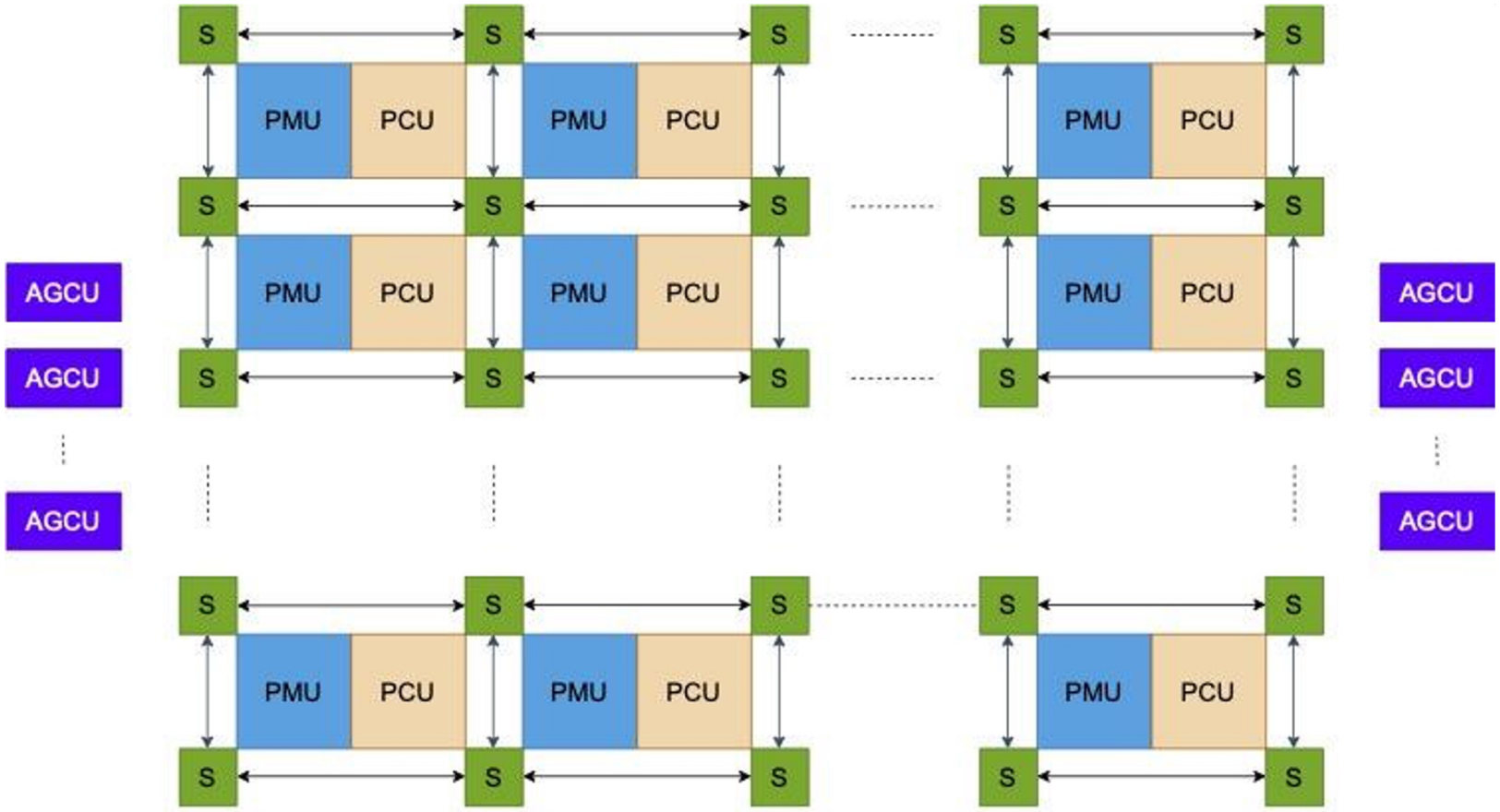
Dataflow enabled by large On-Chip Memory

Low Latency Model Switching

(E.g., < 0.01s for llama3.1 8B)

SambaNova SYSTEMS

SN40L: Tile Architecture



1040 PCUs and PMUs

PCU: Compute unit

PMU: Memory unit

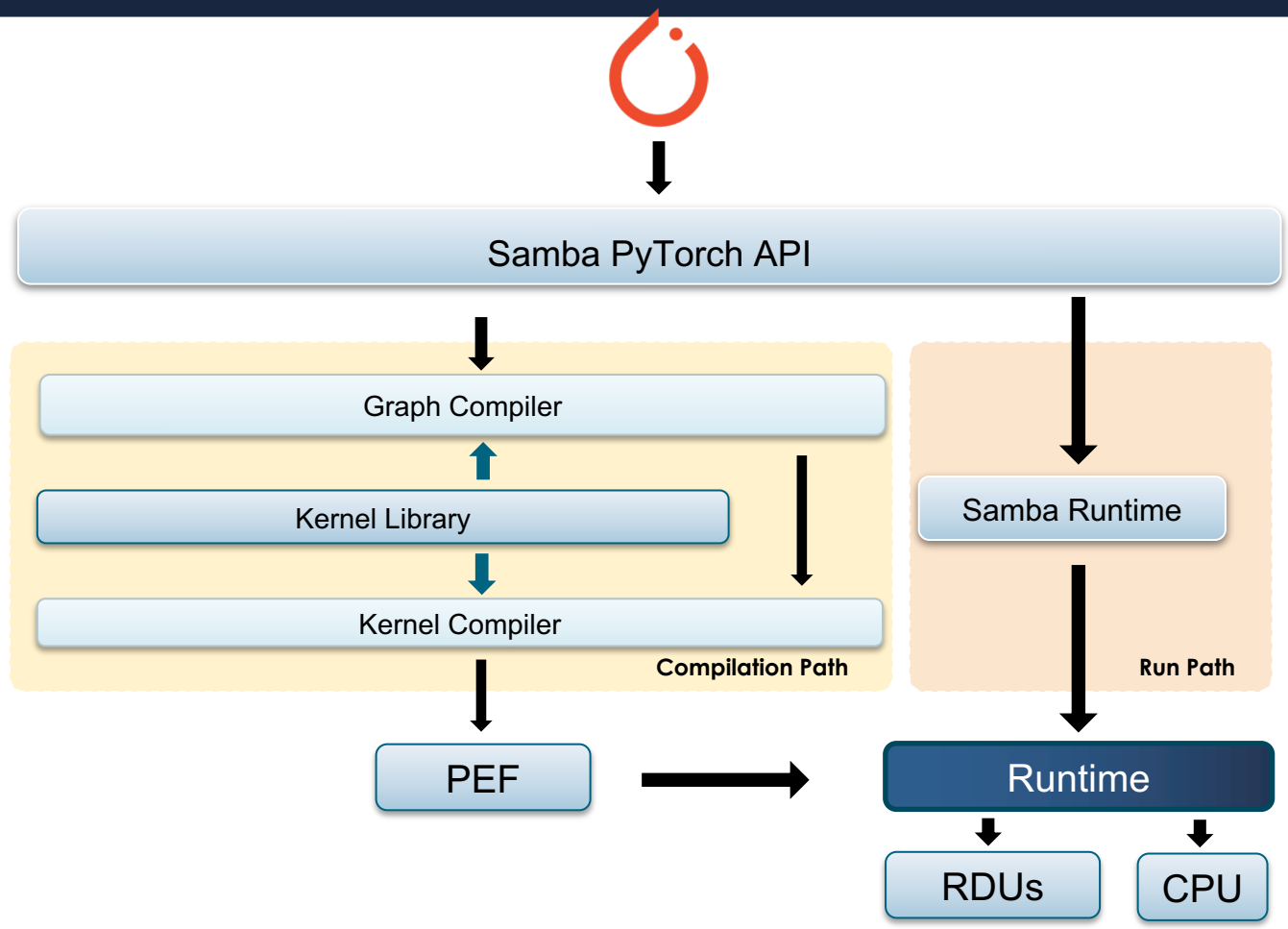
S: Mesh switches

AGCU: Portal to off-chip memory and IO

SambaNova SYSTEMS

Samba Compilation Flow

- **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



Project Redwood
C++ SDK

SambaNova SYSTEMS

Redwood: C++ SDK for the RDU

- What is Project Redwood:
  - Tensor-oriented kernel definition language and RDU scheduling SDK, embedded in standard C++
- Redwood lets users
  - + **Specify tensor functions** in standard C++
  - + **Compile them from** C++ API
  - + **Run them from** C++ API
  - + **Tune** them from C++, aided by SambaTune
  - + **Debug** them from C++ through emulation, watchpoints and alerts

Redwood: Design Objectives and Status

Design Objectives

- Enable expert developers to exploit the capabilities of RDUs
- For new innovation vs. porting
- Example use cases
  - + Convert compute heavy inner loops of existing C++ programs as tensor for RDU offload
  - + Develop high-performance ML operators

Status

- Ramping internal use
- Early preview with select customers
- Feedback collection will inform design choices for needed kernels
- Public release coming soon

Redwood: Goals and Programming Model

Goals for Redwood library

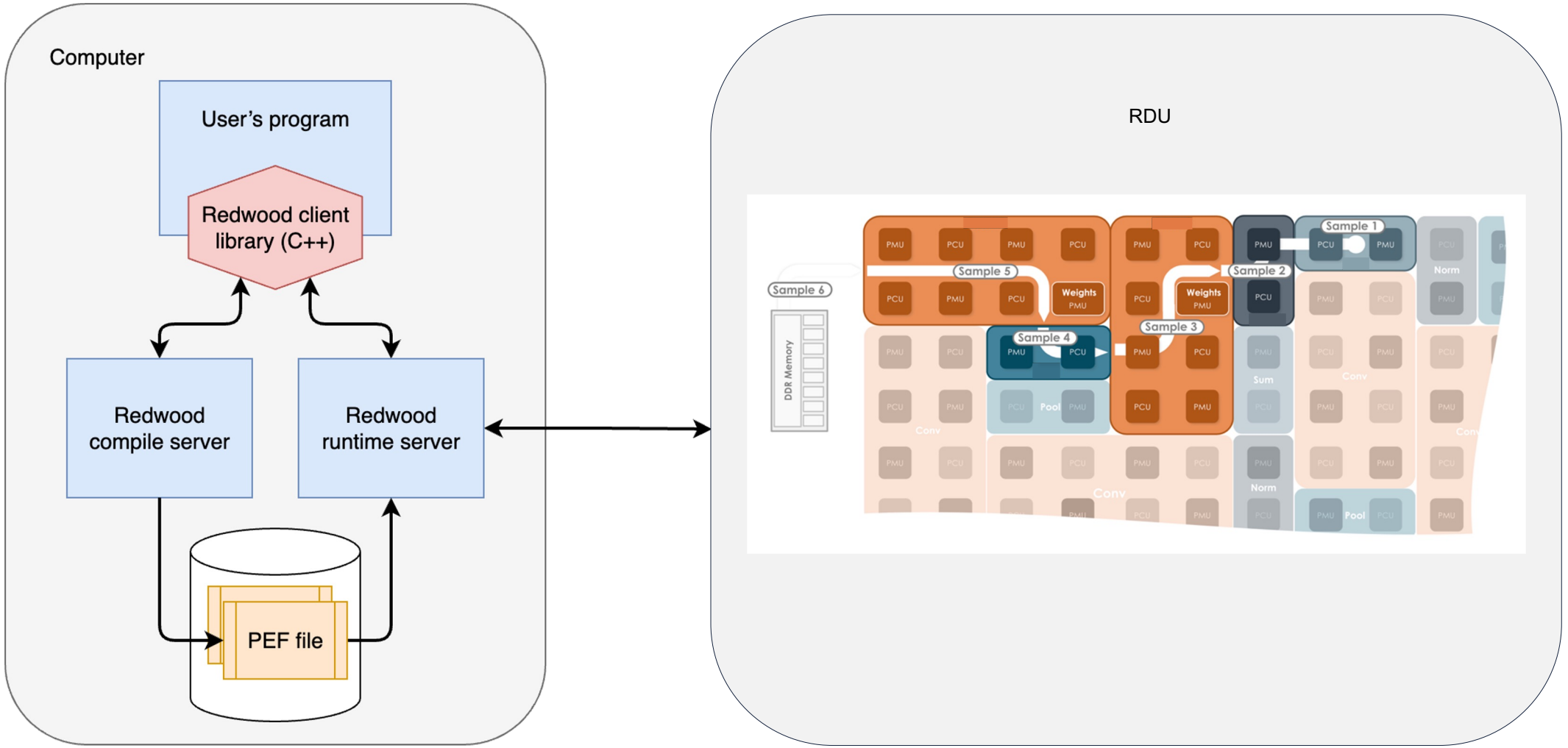
- Simplicity of **numpy** arrays
- Predictable performance characteristics of Fortran
- Leverage compiler to exploit parallel patterns (map, filter, reduce) to specify loop-like constructs
- Optimization directives (fusion, tiling, parallelization) for power users
- Composability, reusability and zero-cost abstractions

Redwood Programming Model

- Redwood **tensor** is "abstract". SDK executes kernels symbolically
- Redwood **array** is concrete and used for data motion
- Tensors expose their statically known shapes; allows for implementation choice based on extent of dimension
- Any function that manipulates **redw::Tensor** can be a kernel; function calls have no overhead in binary

SambaNova SYSTEMS

Redwood: System Components



SambaNova Cloud

Putting large scale applications together

	SambaNova	GPU
Llama 3.2 1B 16-bit	2477	304
Llama 3.1 8B 16-bit	1066	93
Llama 3.1 70B 16-bit	460	32
Llama 3.1 405B 16-bit	200	14

THANK YOU!

vijay.tatkar@sambanova.ai

SambaNova SYSTEMS

Join us
Booth #2309
sambanova.ai/sc24

Meet the Experts and Happy Hours at Booth #2309

Sunday, November 17

- 11:30 a.m. -12:00 noon
  Tutorial: GenAI Training and Inference at Scale

Tuesday, November 19

- 1:00 p.m. - 3:30 p.m.
  SambaNova Customer Experts

- 4:00 p.m. - 5:00 p.m.
  SambaNova Experts Happy Hour

- 5:00 p.m. - 6:00 p.m.
  SambaNova Partner Experts Happy Hour

Wednesday, November 20

- 1:00 p.m. - 2:30 p.m.
  Meet SambaNova Customer Experts

Birds of a Feather

Wednesday, November 20

- 5:15 p.m. - 6:45 p.m.
  Democratizing AI Accelerators for HPC Applications: Challenges, Success, and Support

- 5:15 p.m. - 6:45 p.m.
  The National Artificial Intelligence Research Resource (NAIRR) Pilot User Experience BoF

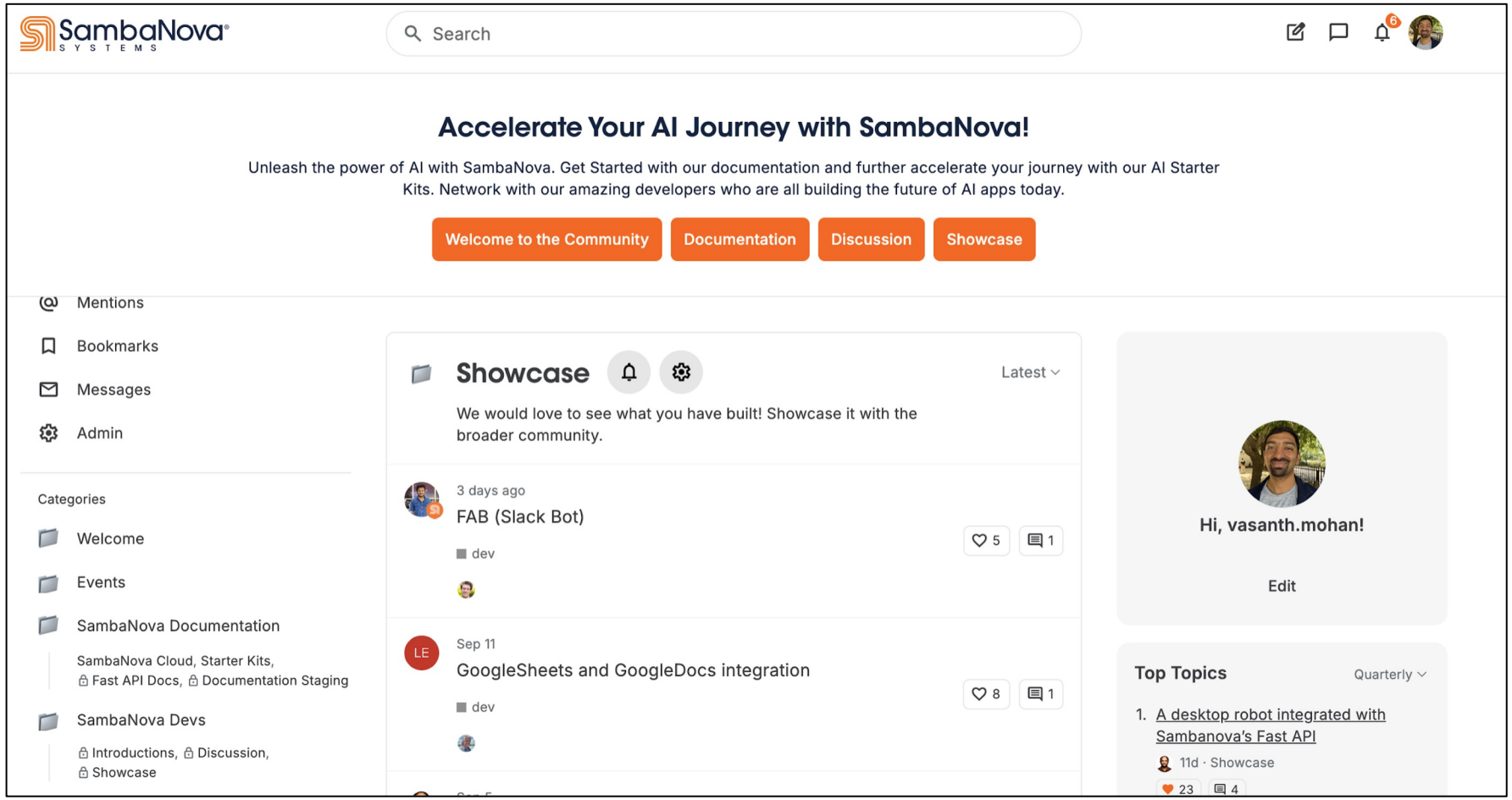
SambaNova SYSTEMS

Try It Today

cloud.sambanova.ai

SambaNova SYSTEMS

Questions? Join the Community



Community.SambaNova.ai