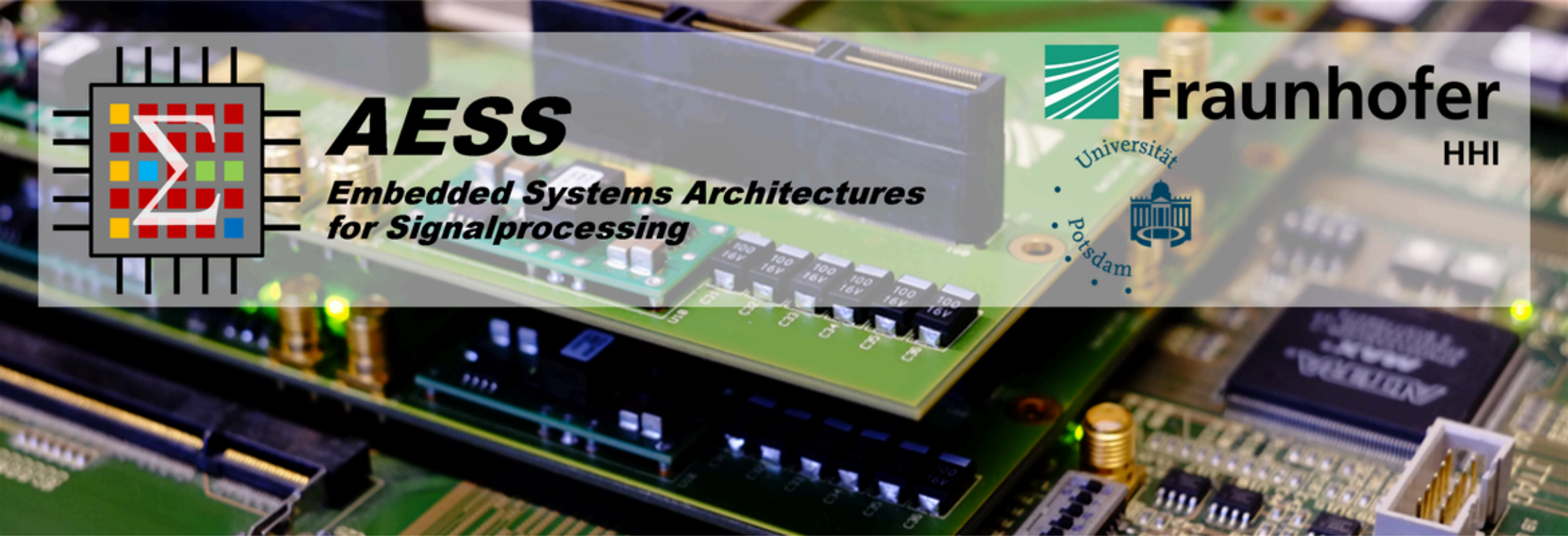
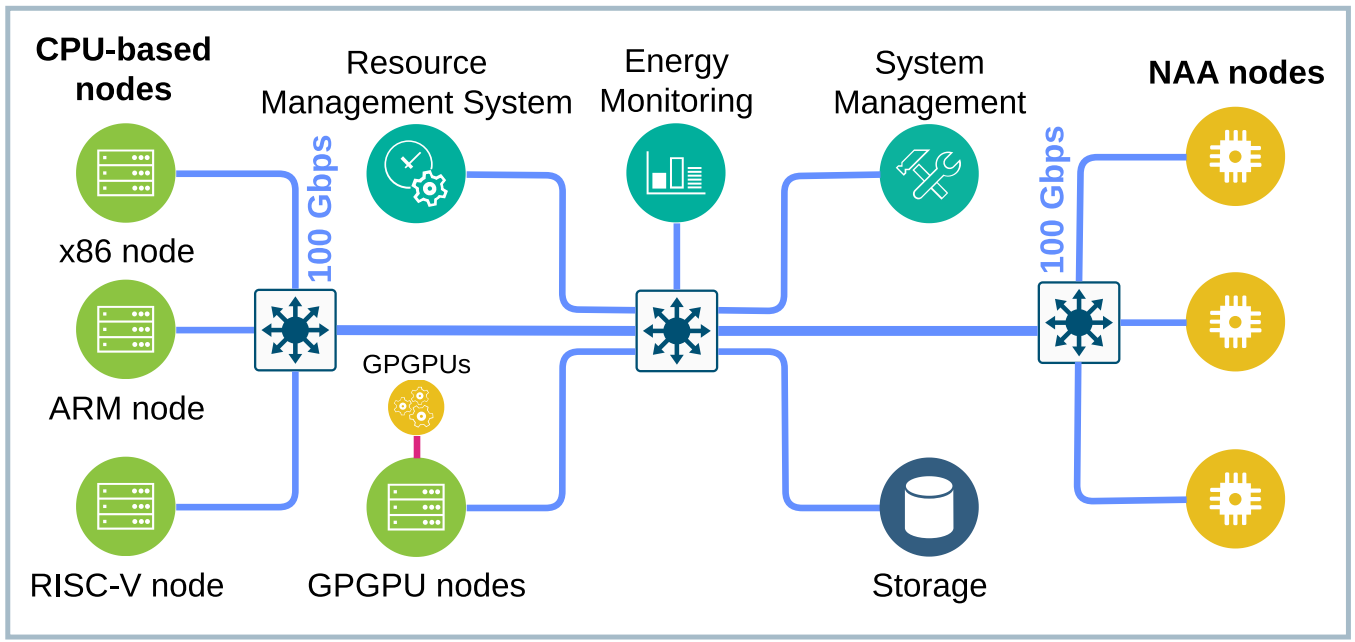


Using FPGA-based Network-Attached Accelerators for Energy-Efficient AI Training in HPC Datacenters

Niklas Schelten, Steffen Christgau, Merit Hutzler, Philipp Kreowsky, Marco De Lucia, Bettina Schnor, Hannes Signer, Johannes Spazier, Benno Stabernack, Serhii Yahdzhyiev | May 25, 2026

niklas.schelten@hhi.fraunhofer.de

NAAs in Heterogenous Computing Environments (NAAICE)



Published @HCW2025

- FPGA Accelerators as first-class citizen in 100Gbps network
- Gain scalable, flexible and energy efficient accelerator infrastructure

AESS Embedded Systems Architectures for Signalprocessing

Fraunhofer HHI

NAAICE Approach

TESSERA Framework (FPGA)

- written in SpinalHDL
- low-latency 100 Gbps UDP/IP Stack
- RoCEv2 (Remote Direct Memory Access)
- Hardware Abstraction Layer (HAL):
  - External Memory access (i.e. DDR4 over 512 bit AXI4 @340 MHz)
  - Management over network (64bit AXI4-Lite)
- Advanced debugging utilities

Supporting Software

- RPC-like invocation of FPGA kernels over RoCEv2
- Exchange of memory regions (MRSP)
- 5 well defined RPC functions (similar to MPI):
  1. `naa_create`
  2. `naa_invoke`
  3. `naa_test / naa_wait`
  4. `naa_finalize`

NAAICE Approach

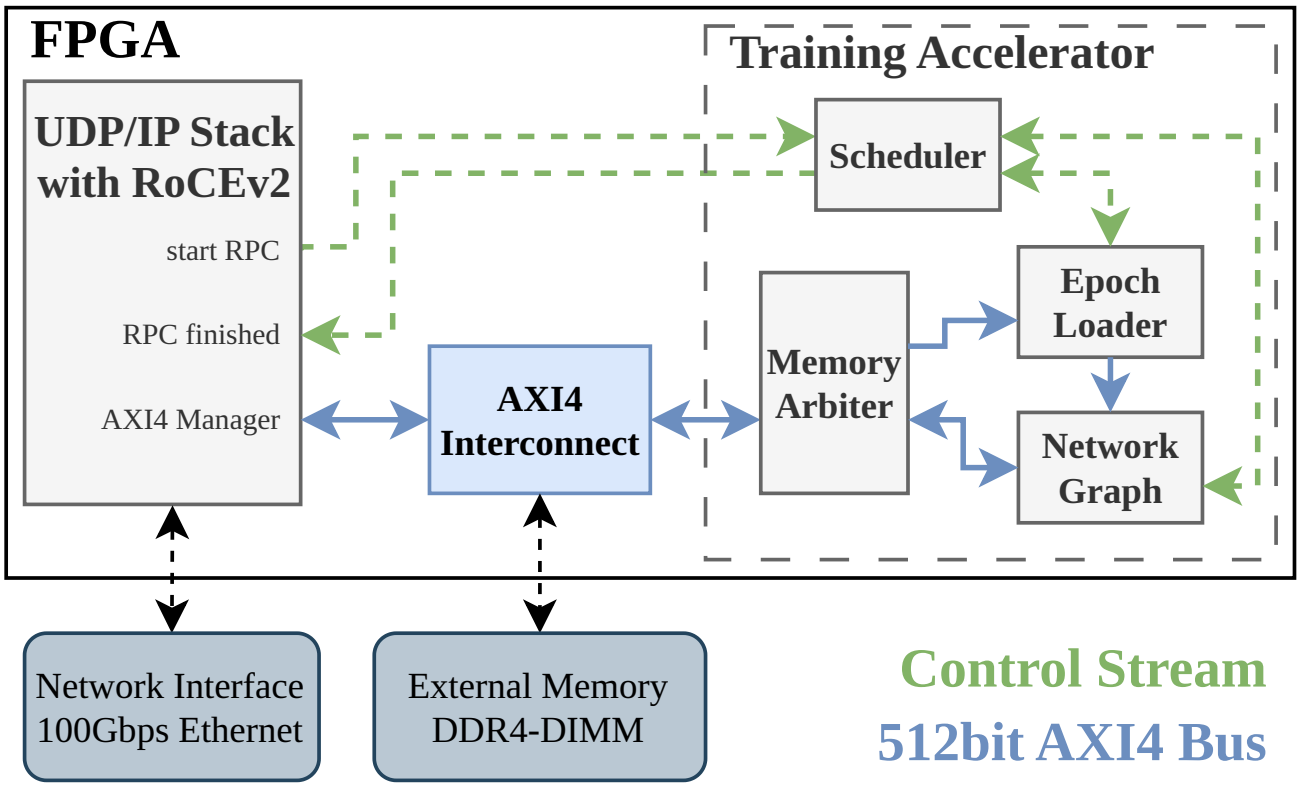
TESSERA Framework (FPGA)

- written in SpinalHDL
- low-latency 100 Gbps UDP/IP Stack
- RoCEv2 (Remote Direct Memory Access)
- Hardware Abstraction Layer (HAL):
  - External Memory access (i.e. DDR4 over 512 bit AXI4 @340 MHz)
  - Management over network (64bit AXI4-Lite)
- Advanced debugging utilities

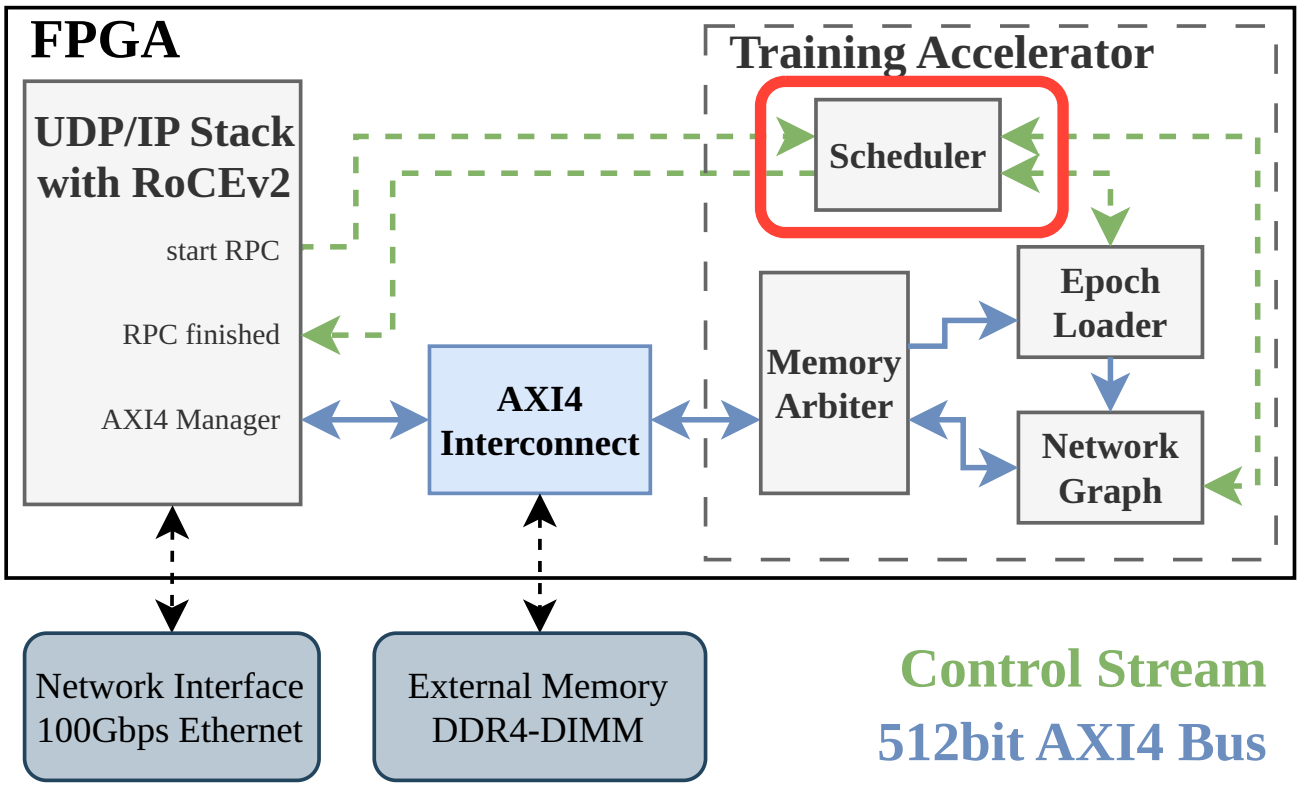
Supporting Software

- RPC-like invocation of FPGA kernels over RoCEv2
- Exchange of memory regions (MRSP)
- 5 well defined RPC functions (similar to MPI):
  1. `naa_create`
  2. `naa_invoke`
  3. `naa_test / naa_wait`
  4. `naa_finalize`

FPGA Implementation



FPGA Implementation



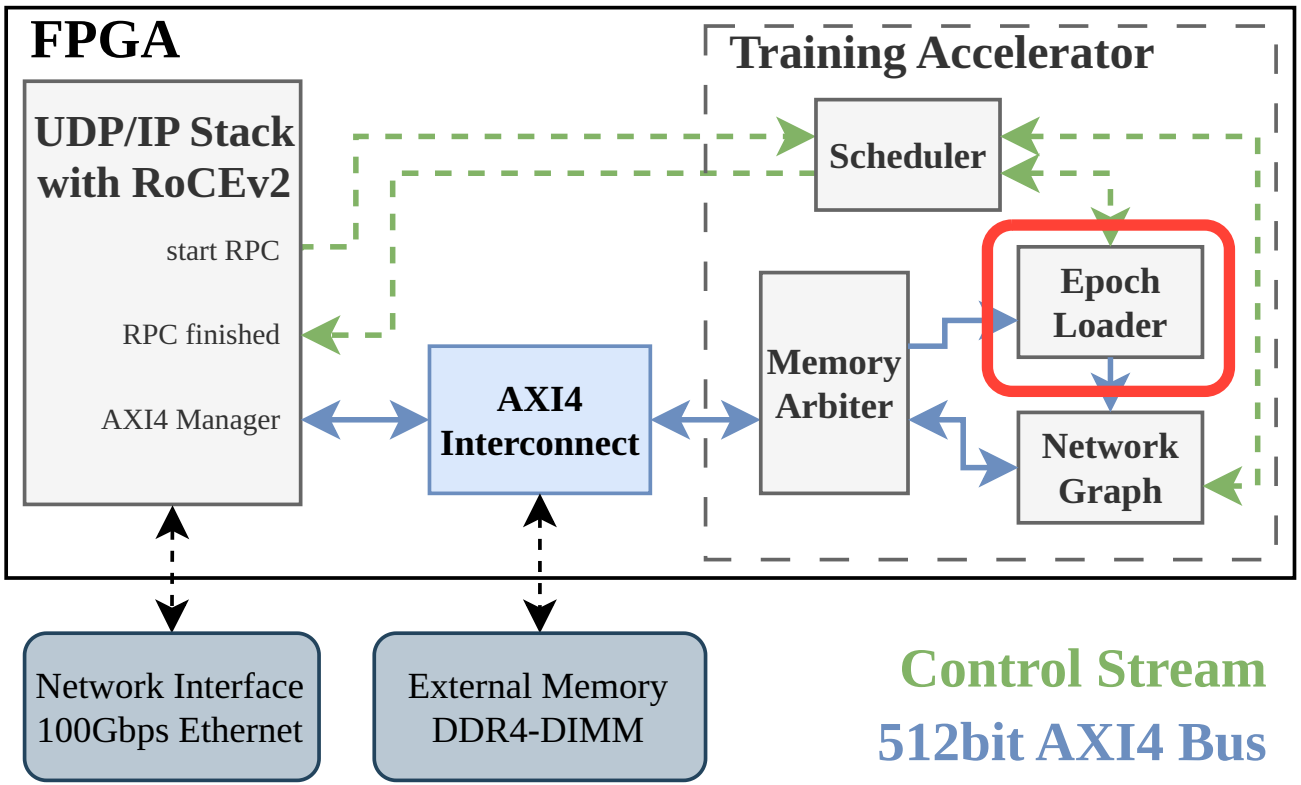
Scheduler

- Handles incoming RPCs
- Forwards MRs to Epoch Loade + Network Graph
- Initiates write-back of result

AESS Embedded Systems Architectures for Signalprocessing

Fraunhofer HHI

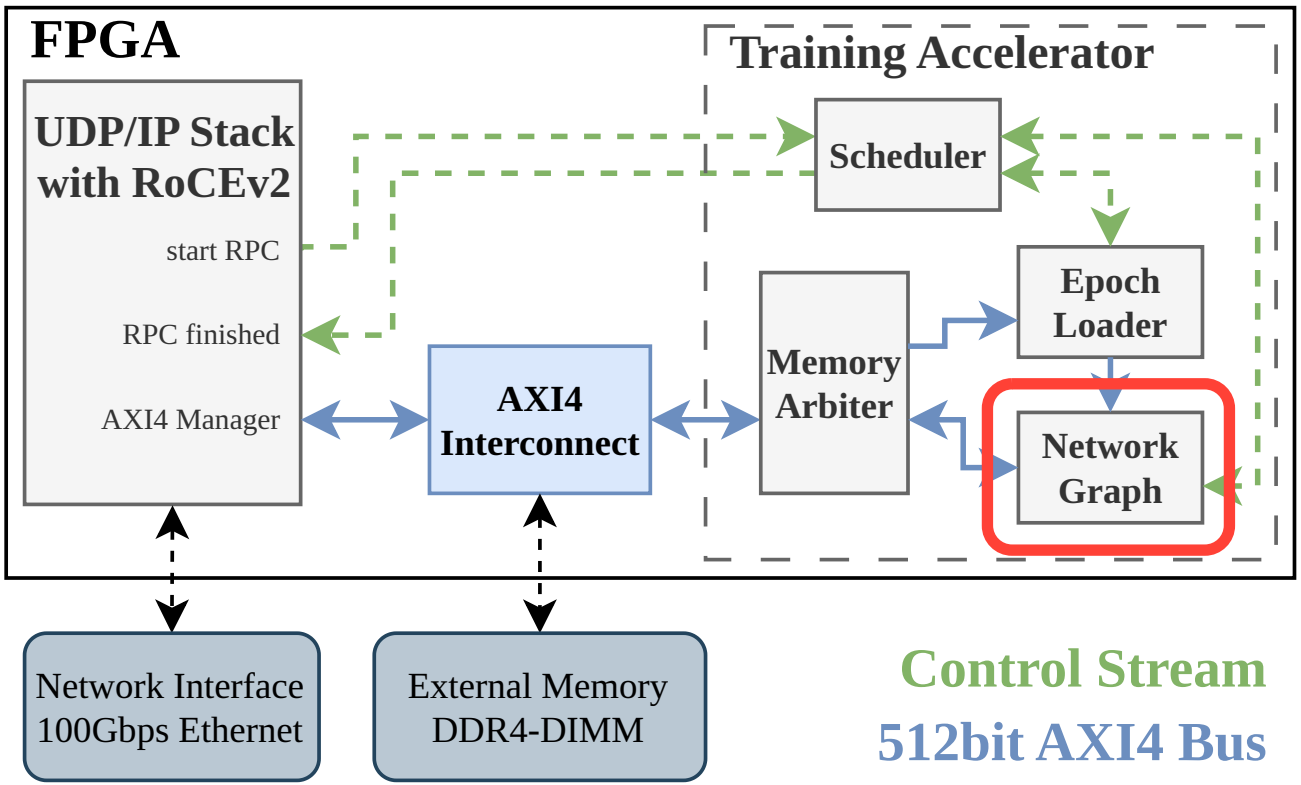
FPGA Implementation



Epoch Loader

- Splits training data into batches
- Randomizes batches with Fisher-Yates shuffle
- Triggers weight update in network graph

FPGA Implementation



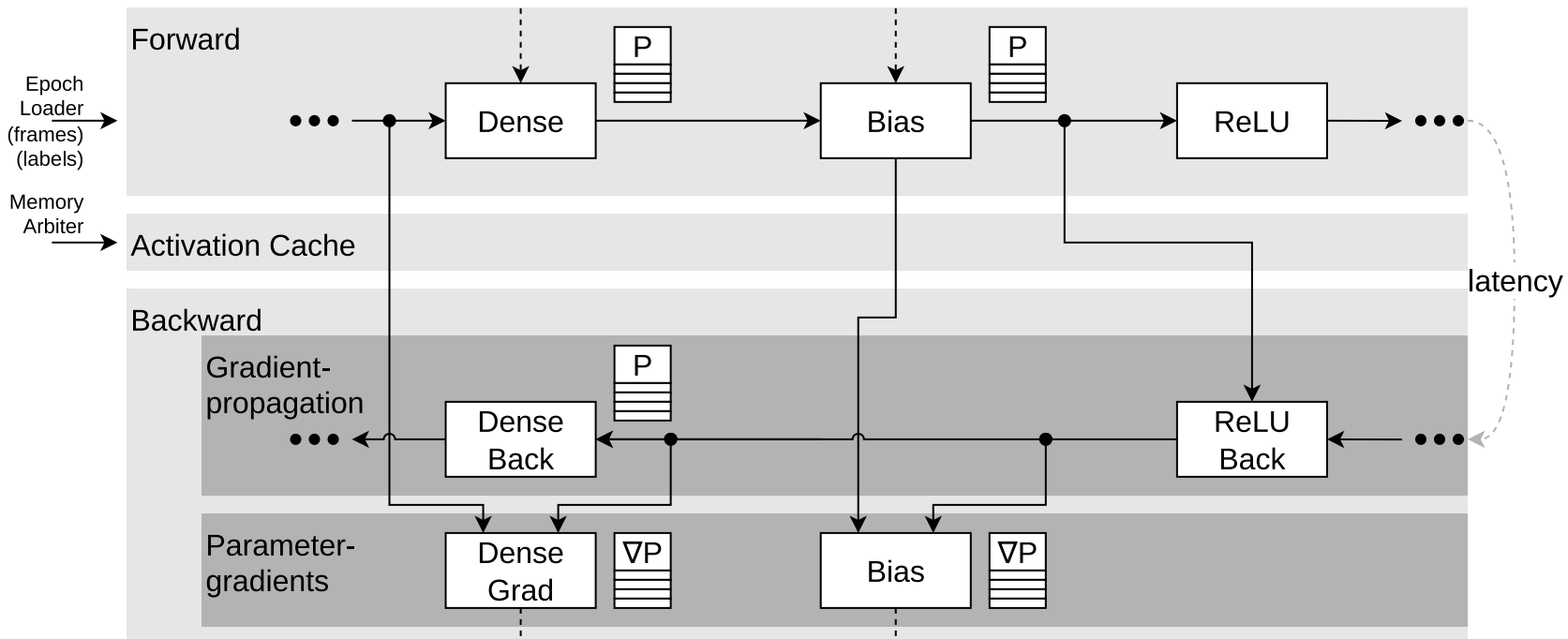
Network Graph

- Based on Keras-like model description
- HW constraints + resource estimation of layers
  → configuration with target frequency and max. throughput

AESS Embedded Systems Architectures for Signalprocessing

Fraunhofer HHI

Network Graph Details



AESS Embedded Systems Architectures for Signalprocessing

Fraunhofer HHI

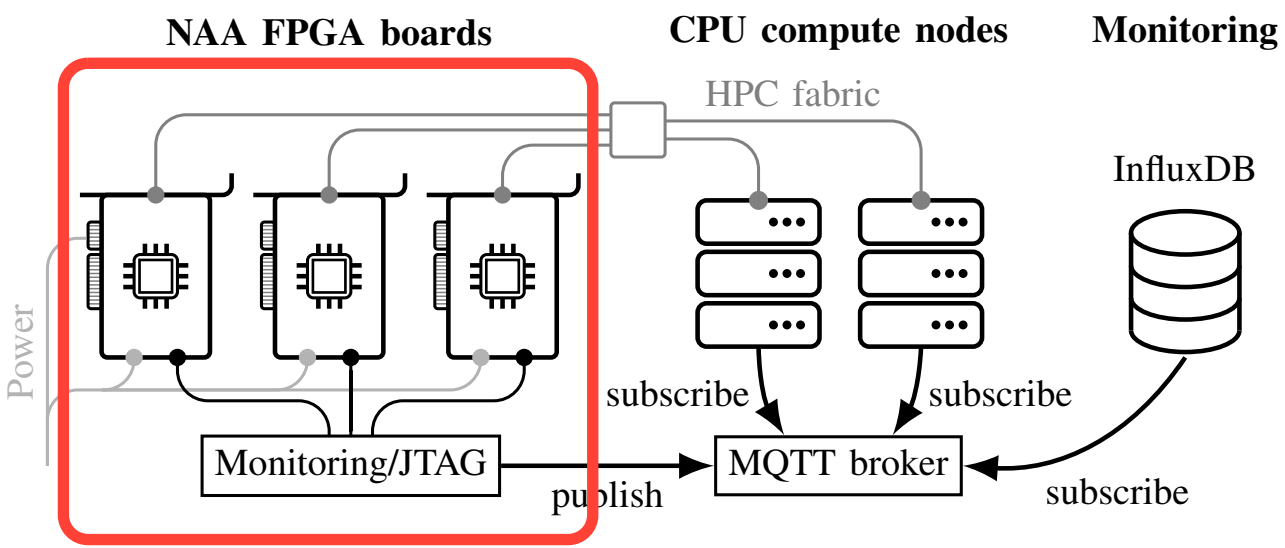
Resource Management System - *Slurm*

- FPGAs commonly treated as node resources

→ disaggregated FPGAs (i.e. NAAs) are not supported

- New *Slurm* Plugin extending its scheduler:
  1. [Prolog] First-Fit scheduler to match NAA requests before scheduling compute nodes
  2. [Prolog] RMS flashes FPGAs with requested bitstream
  3. Job Execution with *NAA Spec*: hostnames/IPs of FPGAs with requested NAA functions
  4. [Epilog] NAAs freed after job termination

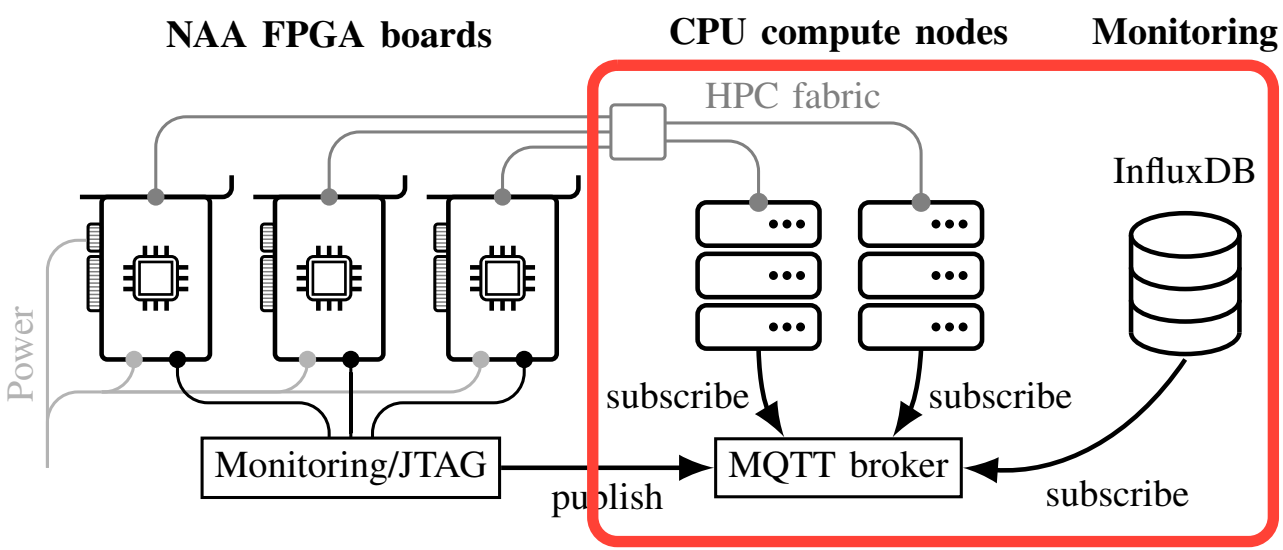
Energy Measurement for Applications (EMA)



Data Recording

- Hardware sensors report raw power data over USB/JTAG
- 1 SBC connected to many FPGAs
- Data aggregated and published to MQTT broker

Energy Measurement for Applications (EMA)



Applications & RMS

- Energy can be consumed by applications via EMA API
- InfluxDB stores time series
- Slurm Plugin can summarize energy consumption per job

Model Architecture & Loss function

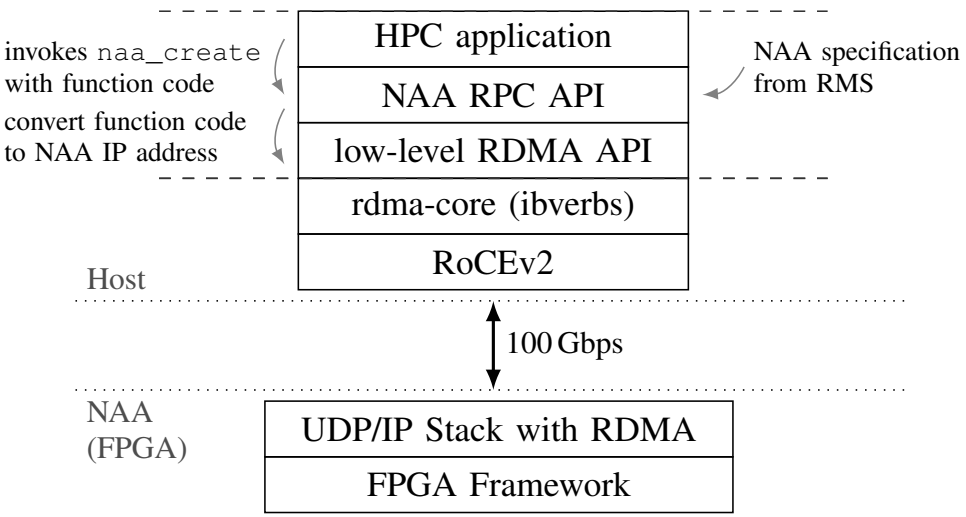
- Training of AI surrogate model for reactive transport simulator (POET)
- FNN with concentrations of dissolved elements as inputs and predicted concentrations for $t + 1$
- PINN inspired loss function based on mass balance:

$$\Delta m_e = \left| \sum_{i \in \mathbb{S}} m_{i,e}(t) - \sum_{i \in \mathbb{S}} m_{i,e}(t+1) \right| \stackrel{!}{=} 0$$

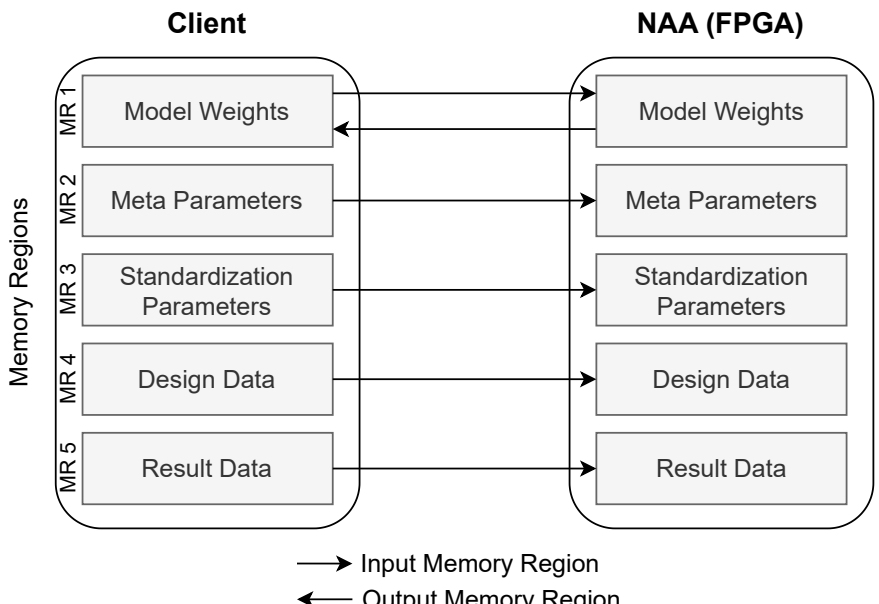
- $\Longrightarrow$ Huber loss as foundation integrating MSE and MAE
- One workload: diffusion of reactive Barite ($BaCl_2$) into mineral celestite ($SrSO_4$)

$$BaCl_2 + SrSO_4 \rightarrow BaSO_4 + Sr^{2+} + 2Cl^-$$

NAA Client / Software



Invocation Abstraction Levels



RPC Memory Regions Exchange

GPU Baseline Implementations

Tested on NVIDIA V100 16GB with AMD EPYC 7313P CPU

Keras Implementation

- Keras v3.12 + TensorFlow v2.20

→ standard AI library

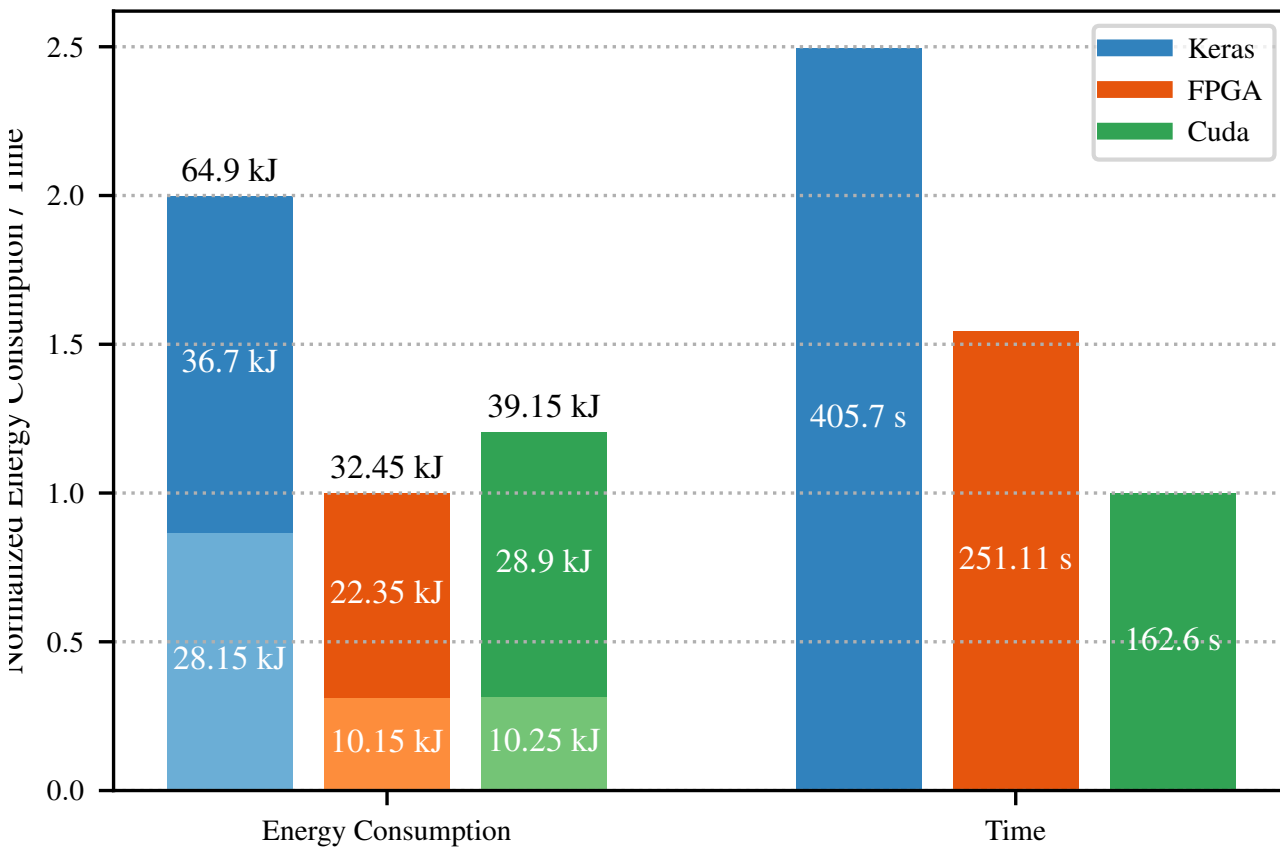
Only 37% average utilization

native CUDA implementation

- preprocessing directly on GPU
- less communication overhead because entire dateset loaded into GPU memory

95% average utilization

Evaluation - Time & Energy Consumption



Accelerator Avg Power:

- Keras: 90W
- CUDA: 178W
- FPGA: 89W

AESS Embedded Systems Architectures for Signalprocessing

Fraunhofer HHI

Summary

- HPC Integration of FPGAs as Network-Attached Accelerators
  - Slurm integration including energy measurement
- Development of AI training accelerator into existing FPGA hardware shell
- AI Training on FPGA improved energy efficiency compared to 2 GPU Baselines

Thanks for listening! Any questions?

With funding from the:

Federal Ministry of Research, Technology and Space

This project has been funded by the German Federal Ministry of Research, Technology and Space (grant # 16ME0622K, 16ME0623, 16ME0624).

AESS Embedded Systems Architectures for Signalprocessing

Fraunhofer HHI