



Semiconductor
Research
Corporation



CENTER FOR
EVOLVABLE
COMPUTING

ACE Plan of Action

ACE Center for Evolvable Computing SAB Meeting, April 2024

Director:
Josep Torrellas (UIUC)



Assistant Director:
Minlan Yu (Harvard)



<https://acecenter.grainger.illinois.edu/>

What ACE is About

Devise technologies for **distributed computing** to improve the **energy efficiency** and the performance of applications by 100x

How to do it?

Leverage hardware accelerators and integration

Minimize data movement

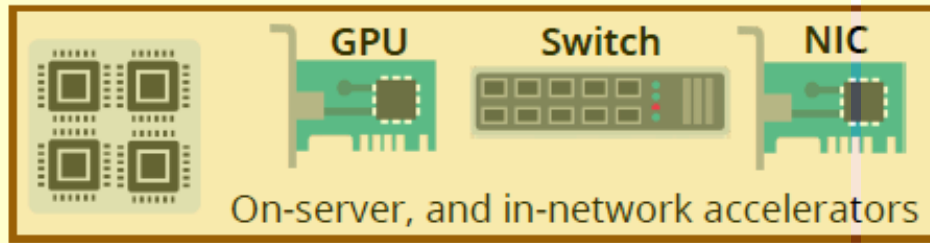
Co-design hardware and software innovations

Integrate security and correctness from the ground up

Tightly Coupled Organization



c) Theme 3: Fine-grained Communication & Coordination



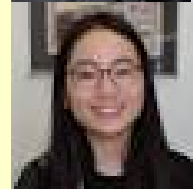
a) Theme 1: Heterogeneous Computing Platforms



b) Theme 2: Distributed Evolvable Memory and Storage

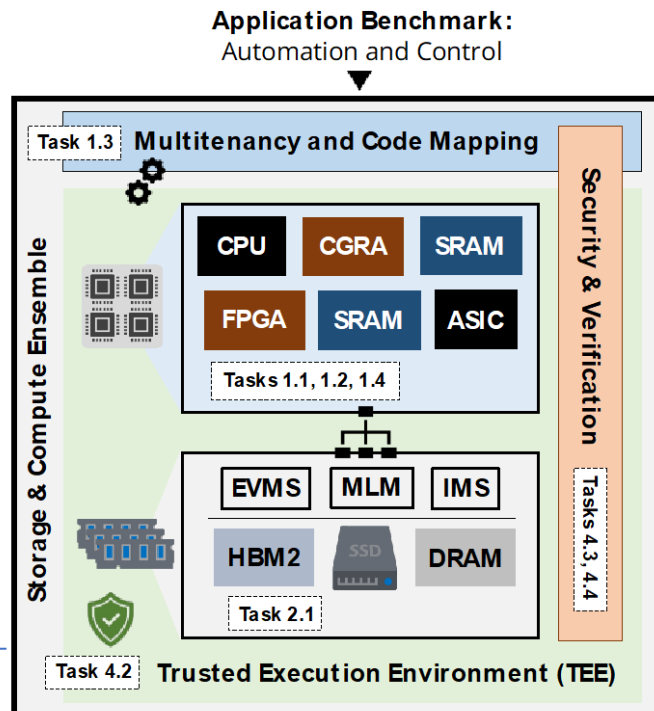


d) Theme 4: Security, Privacy, Correctness



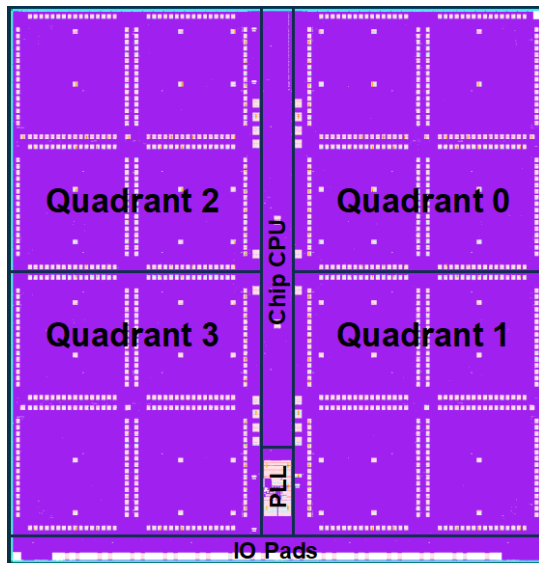
Demonstrator 1: A Reconfigurable Multi-accelerator Compute Ensemble

- Evolvable accelerators (FPGA, CGRA, and ASIC) in chiplets
- Aggregated into a multichip module (MCM) ensemble
- Ensemble will be multitenant
- Compiler will generate code and map it to compute units
- Other software will be ported



Configurable Chiplet for Composable Computing

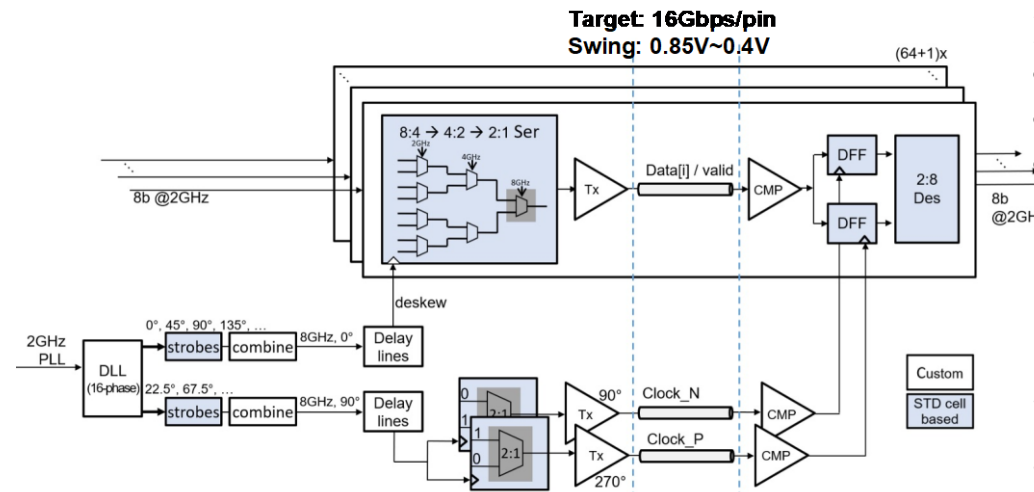
- A configurable chiplet that can adapt to changing workloads (test chip to be taped out in May 2024)
 - Made of a fine-grained mix of CPU and systolic array tiles for flexible workload mapping & adaptation
- A 16Gbps open UCIe interface for connecting chiplets (test chip tapeout planned for August 2024)
 - Make the interface synthesizable and develop an automatic interface generator → Public domain
- PI: Zhengya Zhang (Michigan)
- Chiplet interface research is co-sponsored by Intel and joint work with Intel.



Layout of accelerator test chiplet

Accelerator test chiplet

- Intel16
- Tapeout: 24Q2
- 4 mm x 4 mm
- 700 MHz
- 500-1000 GFLOPS
- Target ML and DSP applications



I/O interface test chip

- Intel16
- Tapeout: 24Q3
- Target 16Gbps/lane

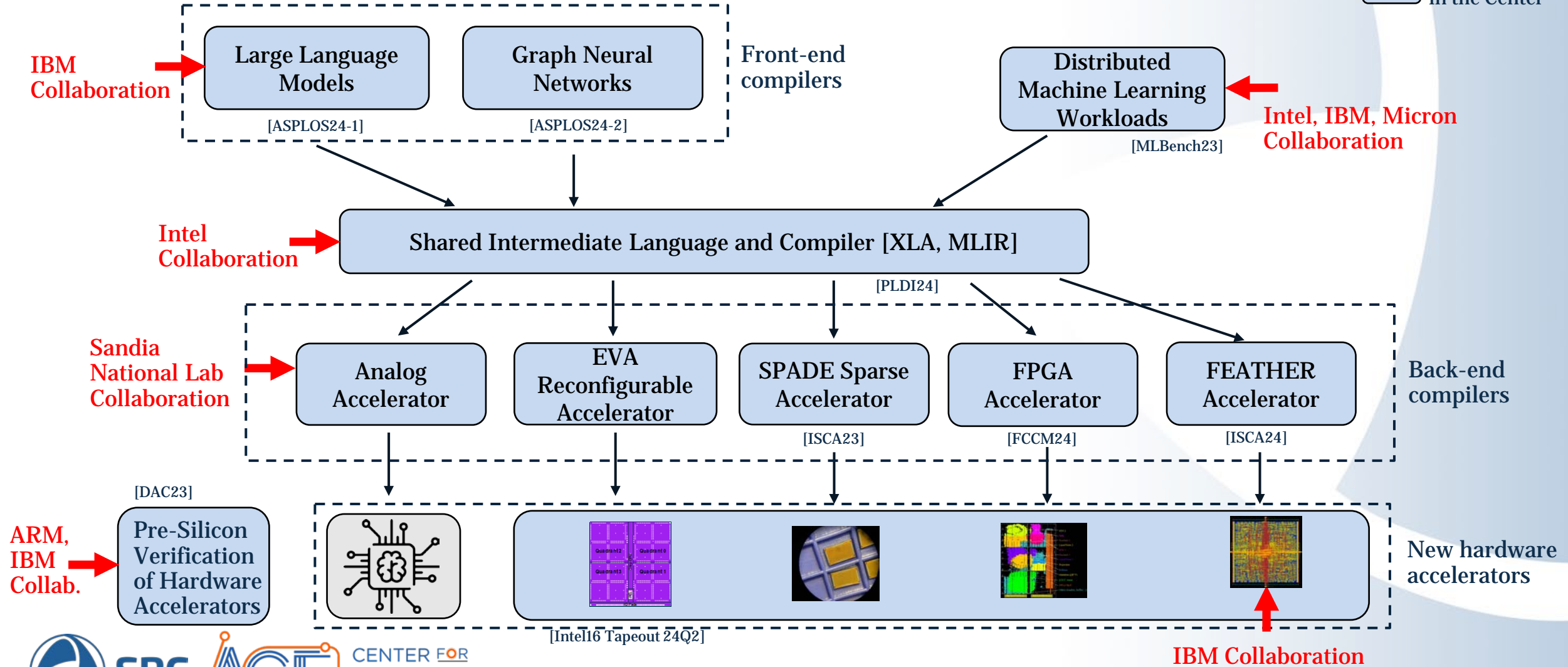
UCIe PHY architecture

Goal: have a chiplet library of highly reusable chiplets.
 Show how they can be connected with modularity into bigger systems.

An Open-Source Software Framework to Program Accelerators for AI

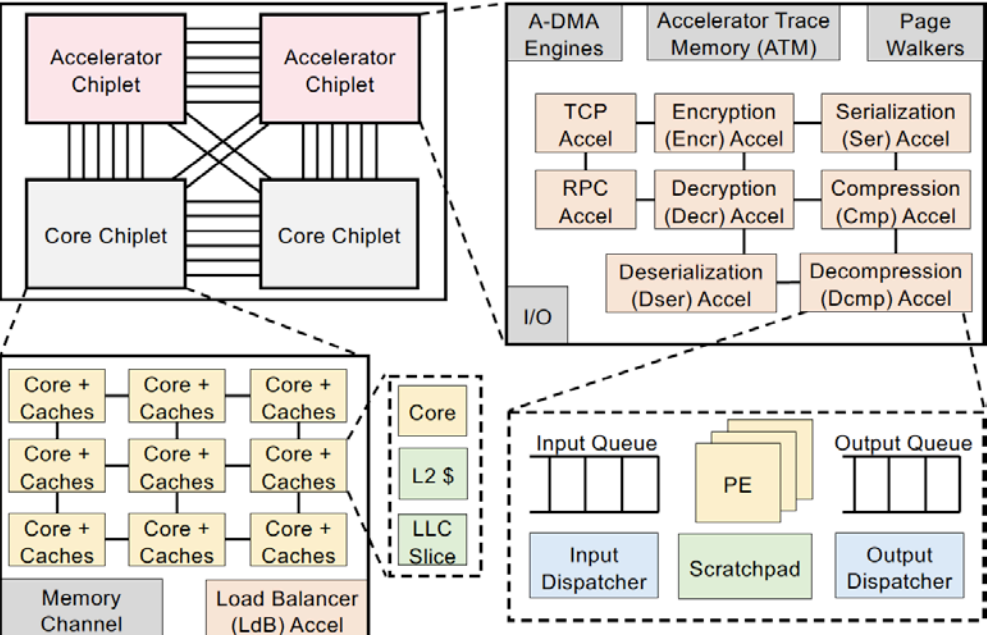
- PI: Charith Mendis (Illinois), Zhiru Zhang (Cornell), several others

Items developed in the Center

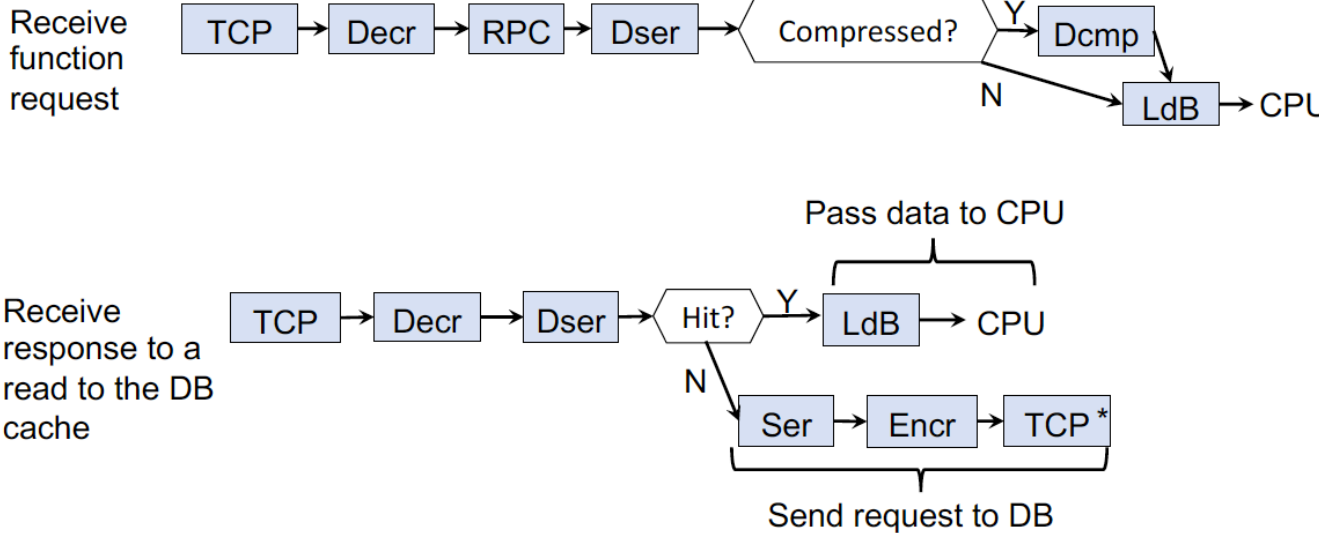


AccelFlow: Orchestrating an Ensemble of Accelerators for Microservice Environments

- Managing ensembles of accelerators for datacenter taxes: TCP, (De)Encryption (Decr and Encr), RPC, (De)Serialization (Dser and Ser), (De)Compression (Dcmp and Cmp), and load balancing (LdB).
- PI: Josep Torrellas (Illinois)



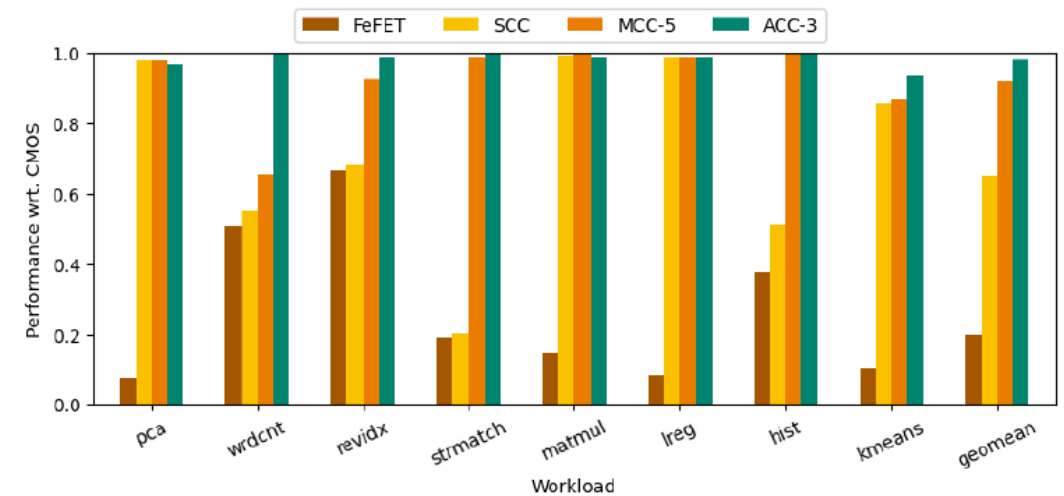
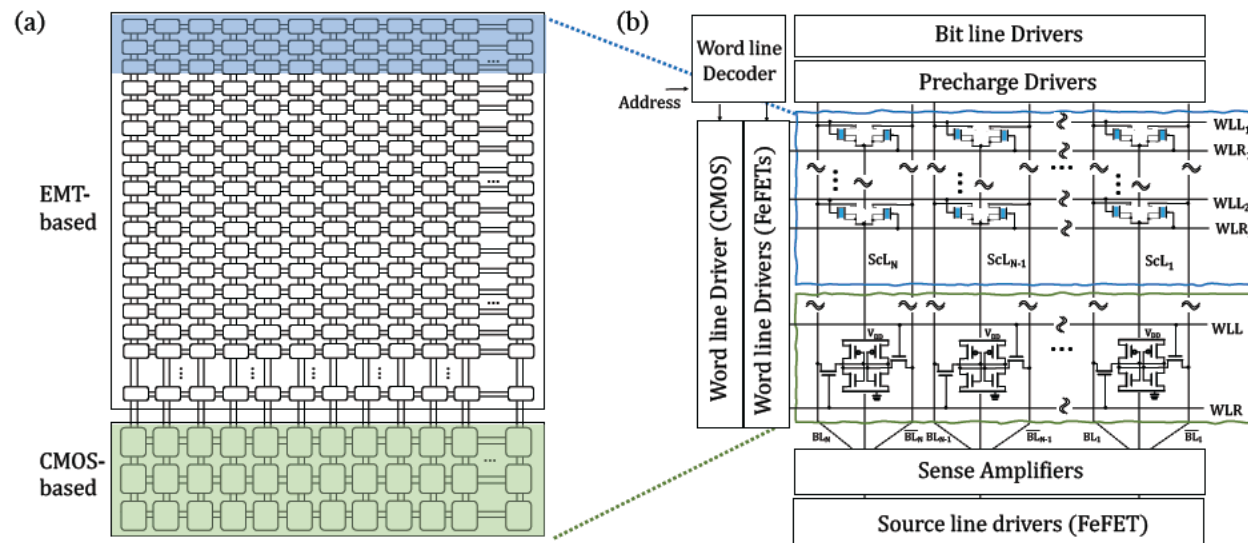
Processor with an AccelFlow accelerator ensemble



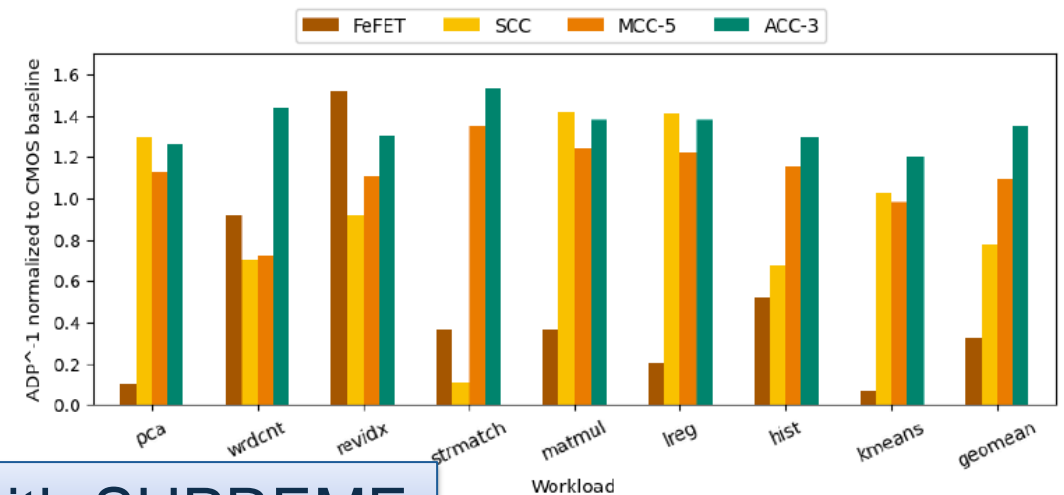
Examples of sequences of accelerators (traces)

HybridCAPE: Associative Processing in Memory Combining CMOS and FeFET

- Integrated, multilevel hybrid-technology SRAM
 - Majority FeFET (area/power \downarrow wr delay \uparrow)
 - CMOS buffers for select wr (area/power \uparrow wr delay \downarrow)
- Hidden from programmer or compiler
- PI: Jose Martinez (Cornell)

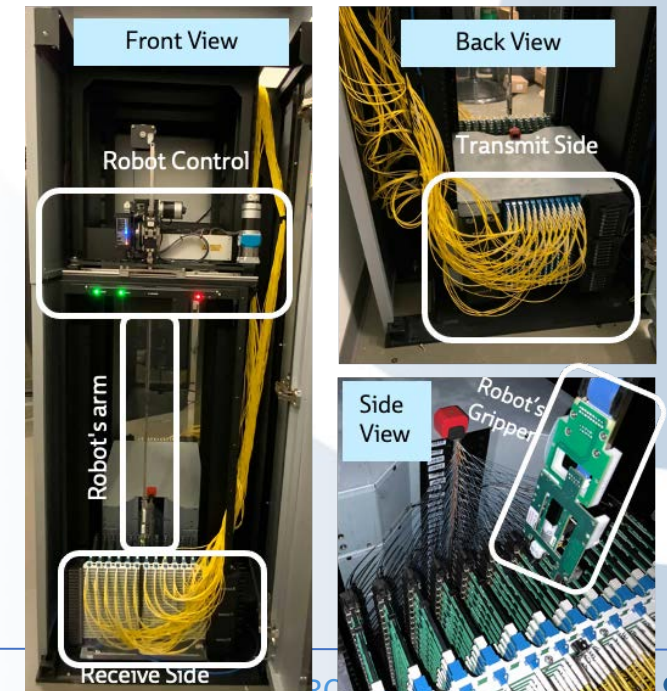
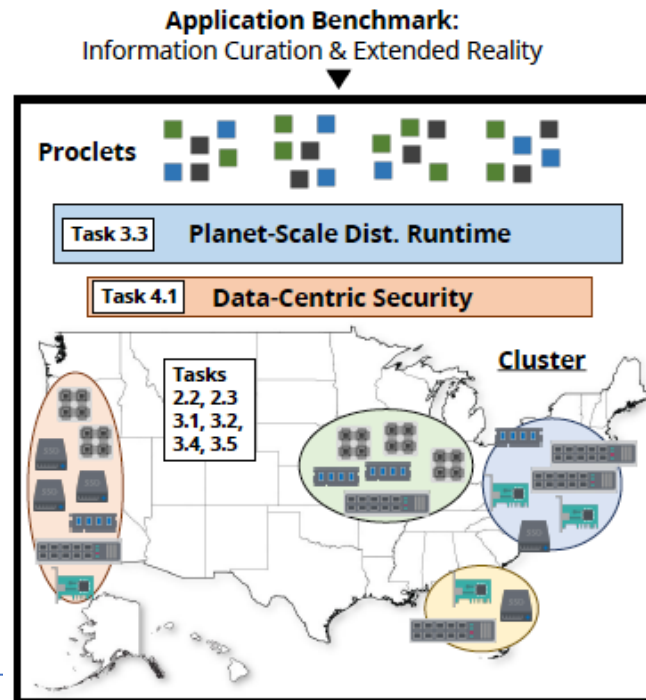


(a) Performance



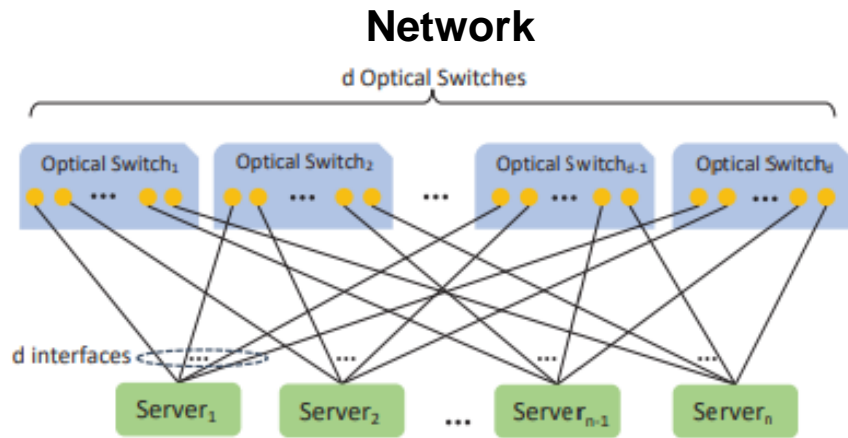
Demonstrator 2: Heterogeneous Large Cluster with Specialized Intelligence

- A large compute cluster composed of accelerators
- Runtime with fine-grained tasks & customized communication stack
- Accelerators in switches and smartNICs
- Security mechanisms

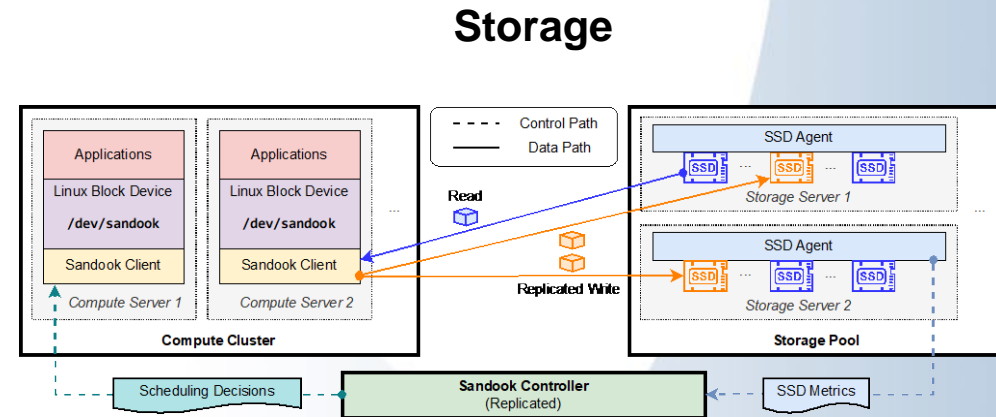


Self-balancing Infrastructure: Reconfigurable Networking & Storage

- A **reconfigurable** optical interconnect that adjusts the network topology based on application requirements.
- Sandook: A logically-centralized storage disaggregation system that **steers I/O** to disks based on knowledge of network topology and reconfiguration delays.
- PIs: Adam Belay (MIT), Manya Ghobadi (MIT)



Goal: Use reconfigurability to optimize networking & storage together

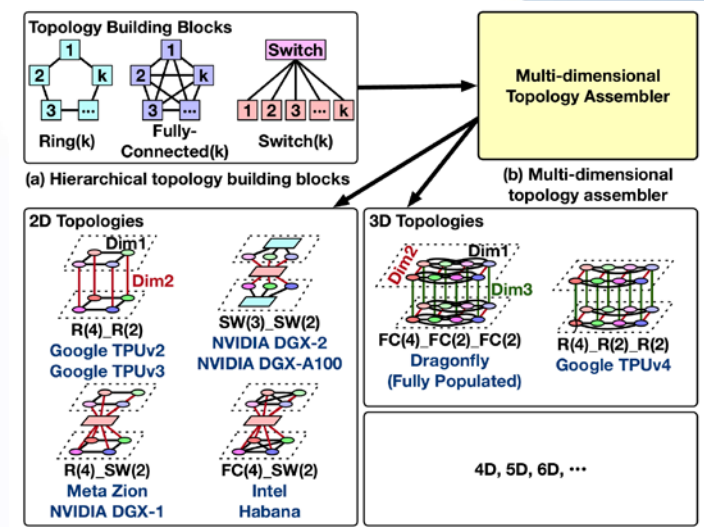
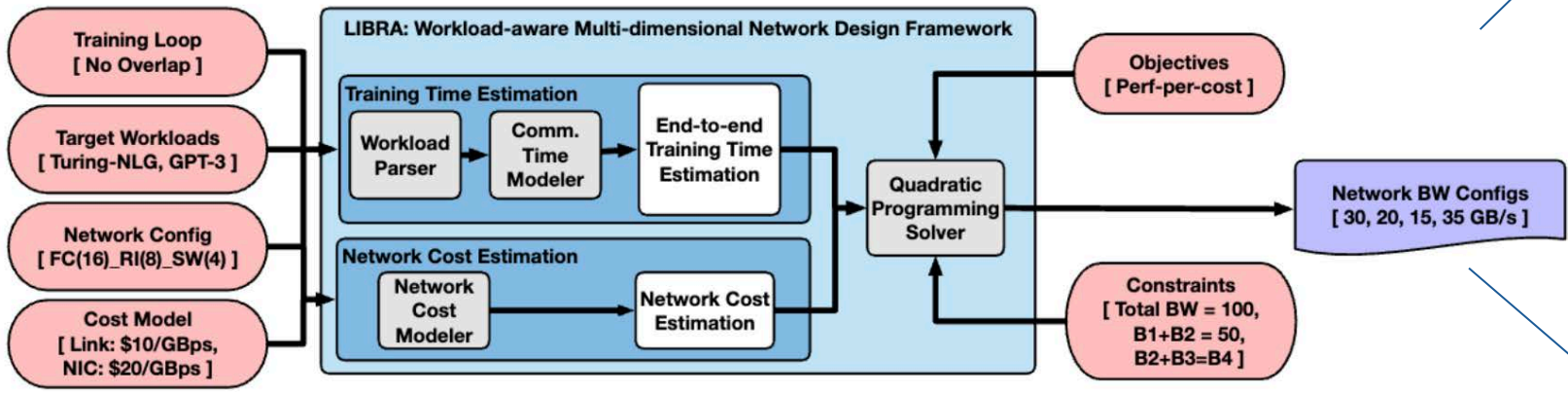


Conceptual sketch of our reconfigurable infrastructure

- Motivation: hardware is massively underutilized in the cloud
- Shard across pool of flash SSDs → poor performance due to hotspots, heterogeneity, and read/write interference
- *Sandook*: Rapid request steering to rebalance resource use

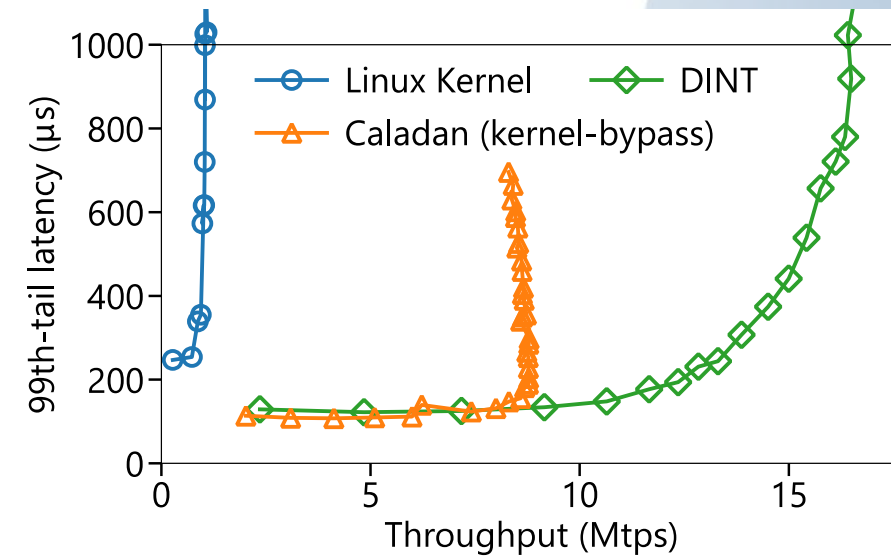
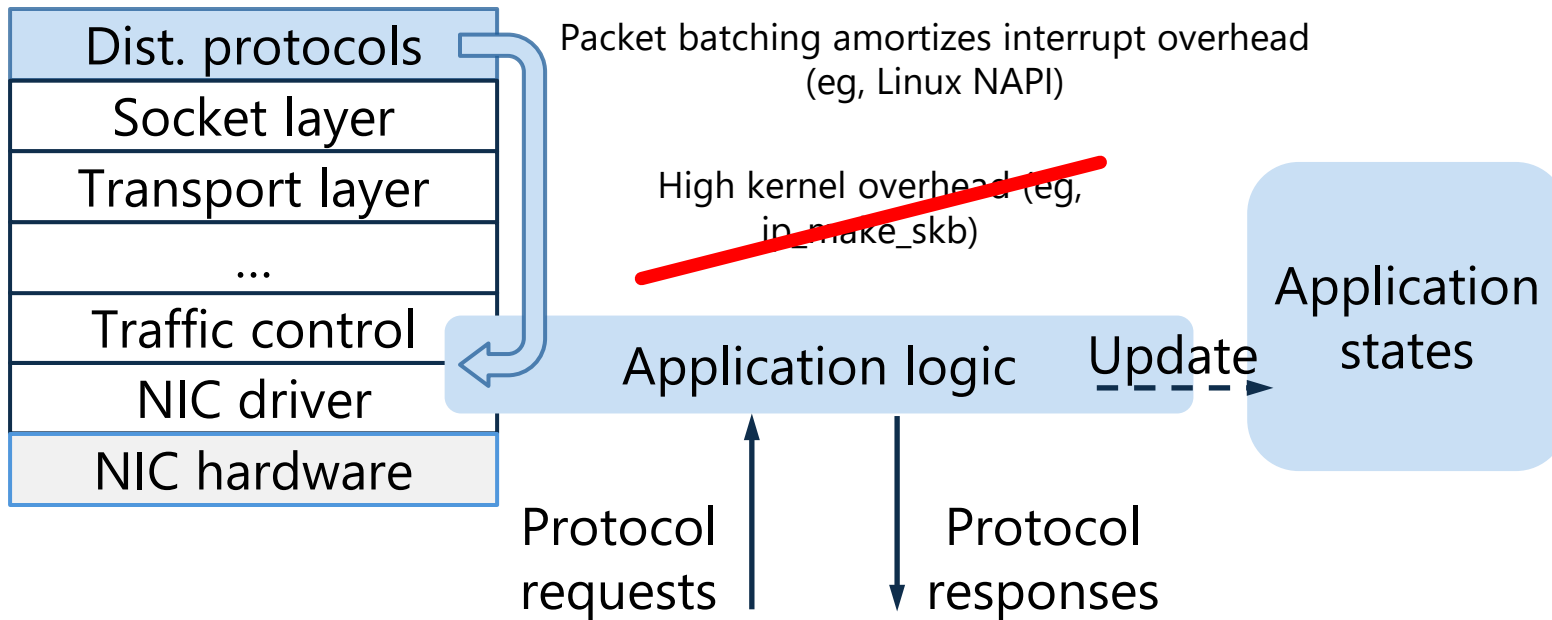
LIBRA: Workload-aware Network Topology Optimization for Distributed AI Training

- Companies are building distributed clusters optimized for AI workloads (e.g., Intel Habana Labs, Nvidia DGX, Meta ZionEX, ..)
 - Scaling to 1000s of accelerators needs network fabrics using a diversity of link technologies
 - On-package → On-rack → Rack-to-rack
- Challenges: Heterogeneous link bandwidths (affects *performance*) + Trade-off of \$/bandwidth across technologies (affects *cost*)
- Solution: We developed LIBRA, a framework to enable design-space exploration of networks
 - Recommend the optimal network topology and BW allocation for maximizing performance/cost for given AI workload(s)
- PI: Tushar Krishna (GATech), in collaboration with Intel



Application-customized Networking Stack with eBPF

- Accelerate distributed system network software while maintaining safety
 - *Electrode*: Accelerate consensus protocols with new communication primitives in eBPF
 - *Dint*: Accelerate distributed transactions (lock manager, key-value store, log manager)
 - Significant throughput speedup and latency reduction
- PI: Minlan Yu (Harvard)



CC-NIC: A Cache-Coherent Interface to the Network Interface Card (NIC)

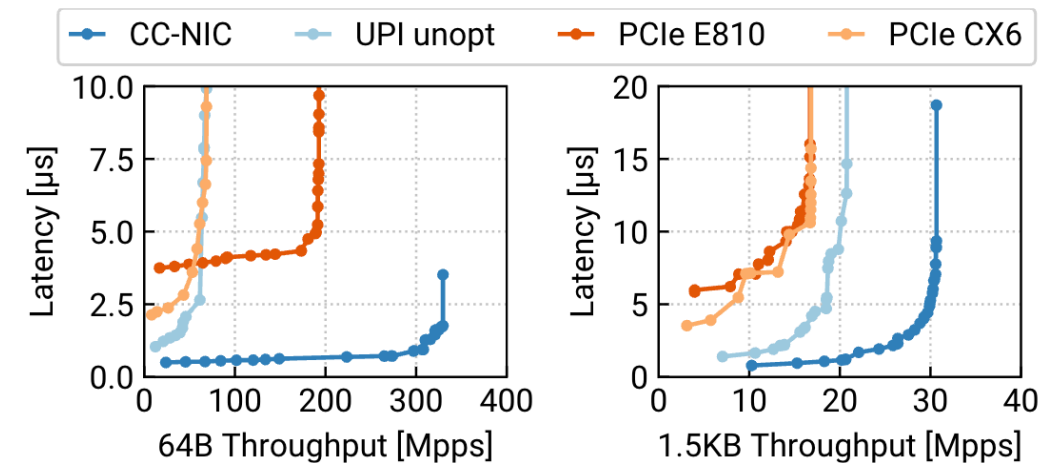
Today's PCIe NIC transmit-receive interfaces are inefficient: PCIe is >80% of intra-rack roundtrip latency

Emerging coherent interconnects (CXL, UPI, UCle & others) offer a solution

- Devices participate in CPU cache coherence protocol

CC-NIC is a cache-coherent NIC design

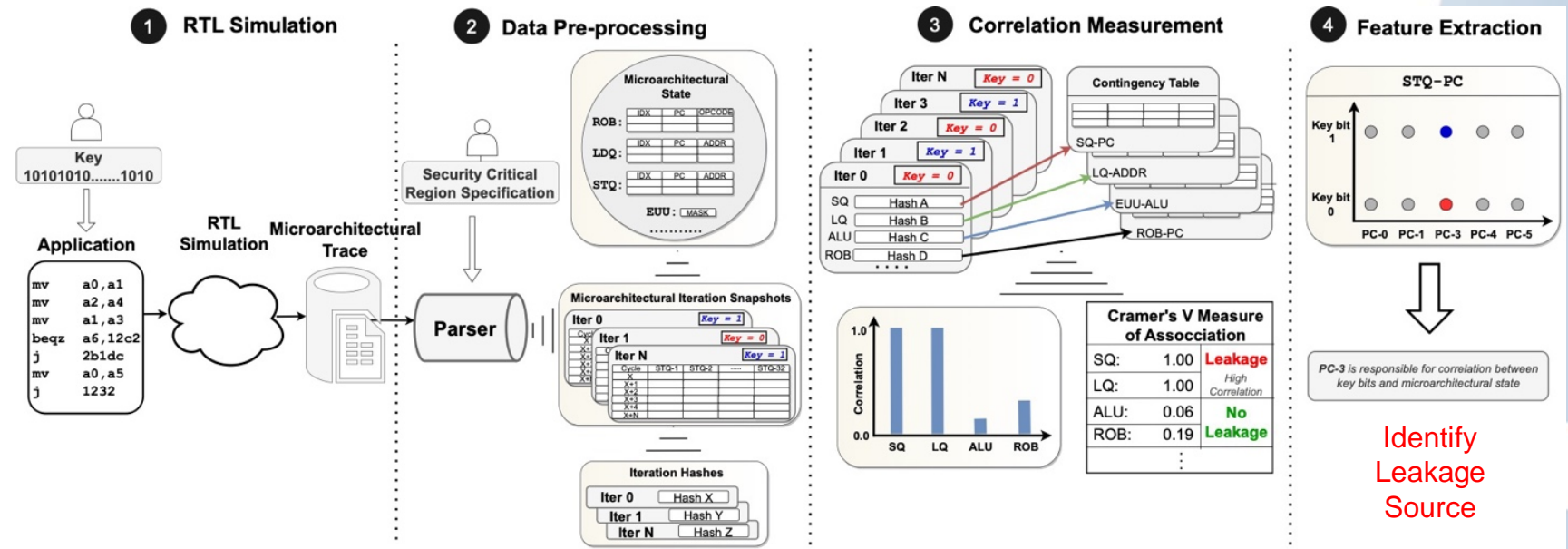
- Coherence calls for new techniques in:
 - *Hardware host-device signaling*
 - *Cache-optimized memory layouts*
 - *Symmetric, shared buffer management*
- CC-NIC evaluated on Intel UPI: terabit throughput on Sapphire Rapids, significant latency reduction vs PCIe, up to 50% core savings for app workloads
- PI: Arvind Krishnamurthy (U Washington)



MicroSampler: Microarchitecture-Level Leakage Detection in Constant Time Execution

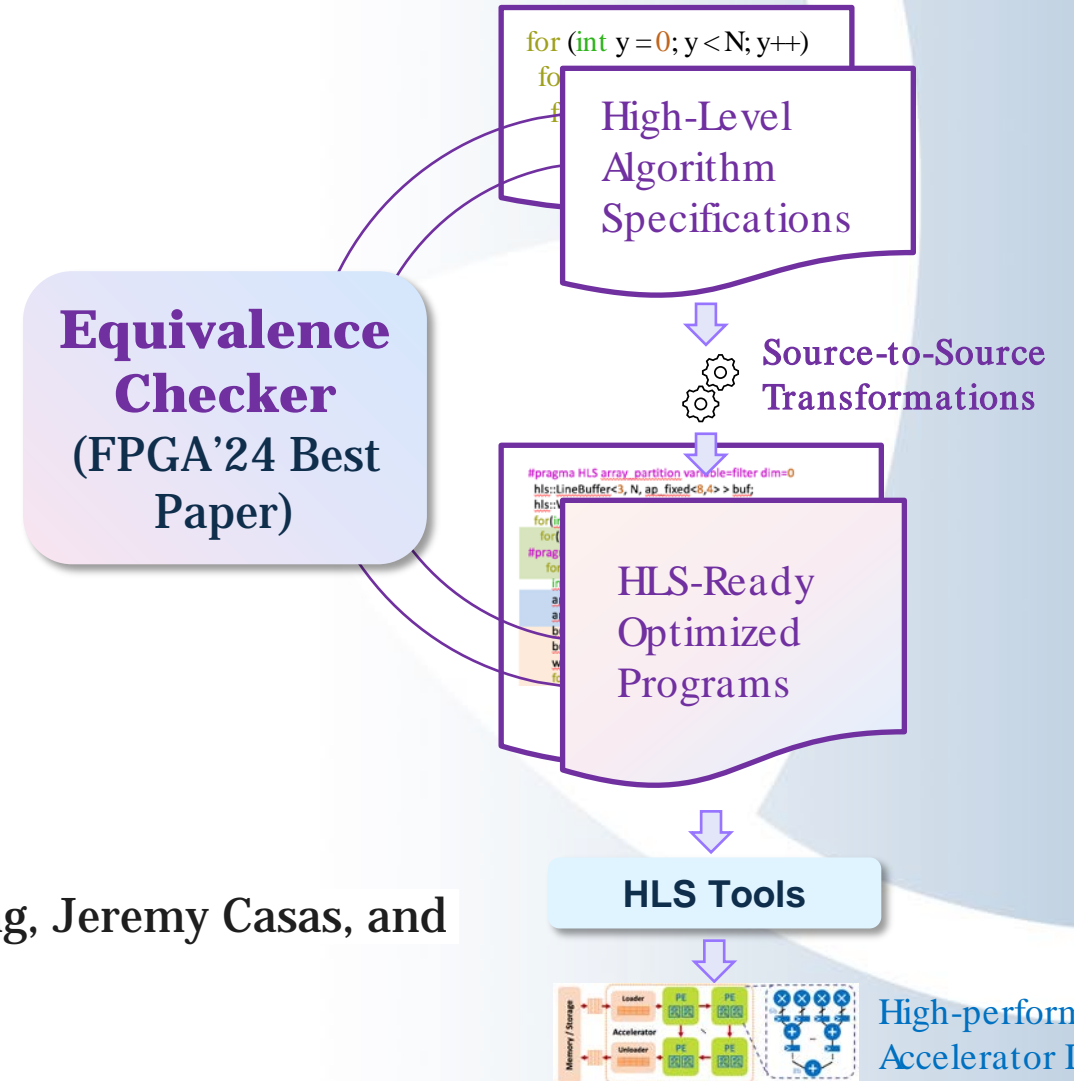
- Constant-time programming, a principal defense against timing side channels, relies on frequently incorrect, incomplete or untested assumptions about underlying microarchitecture
- **MicroSampler**: a dynamic verification framework for automatically identifying execution behavior that exhibits correlations with secret data:
 - (1) RTL simulation of target system running crypto code, (2) extraction of complete microarchitectural state, (3) correlation analysis with secret data, (4) pinpointing leakage source if correlation identified
- PI: Radu Teodorescu (Ohio State) in collaboration with Intel

Goal: Open-source toolchain for automatic leakage detection in constant-time security code, seamlessly integrated into existing verification flows.



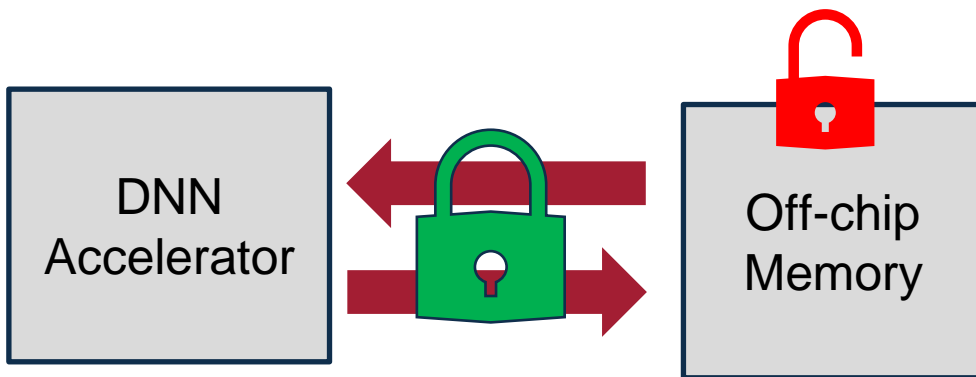
Formal Verification of Source-to-Source Transformations for HLS

- Efficiently checks semantic equivalence between two HLS C/C++ programs
- Robust and practical
 - Build a form of canonical representation of the HLS program, independent of the syntax used
 - Support custom memory optimizations, dataflow concurrency, etc.
- Highly scalable
 - Verification in time/space linear w.r.t. OPs executed (~500K Ops/s in verification throughput)
 - A complex 64x64 systolic array verified in 16 minutes
- PI: Zhiru Zhang (Cornell) with feedback from Intel (Jin Yang, Jeremy Casas, and Zhenkun Yang)

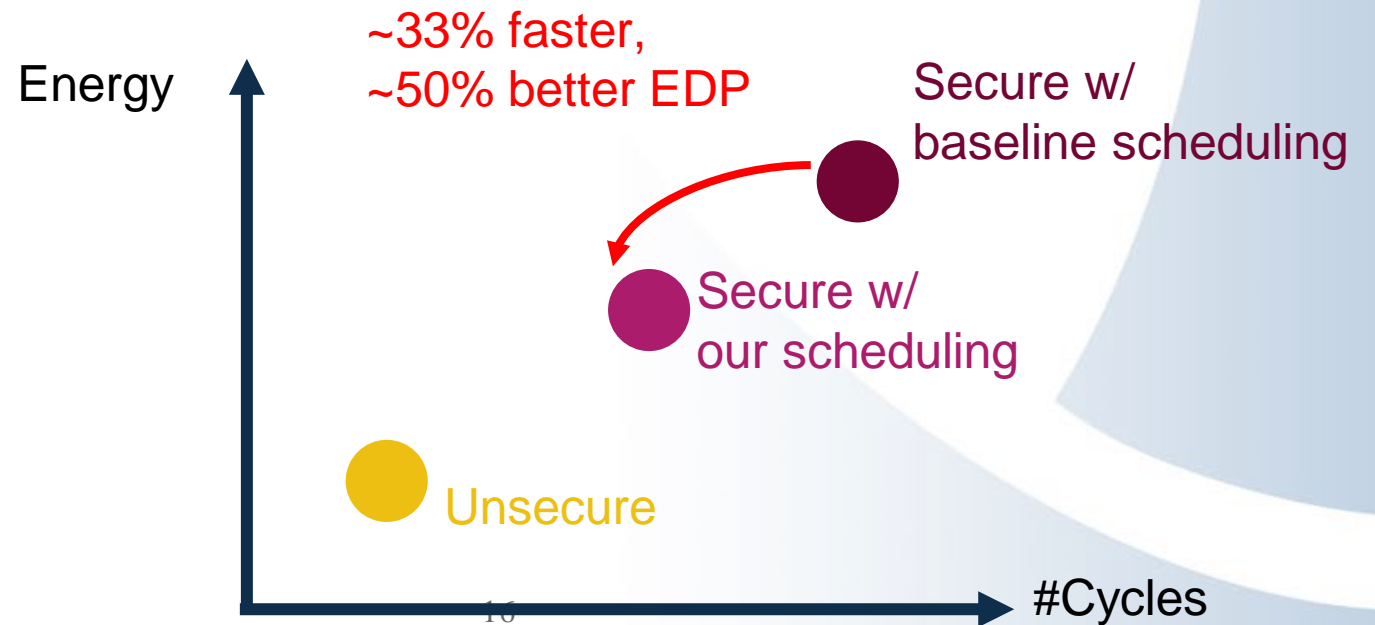


SecureLoop: Trusted Execution Environments (TEEs) on Accelerators

- SecureLoop: tool for design space exploration of TEEs on diverse DNN accelerators
 - Mathematical formulation of the cryptographic overhead
 - Cross-layer fine tuning optimization
- Identifies the optimal mapping of hardware blocks for the desired cryptographic operations
- PI: Mengjia Yan (MIT)



- 1) Encrypting/Decrypting Data Traffic
- 2) Compute Hashes for Integrity Check

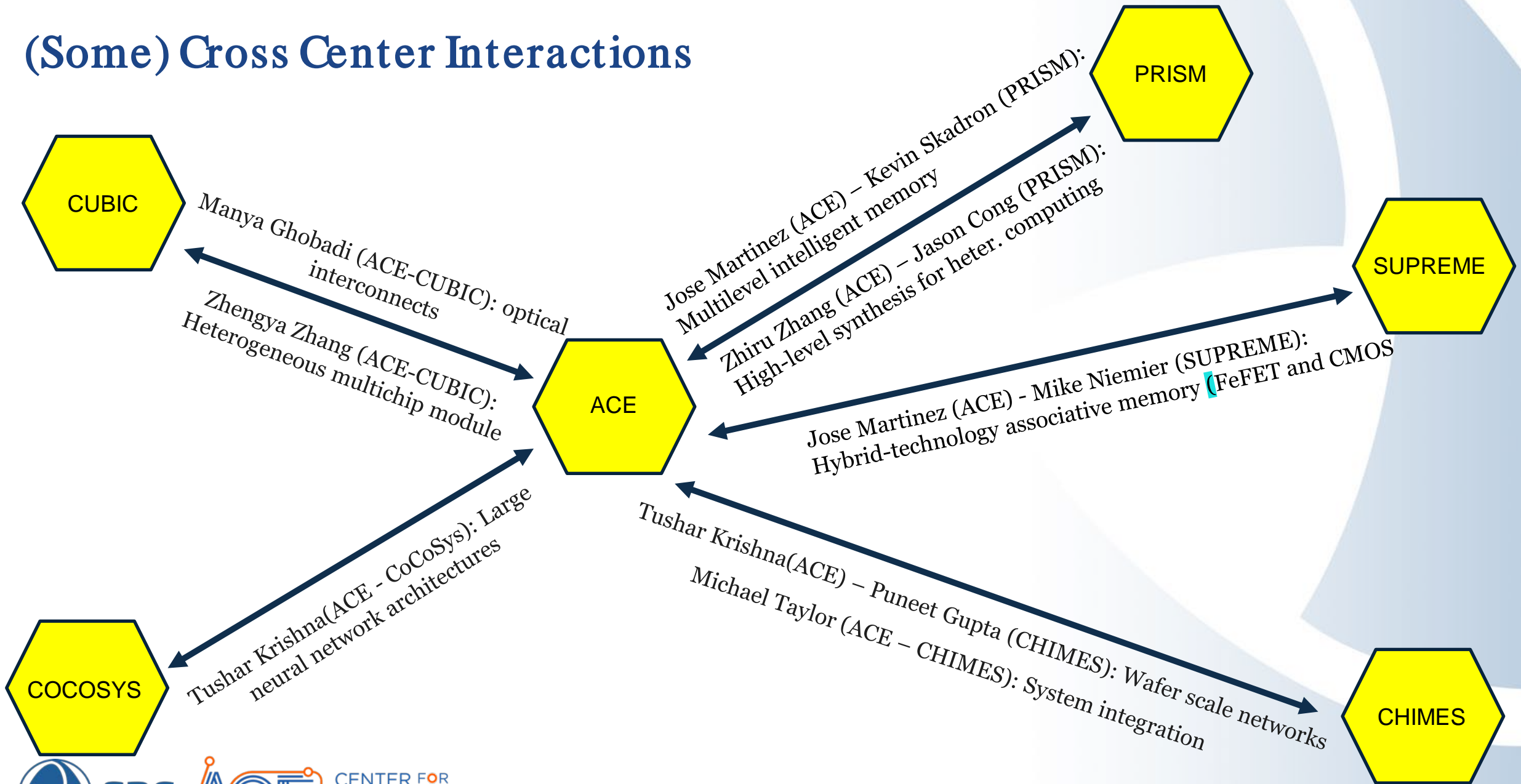


Example Impact of ACE Security Work (Q1 2024)

- Tutorial on *Learning-based Detection of Microarchitectural Attacks*
 - PI Mohit Tiwari and Postdoc Mulong Luo (UT Austin)
 - In ASPLOS'24 (April 2024)
- Tutorial on *G-QED Pre-silicon Verification Framework* for ACE and Industry Liaisons
 - PI Subhasish Mitra's team (Mo Fadiheh and Saranyu Chattopadhyay from Stanford)
 - Total of 18 participants from ACE and industry liaisons
- Best Paper Award at FPGA 2024 (March 2024)
 - PI Zhiru Zhang (Cornell), students and collaborators
 - ACE-sponsored paper "*Formal Verification of Source-to-Source Transformations for HLS*"

17

(Some) Cross Center Interactions



Ongoing Efforts

Work towards two demonstrators that integrate multiple innovations

Deepen existing connections with JUMP 2.0 companies and produce impactful out-of-the-box results

Robust student engagement with JUMP 2.0 member companies in the form of mentorship, collaboration, and internships



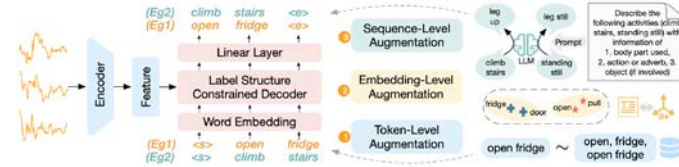
ACE Annual Review, October 4-5, 2023

END

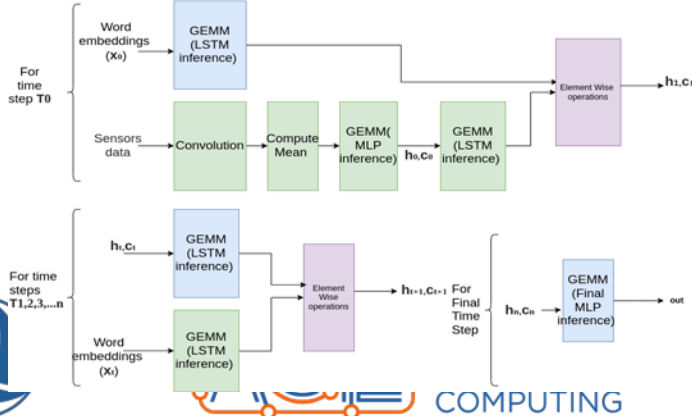
Architecture/algorithm co-design for Edge ML (PI Gupta)

- Efficient neural network algorithms for edge-based applications (algorithm benchmark planned for May 2024)
 - Small Conv-LSTM-based neural networks incorporated contextual knowledge for enhancing human activity recognition
- A Conv-LSTM engine with on-the-fly reconfigurable dataflow mapping for accelerating ML inference (architecture benchmark planned for Sep 2024)
 - Optimal systolic arrays and SIMD array synthesis for latency reduction on parallel paths
- PI: Rajesh Gupta

Contextual Knowledge-Guided Neural Network Algorithms for Human Activity Recognition



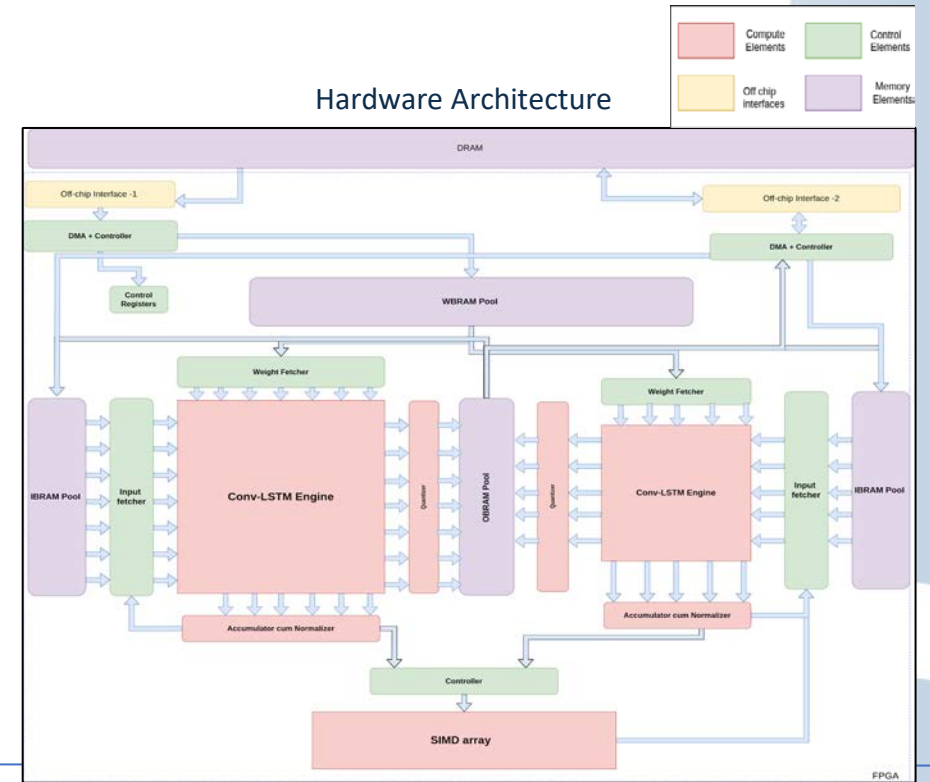
Different stages of Conv-LSTM Inference



COMPUTING

Goal: have algorithm and architecture benchmarks for edge ML. Show how they can be accelerated for distributed sensing applications.

Hardware Architecture




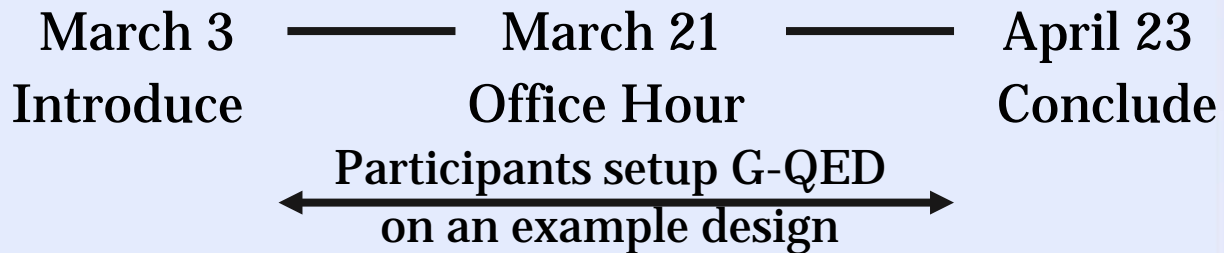
G-QED Verification for Large Evolvable Hardware Accelerators

Goal: Design-for-verification for Billion gate designs

G-QED (Sound & Complete)

- ❑ Drastic benefits (industrial AI chips for cars)
- ❑ Handles up to <0.5M gate designs

Hands-on G-QED tutorial at 



Participants:    

Scaling G-QED (In progress)

- ❑ Unique G-QED enabled abstraction
 - Complete ✓ overapprox. of original design
 - Sound ? false fails theoretically possible

Multiple hardware accelerators explored:
No false fails for “loosely-coupled” sub-designs

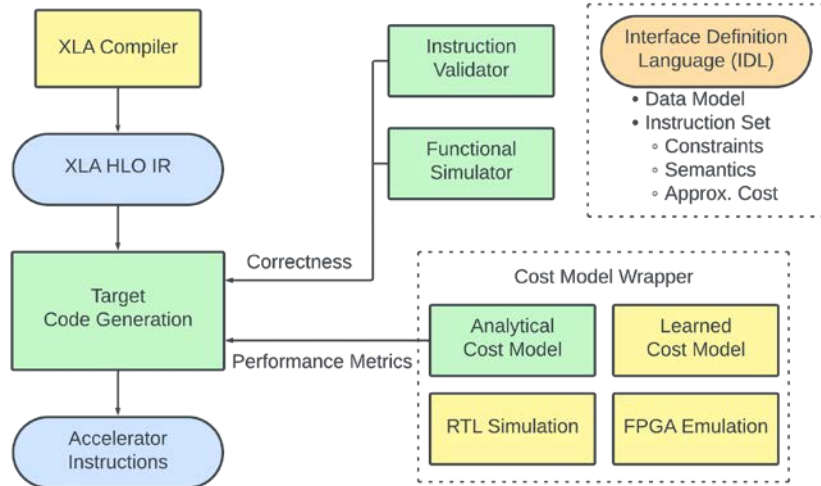
Design guidelines for soundness with min overhead

PIs: Subhasish Mitra, Radu Teodorescu, Mengjia Yan, Rajesh Gupta

Liaisons: ARM, IBM

ACT: Accelerator Compiler Toolkit

- A highly-retargetable compiler for tensor accelerators of different technologies
 - Unified Interface Design Language (IDL) to specify data-models and instruction set descriptions
 - Automatic generation of fast functional simulators, instruction validators and cost models -> completed
 - Code generation for multiple tensor accelerator platforms (analog, dataflow) -> Oct 2024
- PIs: Charith Mendis



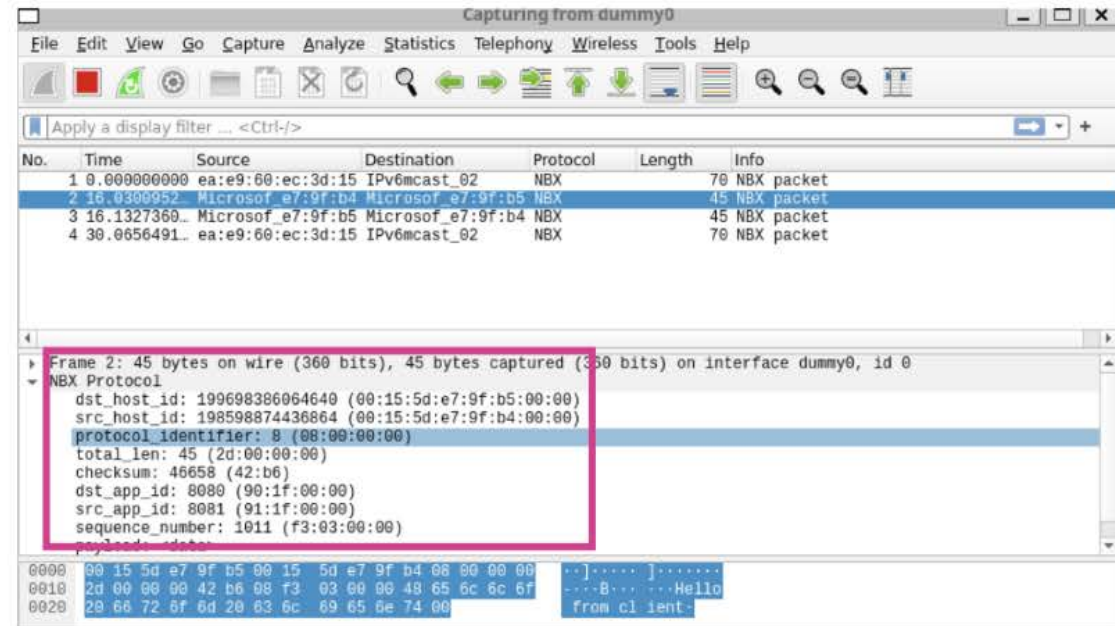
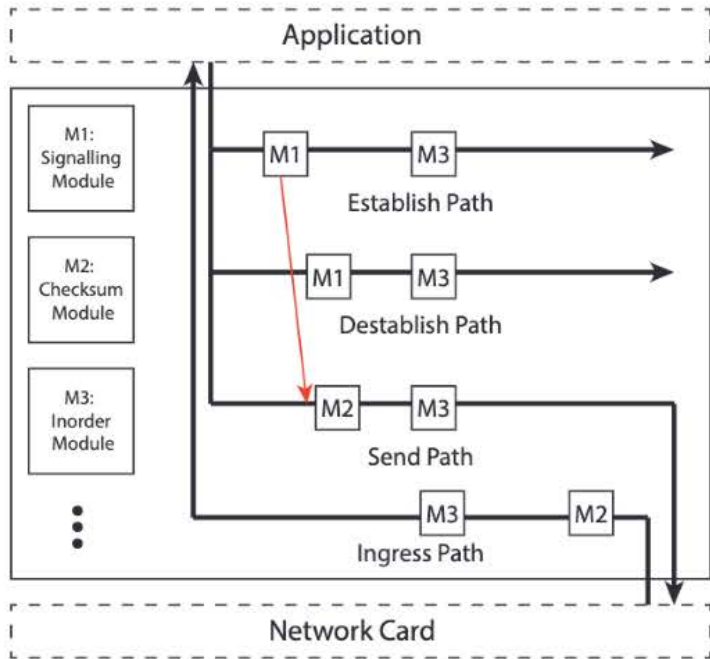
Goal: Evolvable compilation infrastructure for different ACE center and other hardware platforms that perform tensor computations.

```
1 # Instruction Set
2 ...
3 [read_weights] (addr)
4 ## Constraints
5 assert addr >= 0;
6 assert occupancy_ < 4;
7 ## Cost
8 cost 1024;
9 ## iSemantics
10 %In:65536xi8 <- $hbm[addr:addr+65536];
11 %Out:1x256x256xi8 = reshape(%In);
12 %Out:1x256x256xi8 -> $fifo[push_:push_+1];
13 ## Named-Register Update
14 update occupancy_ = occupancy_ + 1;
15 update push_ = (push_ + 1) % 4;
16 update pop_ = (pop_ == -1) ? 0 : pop_;
17 ...
```

IDL definition of a TPUv1 [16] instruction

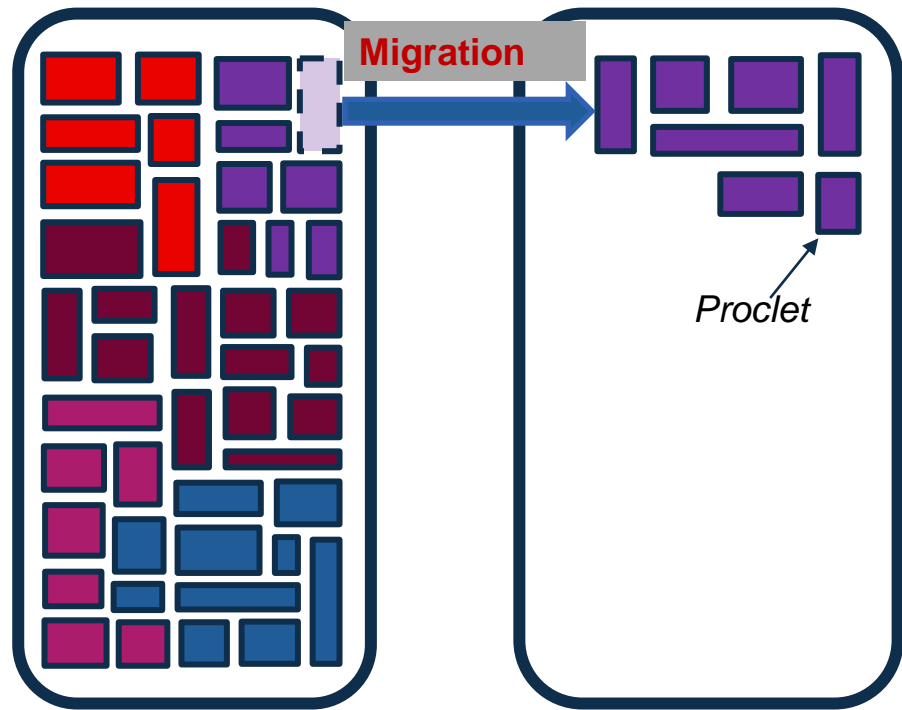
Manya: An Evolvable Network Stack

- PI: Manya Ghobadi

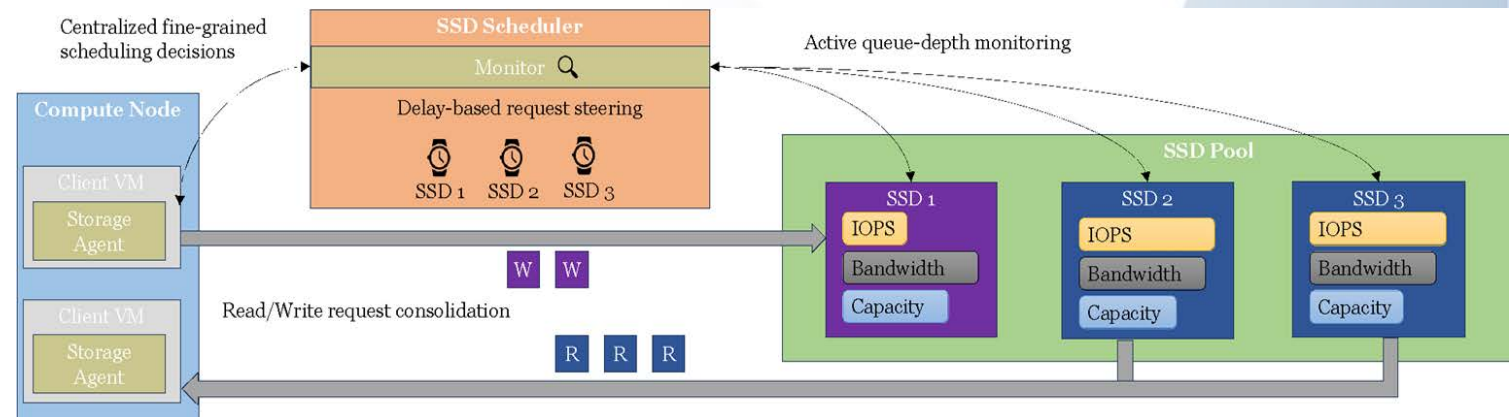


Hardware is Massively Underutilized in Cloud: Self Balancing Resources (PI Belay)

- Fast migration to rebalance compute and memory across machines [NSDI 2023]
- PIs: Adam Belay, Minlan Yu



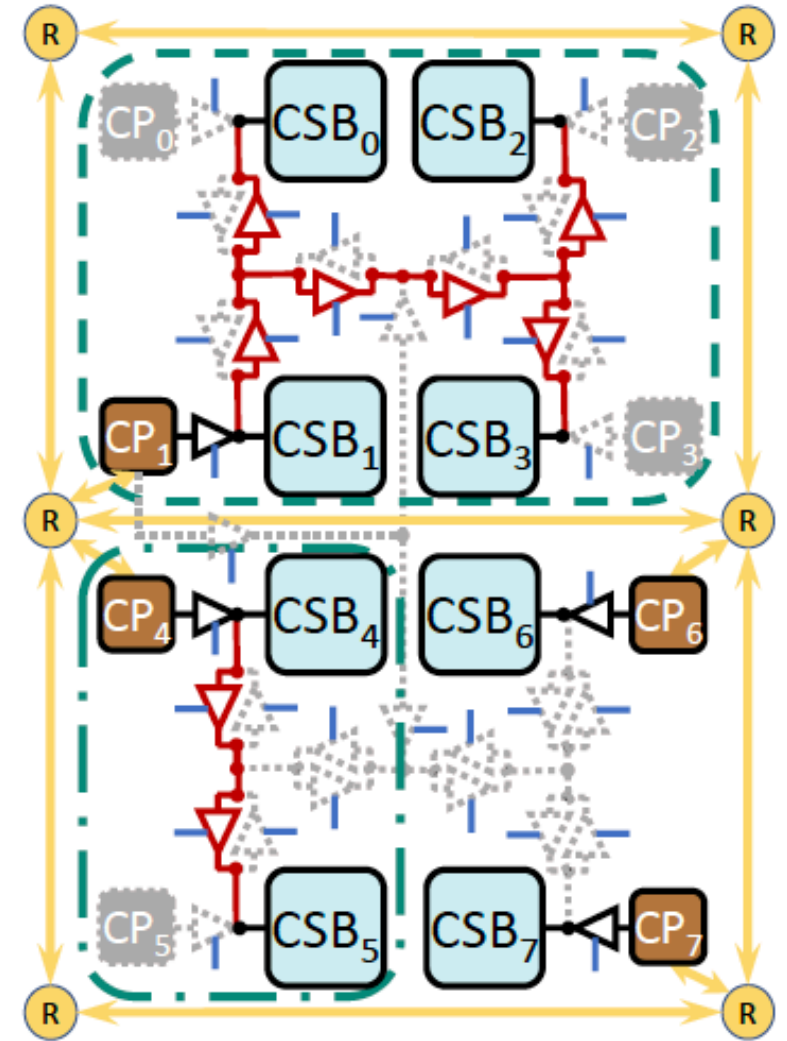
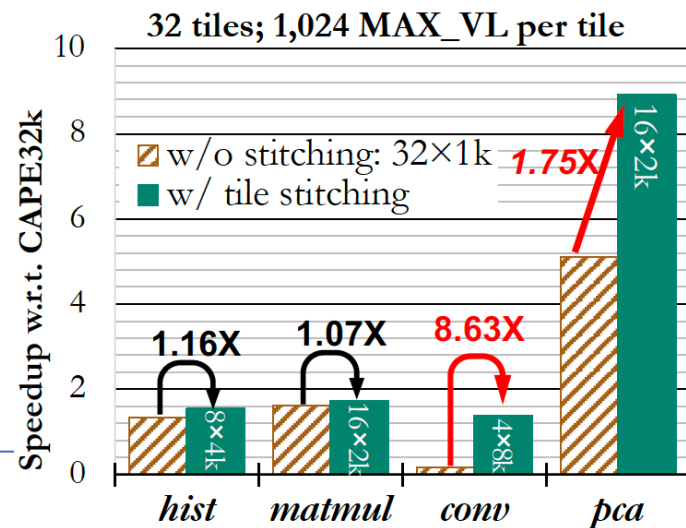
- **Next:** Make other resources self balancing → Flash SSDs
- Shard across pool of flash SSDs → poor performance
 - Hotspots
 - Heterogeneity
 - Read/Write interference
- *Sandook*: Rapid request steering to rebalance resource use



VersaTile: Composable IMS Blocks

J. Martínez (Cornell)

- Composable IMS processing array
 - CAM-based associative processor
- “Stitch” together larger IMS engines dynamically
 - Tiled Compute-Storage Block (CSB)
 - Shared control processor (CP)
 - Reconfigurable links for command distribution
- Adaptively exploit thread- and data-level parallelism

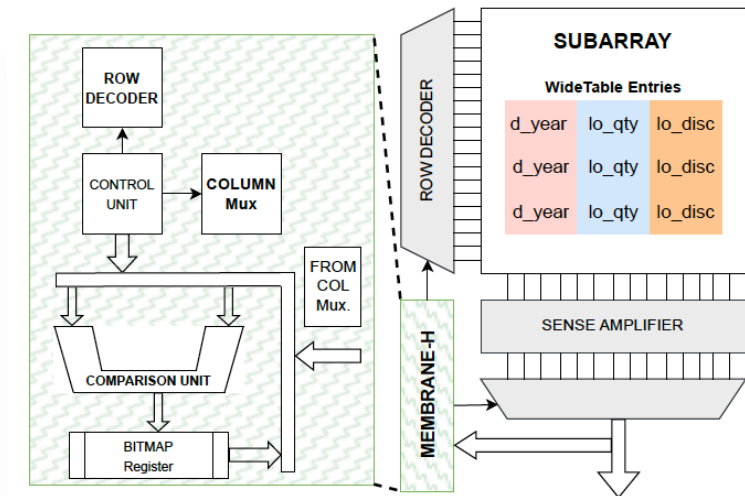
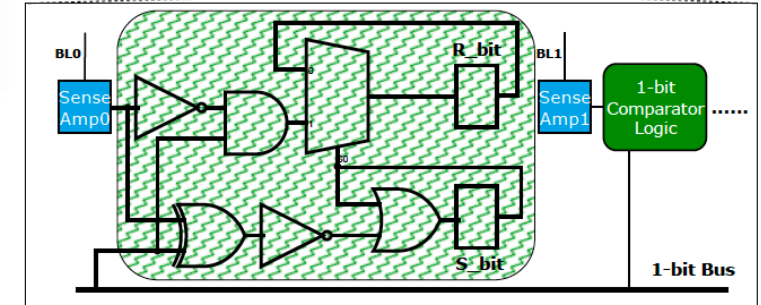
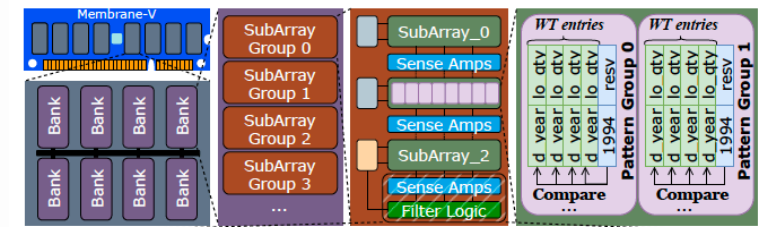
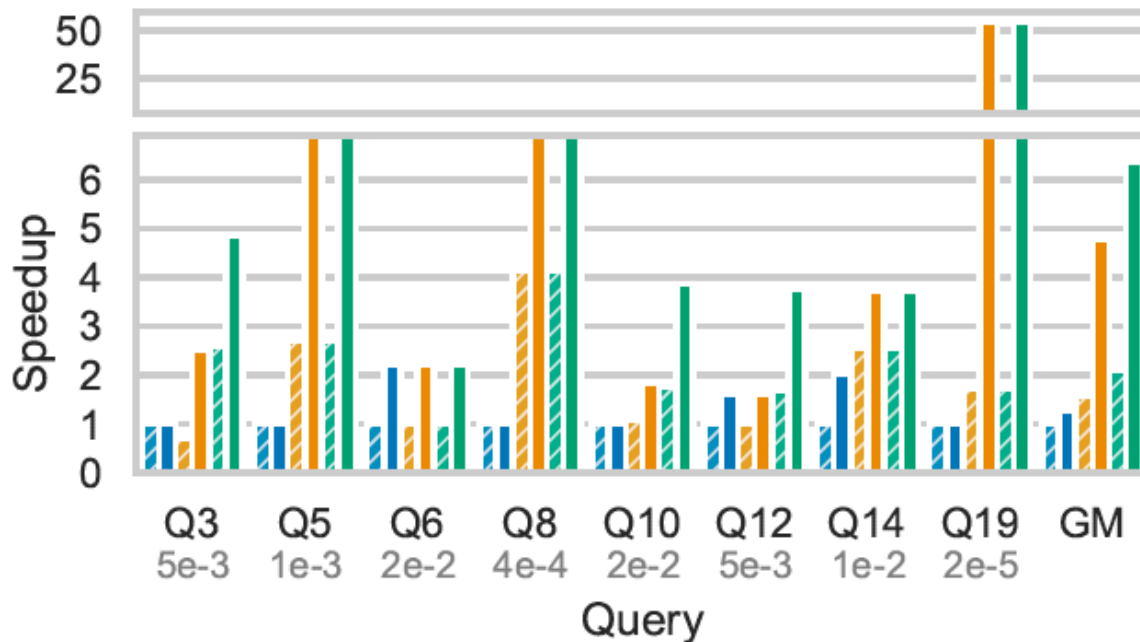


A quad-super-unit (upper), a dual-super-unit (lower-left), and two independent APs.

Membrane: Cooperative CPU+PIM Exec for OLAP

J. Martínez (Cornell)

- Join operations not naturally fit for in-DRAM computation
 - CPU-side software pre-join/denormalization



Telepathic Data Centers: Replacing IP with CXL

S. Swanson (UCSD)

- Current data centers run conventional network (IP, RDMA, etc.)
- CXL brings large-scale shared-memory
- We are building shared-memory containers
 - By-reference semantics
 - Zero-overhead sharing
- Challenges
 - Isolation
 - Security
 - Management
 - Graceful fallback to RDMA
- Results
 - Near-function call latency
 - Improved end-to-end latency for large applications
 - More natural programming model

