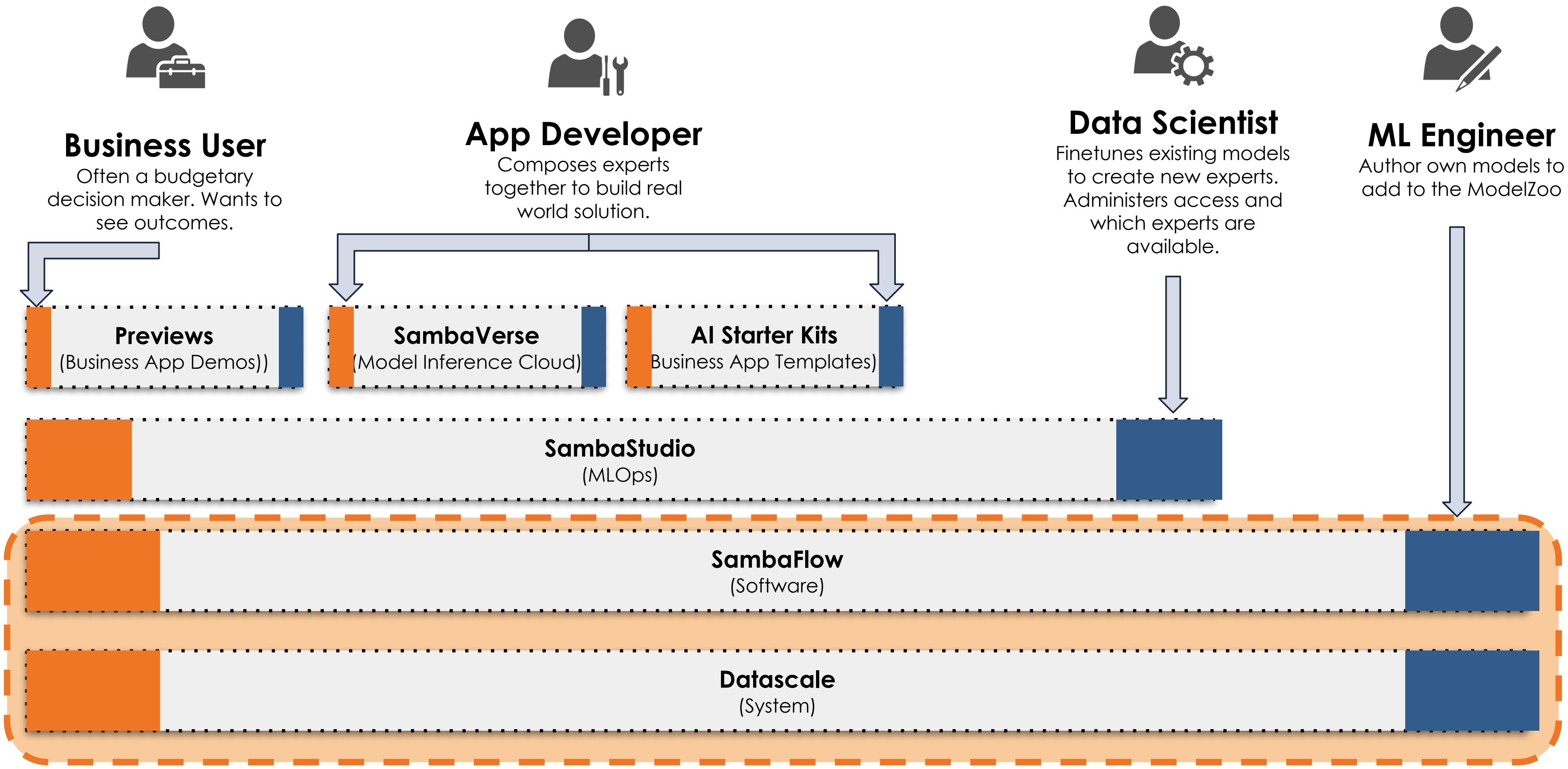
SambaNova® SYSTEMS

DataScale: Software Overview

May 2024

SambaNova Software Stack



Business User
Often a budgetary decision maker. Wants to see outcomes.

App Developer
Composes experts together to build real world solution.

Data Scientist
Finetunes existing models to create new experts. Administers access and which experts are available.

ML Engineer
Author own models to add to the ModelZoo

Previews
(Business App Demos))

SambaVerse
(Model Inference Cloud)

AI Starter Kits
Business App Templates)

SambaStudio
(MLOps)

SambaFlow
(Software)

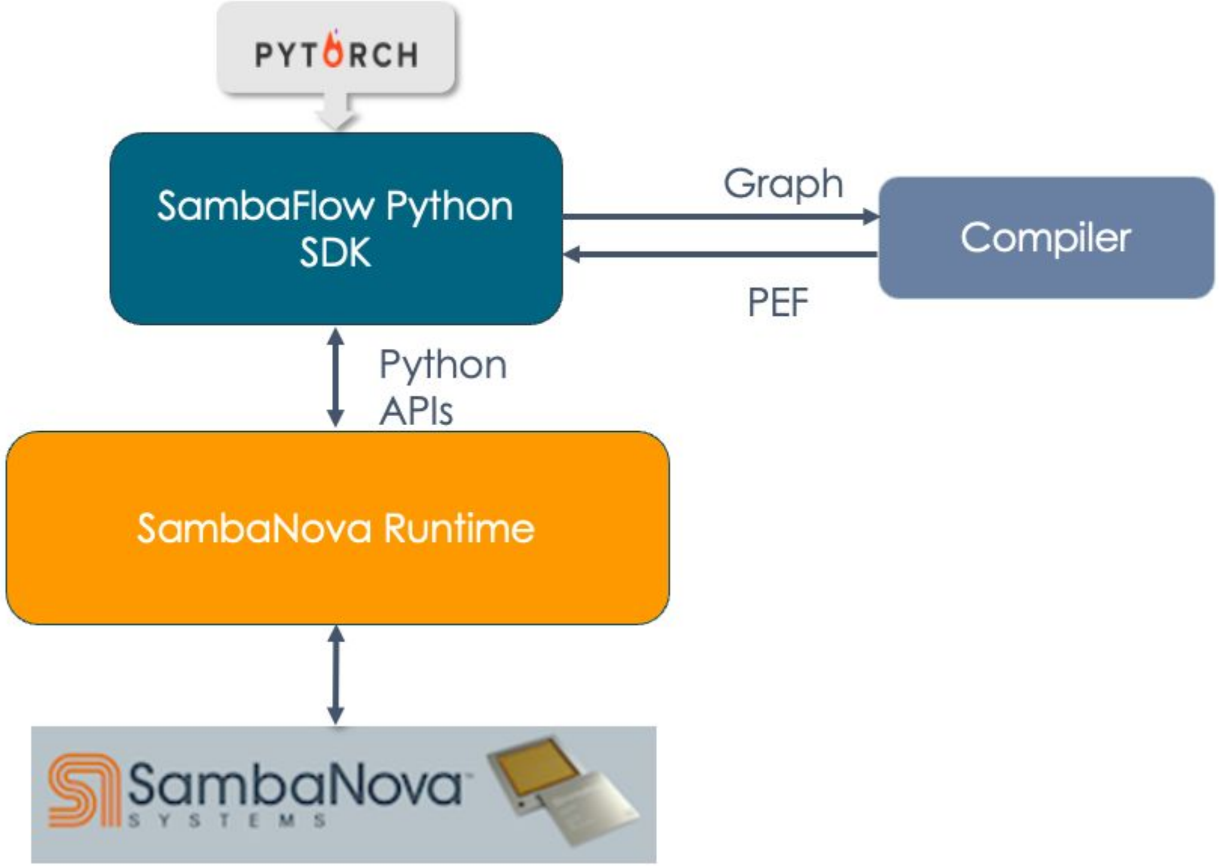
Datascale
(System)

🟧 Free,hosted multi-tenant access

🟦 Paid, dedicated, single tenant access

SambaFlow

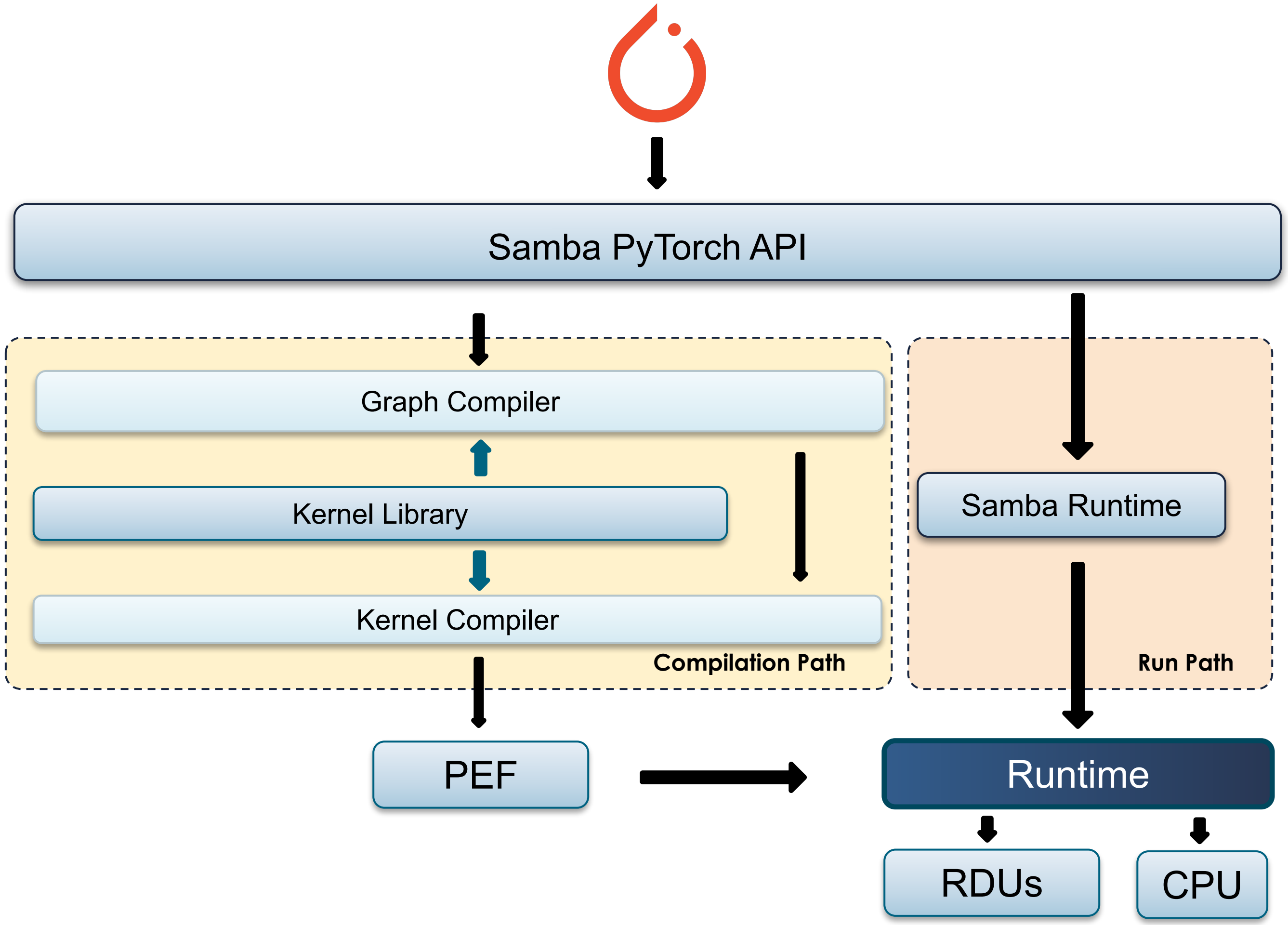
- Supports standard ML frameworks such as Pytorch
- Automatically extracts, optimizes and maps dataflow graphs onto RDUs
  - Achieve high performance without the need for low-level kernel tuning
- A consistent programming model for scaling from 1-RDU to multi system configurations
- Key components:
  - A Python interface to compile & run models
  - Compiler, intakes a Pytorch graph and outputs a PEF
  - Runtime, custom OS for RDUs



SambaFlow Compiler

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



Compiler Modes

O0 Operator Mode

- Initial bring up and model testing
- Each operator is run as a separate function
- Some optimizations applied

O1 Module Mode

- Fuse operators into modules for optimization
- Fusion rules defined in YAML files, heuristics automatically applied
- Reusability

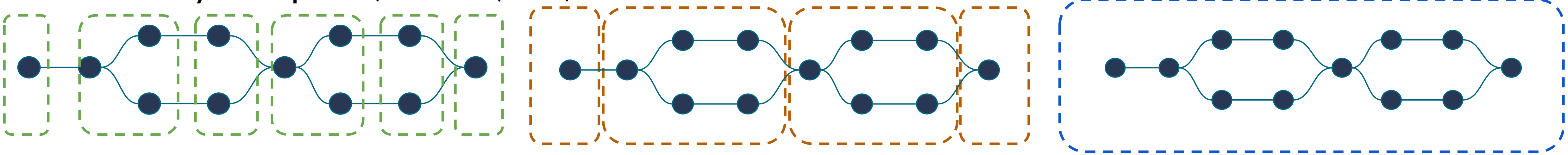
O1HD

- User directed heuristic optimization

O3 Full Graph Mode

- Fuse and optimize across entire graph
- Configuration specific
- HD files provide expert tuning
- Limited reusability

Each node is a PyTorch operator, i.e GEMM, ReLU, etc.

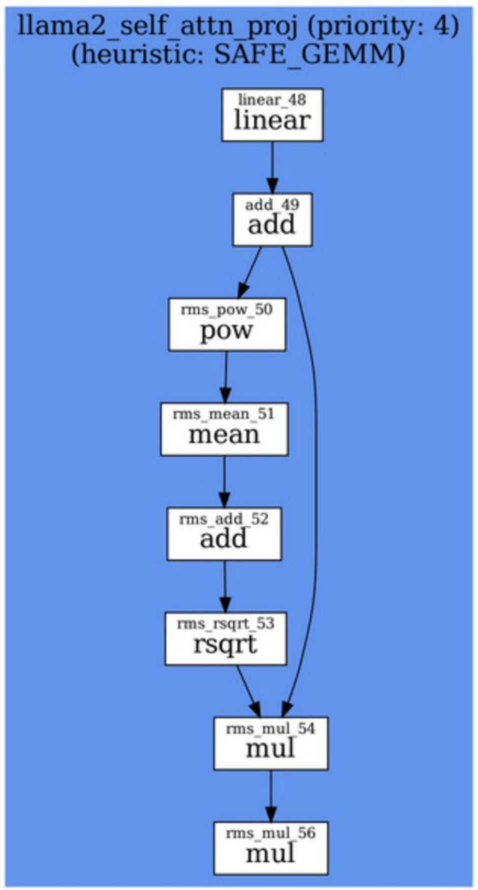


01 Operator Fusions

- Patterns of operators to fuse into a dataflow
  - + Users can also define their own patterns in yaml, or define directly in the app

- Each pattern can also specify a “heuristic”
  - + A heuristic is a specific strategy for optimization, put together as a package deal
    - ○ e.g. sharding, tiling, & section cuts
  - + Heuristics are flexible, being applicable to any pattern that meets its requirements

```yaml
llama2_self_attn_proj:
    priority: 4
    heuristic: SAFE_GEMM
    pattern:
        linear_48:
            op_type: linear
            child: add_49
            set m_shard_degree: 4
            set k_shard_degree: 2
        add_49:
            op_type: add
            children:
                - rms_pow_50
                - rms_mul_54
        rms_pow_50:
            op_type: pow
            child: rms_mean_51
        rms_mean_51:
            op_type: mean
            child: rms_add_52
        rms_add_52:
            op_type: add
            child: rms_rsqrt_53
        rms_rsqrt_53:
            op_type: rsqrt
            child: rms_mul_54
        rms_mul_54:
            op_type: mul
            child: rms_mul_56
        rms_mul_56:
            op_type: mul
```



Heuristics

- Each heuristic defines a different compiler optimization strategy
  - Different heuristics are different optimization strategies in deciding tiling/sharding/par-factors/section-cuts
- Three main heuristics, with more variations planned
  - Default O3 heuristics
  - GEMM-dominated Heuristic
  - MHA Heuristic
- Heuristics are plug-n-play: users can control which op-fusion pattern uses what heuristics

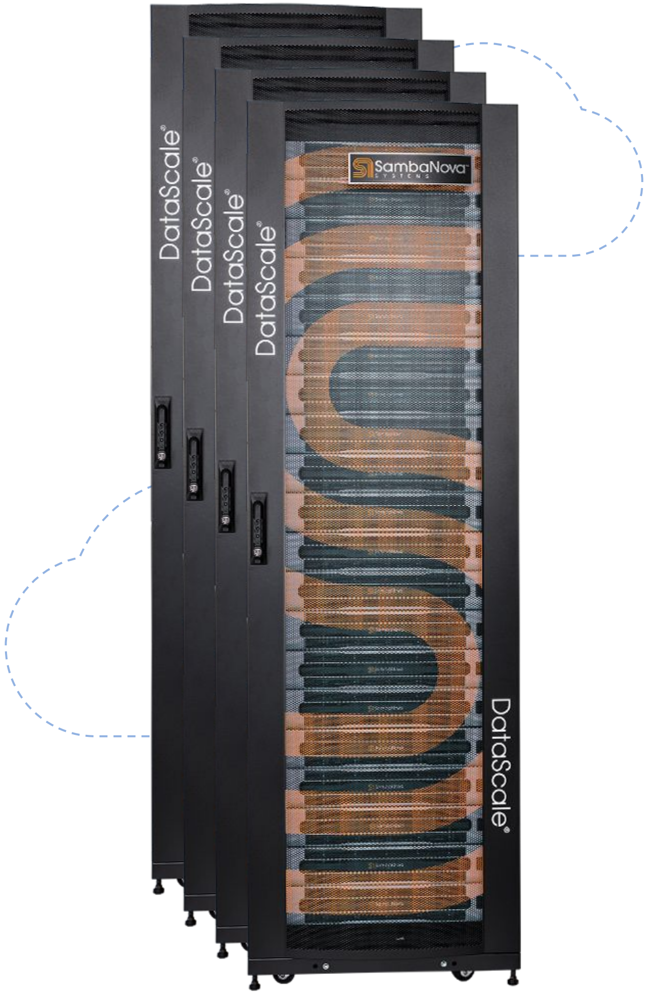
SambaNova SYSTEMS

SambaFlow Runtime

SambaNova SYSTEMS

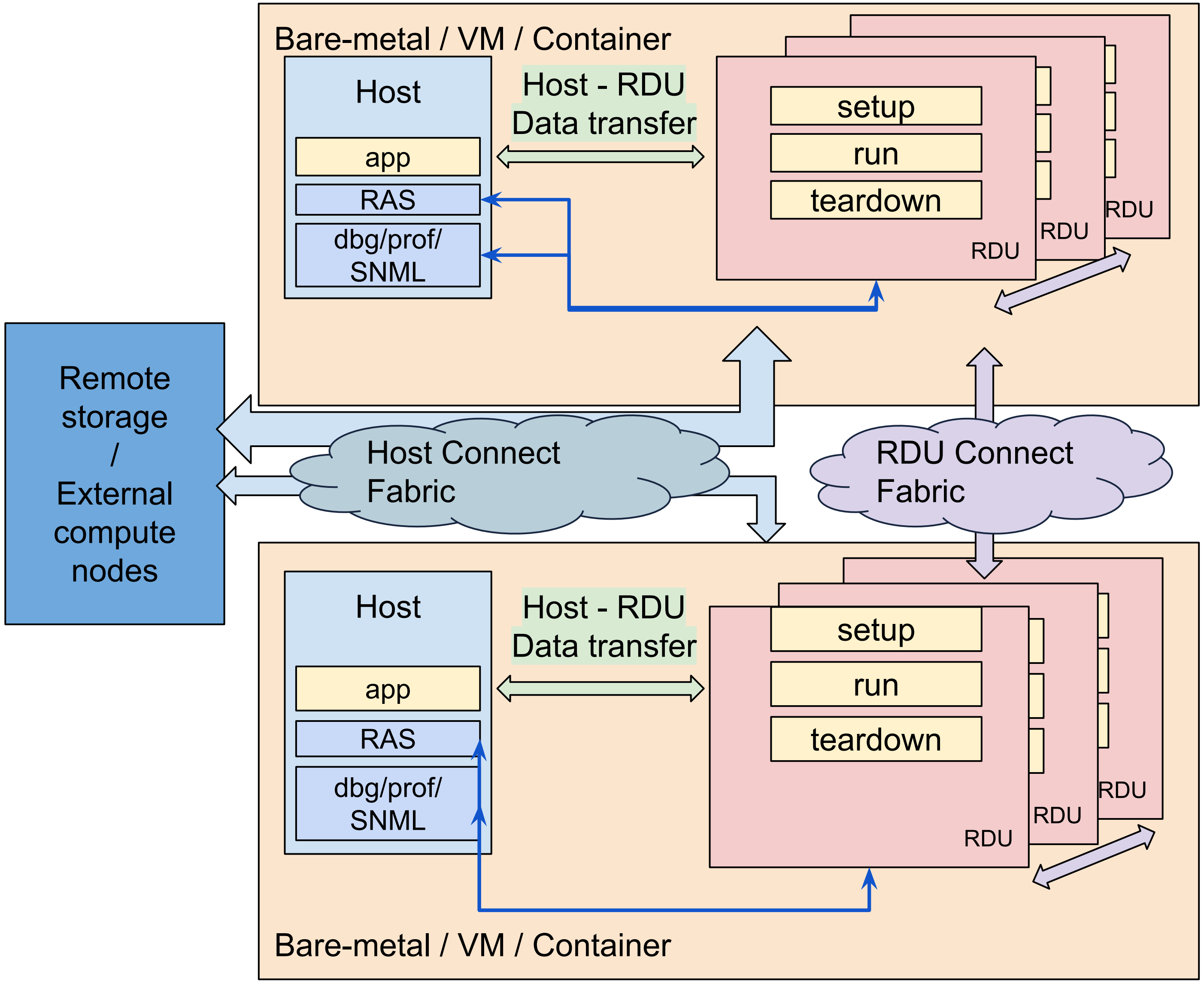
Overview

- Scalable high-performance runtime stack for SambaNova dataflow distributed systems.
- Operates as an **operating system** for RDUs
  - Manages AI compute, memory, I/O including PCIe and networking
  - Manages application/graph setup, scheduling, execution and tear-down
- Multi-OS support : Ubuntu 20.04.3 LTS, RedHat 8.5
- Minor-version backward compatibility for all Runtime interfaces



Core features of Runtime

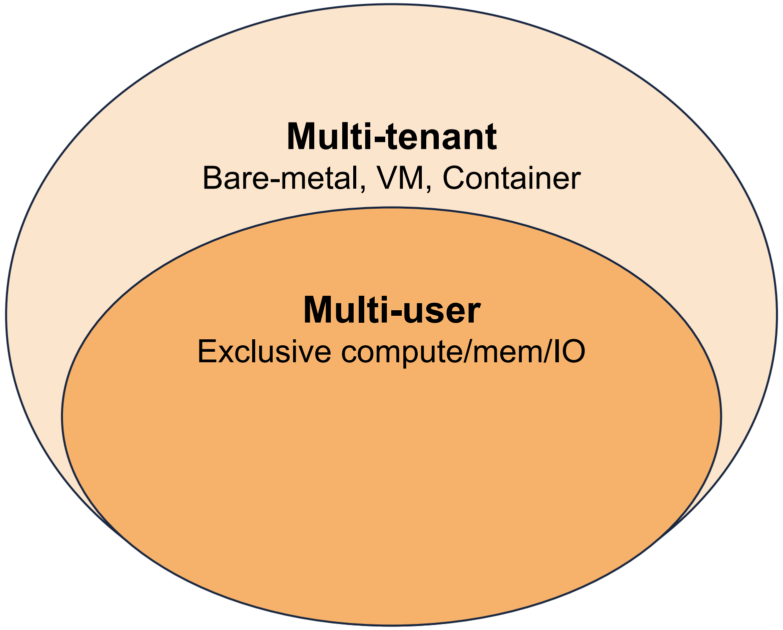
- Model parallel within a node
- Data-Parallel within and across nodes over RDUConnect (Inter-RDU) networking fabric
- Reliability, Availability, Serviceability (RAS)
- Support for external compute nodes and remote storage via host network fabric
- Debugger, performance & system management tool chain
- Language agnostic system management layer (SNML) interface for customers



SambaNova SYSTEMS

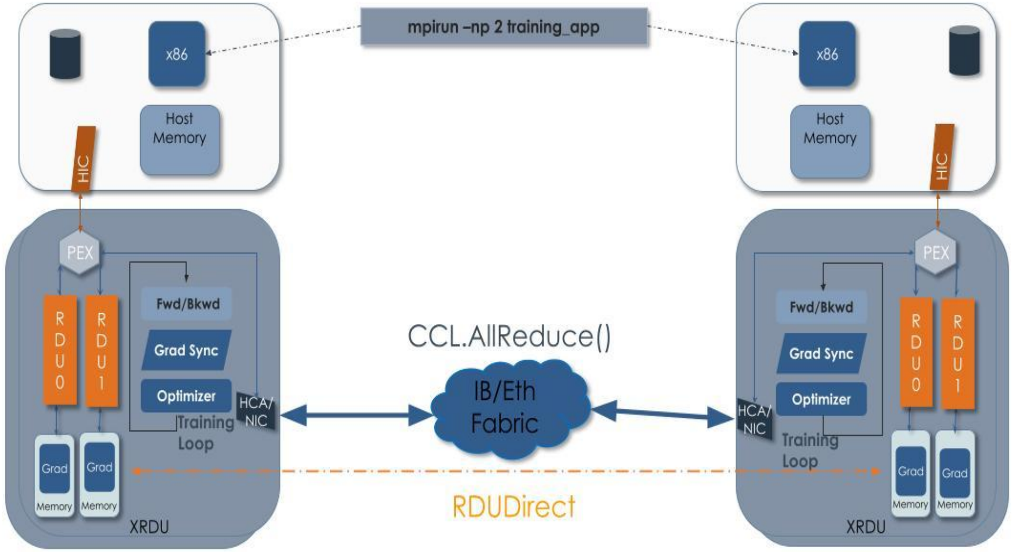
Multi-user and Multi-tenancy

- Multi-Tenant support
  - OCI-compatible Container support
- Multi-User support
  - Support upto 8 applications simultaneously on a node
  - Mutually exclusive compute, memory and IO resources between applications



Distributed Data Parallel Training

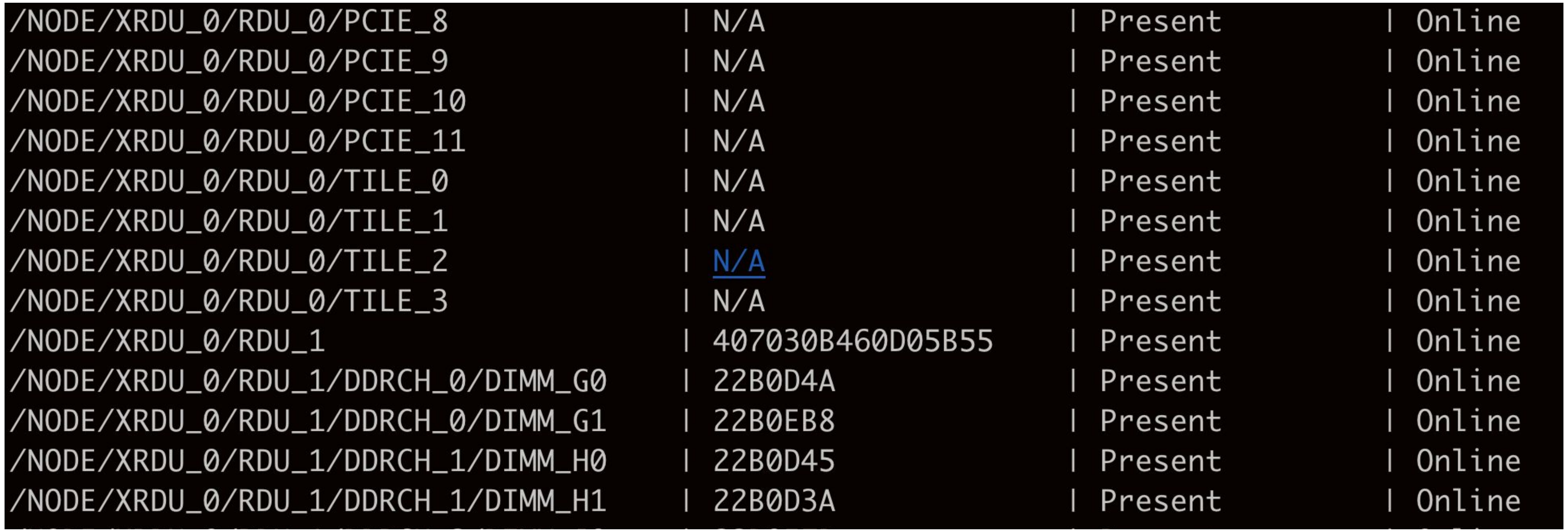
- Distributed training through data parallel
  - Across RDUs, nodes and racks
  - Support > 1k RDUs over RDMA transport
- Algorithm-Topology library
  - Multi bi-directional ring, All-to-All, Hierarchical allreduce



- Optimized Dataplane using Collective Communication Library (CCL) functions
  - Achieve high bandwidth over multiple IO fabrics
- Support primitives such as allreduce, allgather, send, recv
  - Support mixed precision (FP32/BF16) reduce, gradient grouping & sync overlap

System Reliability, Availability & Serviceability

- Hardware fault/error management
  - + Database-based hardware fault/error management
    - ○ Provide records of error events, faulty hardware and recovery suggestions
  - + Provide a tool interface for the fault/error management
    - ○ **/opt/sambaflow/bin/snfadm**



```
/NODE/XRDU_0/RDU_0/PCIE_8              | N/A                 | Present          | Online
/NODE/XRDU_0/RDU_0/PCIE_9              | N/A                 | Present          | Online
/NODE/XRDU_0/RDU_0/PCIE_10             | N/A                 | Present          | Online
/NODE/XRDU_0/RDU_0/PCIE_11             | N/A                 | Present          | Online
/NODE/XRDU_0/RDU_0/TILE_0              | N/A                 | Present          | Online
/NODE/XRDU_0/RDU_0/TILE_1              | N/A                 | Present          | Online
/NODE/XRDU_0/RDU_0/TILE_2              | N/A                 | Present          | Online
/NODE/XRDU_0/RDU_0/TILE_3              | N/A                 | Present          | Online
/NODE/XRDU_0/RDU_1                     | 407030B460D05B55    | Present          | Online
/NODE/XRDU_0/RDU_1/DDRCH_0/DIMM_G0     | 22B0D4A             | Present          | Online
/NODE/XRDU_0/RDU_1/DDRCH_0/DIMM_G1     | 22B0EB8             | Present          | Online
/NODE/XRDU_0/RDU_1/DDRCH_1/DIMM_H0     | 22B0D45             | Present          | Online
/NODE/XRDU_0/RDU_1/DDRCH_1/DIMM_H1     | 22B0D3A             | Present          | Online
```

Application Diagnostics and Debugging

- **Debuggability** - debug when something is wrong
  - slurm_feeder for pef contents
  - stdout
  - Syslog-based logging:
    - **sn.log/snd.log**
    - /var/log/sambaflow/runtime
- **Observability** - show what happens in the application
  - Raise exceptions to the application programmatically
  - Syslog-based logging:
    - **sn.log/snd.log**
    - /var/log/sambaflow/runtime

- **Diagnostics** - show what happens on RDU
  - Compute statistics
    - sntilestat tool
  - Memory statistics
    - snddrstat tool
  - IO statistics
    - snpciestat tool
- **SambaTune**
  - A tool to help users gain insights in model performance

More Details

- Get more details on Sambanova Public Docs
  - [SambaFlow developer documentation]
- Contact Sambanova Support team
  - help@sambanova.ai
- Go to the Support Portal
  - support.sambanova.ai

Thank you