Semiconductor Research Corporation

CENTER FOR EVOLVABLE COMPUTING

ACE Center for Evolvable Computing

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

Director:
Josep Torrellas (UIUC)

Assistant Director:
Minlan Yu (Harvard)

JUMP2.0 Center on Systems and Architectures for Distributed Compute

What Motivates ACE

Personalized information summarization and creation (with ML)

- **Scalability:** Small queries that require little computation and data from locations
- **Bandwidth:** Download/transfer large streams of data in the cloud will take too long

Extended reality with real-time 3D immersion

- **Latency:** Maintain 15ms Motion-to-Photon (MTP) latency in the presence of increasing computational
- **Energy-Efficient Computation:** Need more compute in the Head Mounted Display (HMD)

Automation and control

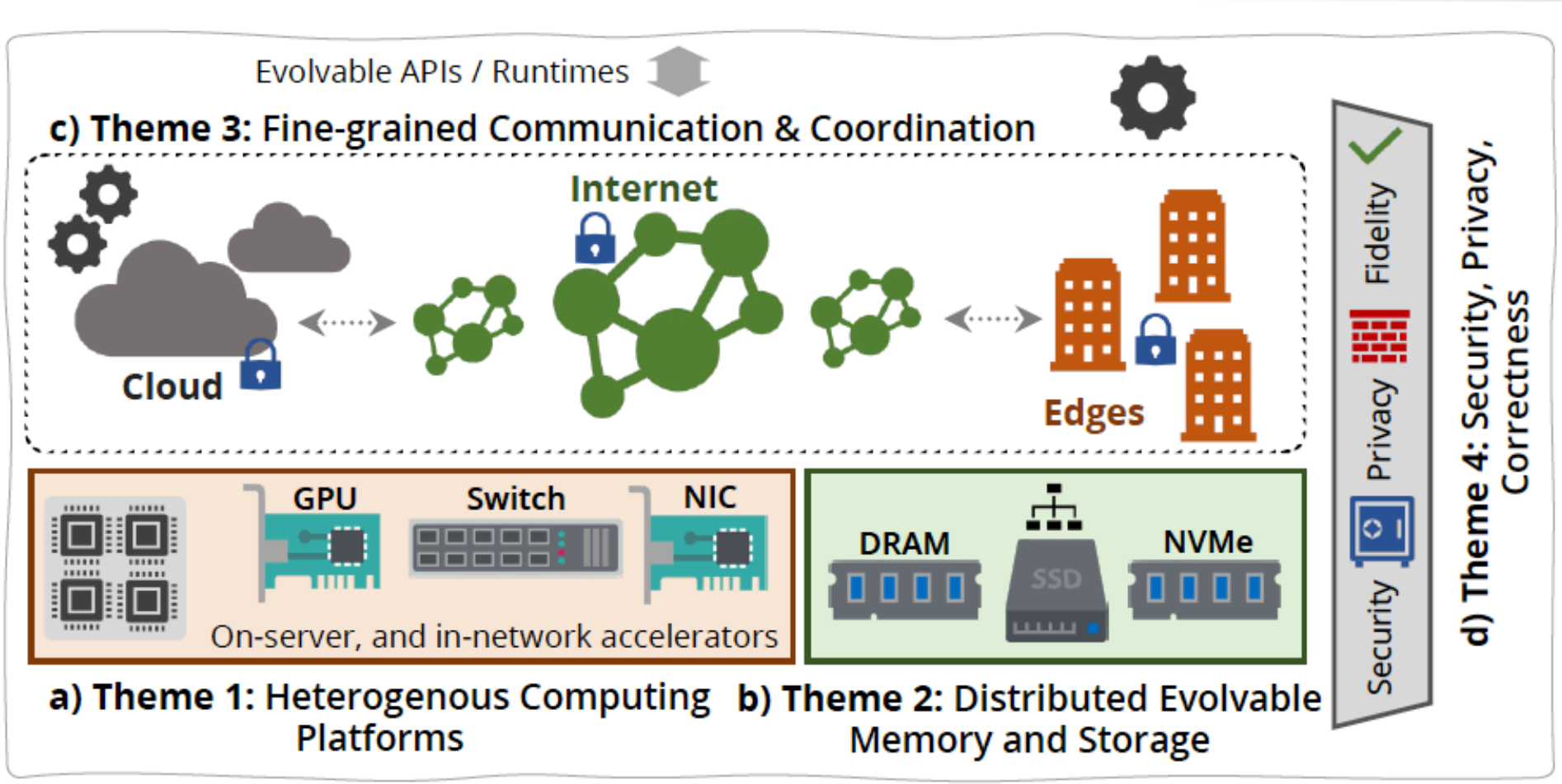
- **Efficient Event Processing:** Hardware units at the edge must face the sudden arrival of discrete events.
- **Security:** The computing and communication resources of autonomous agents are shared by multiple users.

SRC ACE CENTER FOR EVOLVABLE COMPUTING

Distributed Computing of 2030+

- Process vast swaths of data for insights in a timely manner
- Overriding constraint: curtail energy consumption
- Compute infrastructure: hierarchy of compute centers: edge → geo-dist megadatacenters
- Each compute center, to minimize energy:
  - Large number of heterogeneous specialized compute units (i.e., accelerators)
  - Small tasks ship computation to where data is

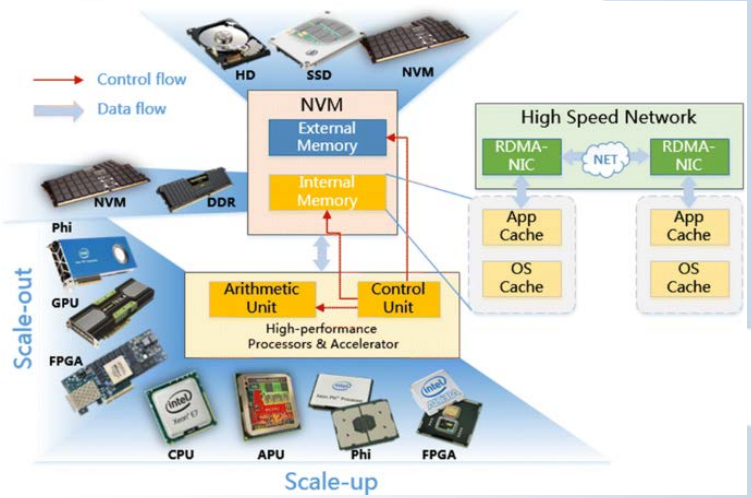
ACE: Planet-scale Distributed Computing Infrastructure for 2030+



Evolvable Computing

Evolvable Computing is what ACE is about

- Hardware/Software: Design accelerator hardware, memory structures, communication stacks, and security mechanisms for extensibility and composability
  - Standard and composable interfaces
  - Easily assembled into systems of different form factors
  - Survive upgrades of their external environments
  - Easily replaced by (and co-exist with) a next-generation design
- Applications: Built as collections of functions that abstract details
  - What accelerator they will run on
  - What communication mechanisms they will use



Organized the Team to Identify Paradigm-Changing Solutions

- **Theme 1: Heterogenous Computing Platforms**

  Gupta, Kozyrakis, Krishna, Martinez, Mendis, Shahbaz, Taylor, Teodorescu, Torrellas, Zhengya Zhang, **Zhiru Zhang**

- **Theme 2: Distributed Evolvable Memory and Storage**

  Alian, Krishnamurthy, **Martinez**, Swanson, Zhiru Zhang

- **Theme 3: Fine-grained Communication and Coordination**

  Alian, Belay, Ghobadi, Kozyrakis, Krishna, Krishnamurthy, Mendis, Shahbaz, Torrellas, **Yu**

- **Theme 4: Security, Privacy, and Correctness**

  Belay, Gupta, Kozyrakis, Mitra, Suh, **Teodorescu**, Tiwari, Zhiru Zhang

- **Theme 5: Demonstrators**

  Abdelzaher, Alian, Belay, Ghobadi, Gupta, Kozyrakis, Krishna, Krishnamurthy, Martinez, Mendis, Mitra, Shahbaz, Swanson, Suh, Taylor, Teodorescu, Tiwari, Torrellas, Yu, **Zhengya Zhang**, Zhiru Zhang

Semiconductor Research Corporation

ACE CENTER FOR EVOLVABLE COMPUTING

Theme 1: Heterogeneous Computing Platforms

Storage

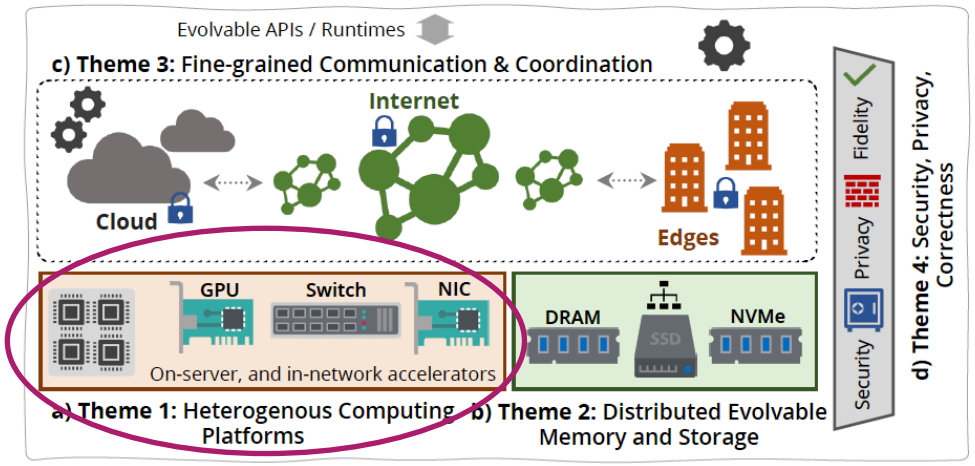
Communication

Security

Demonstrators

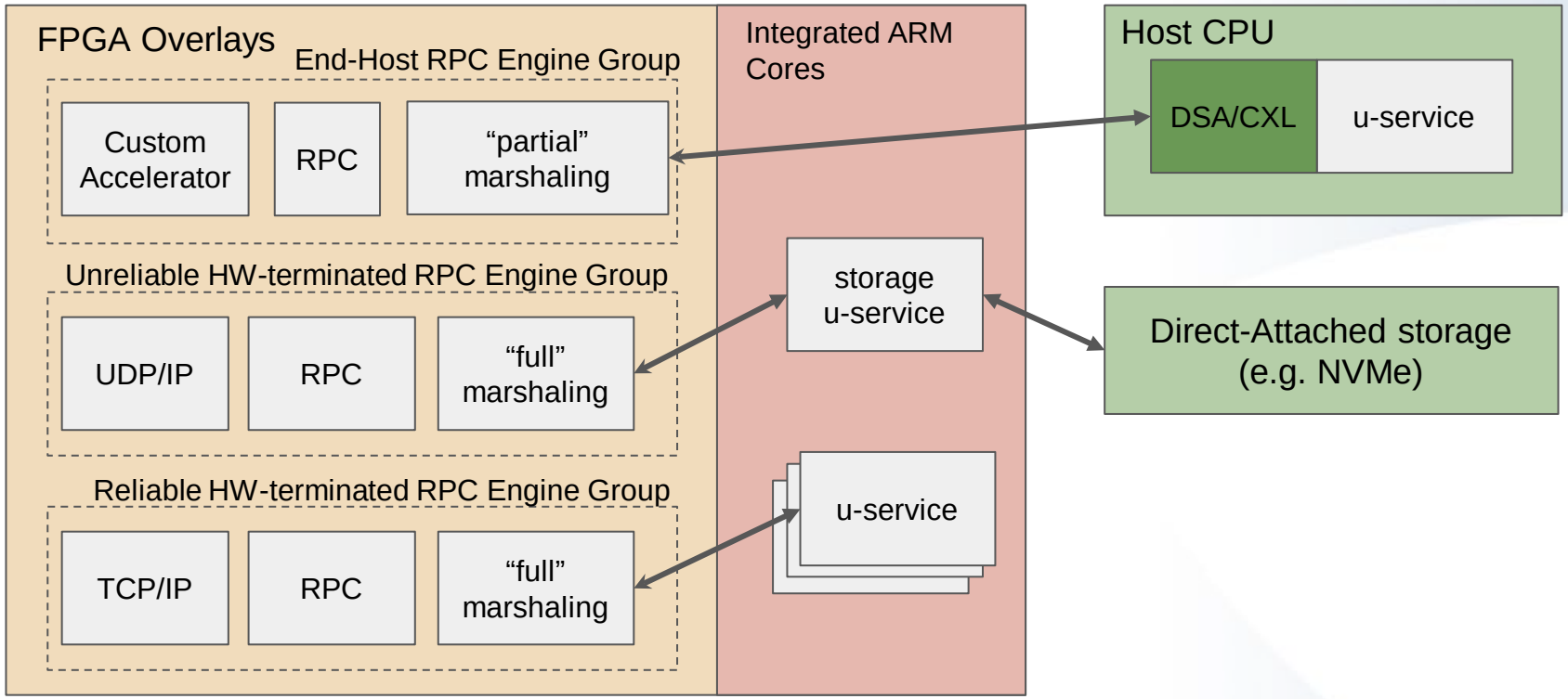
Theme 1: Heterogeneous Computing Platform

- Myriad of heterogenous hardware accelerators in all data centers
- New methodologies to generate, deploy and, in seconds reconfigure accelerators
- Accelerators organized into ensembles, within and across datacenters
- Smart compilers/runtimes pick groups of accelerators from an ensemble and map/schedule apps
- Ensembles are spatially and temporally shared by multiple tenants securely
- General-purpose cores are specialized for different workloads classes



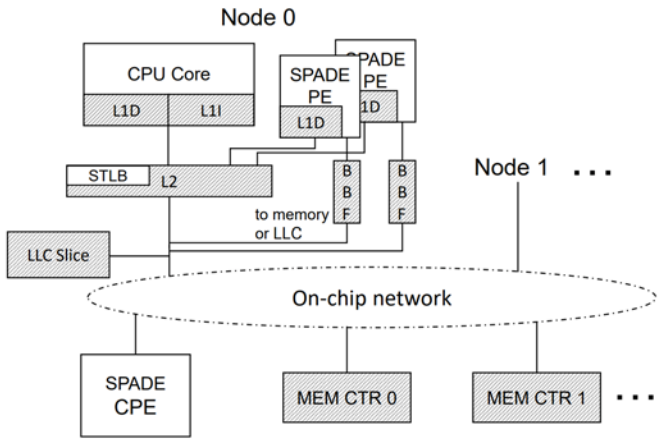
SRC ACE CENTER FOR EVOLVABLE COMPUTING

Reconfigurable Acceleration of Cloud Services

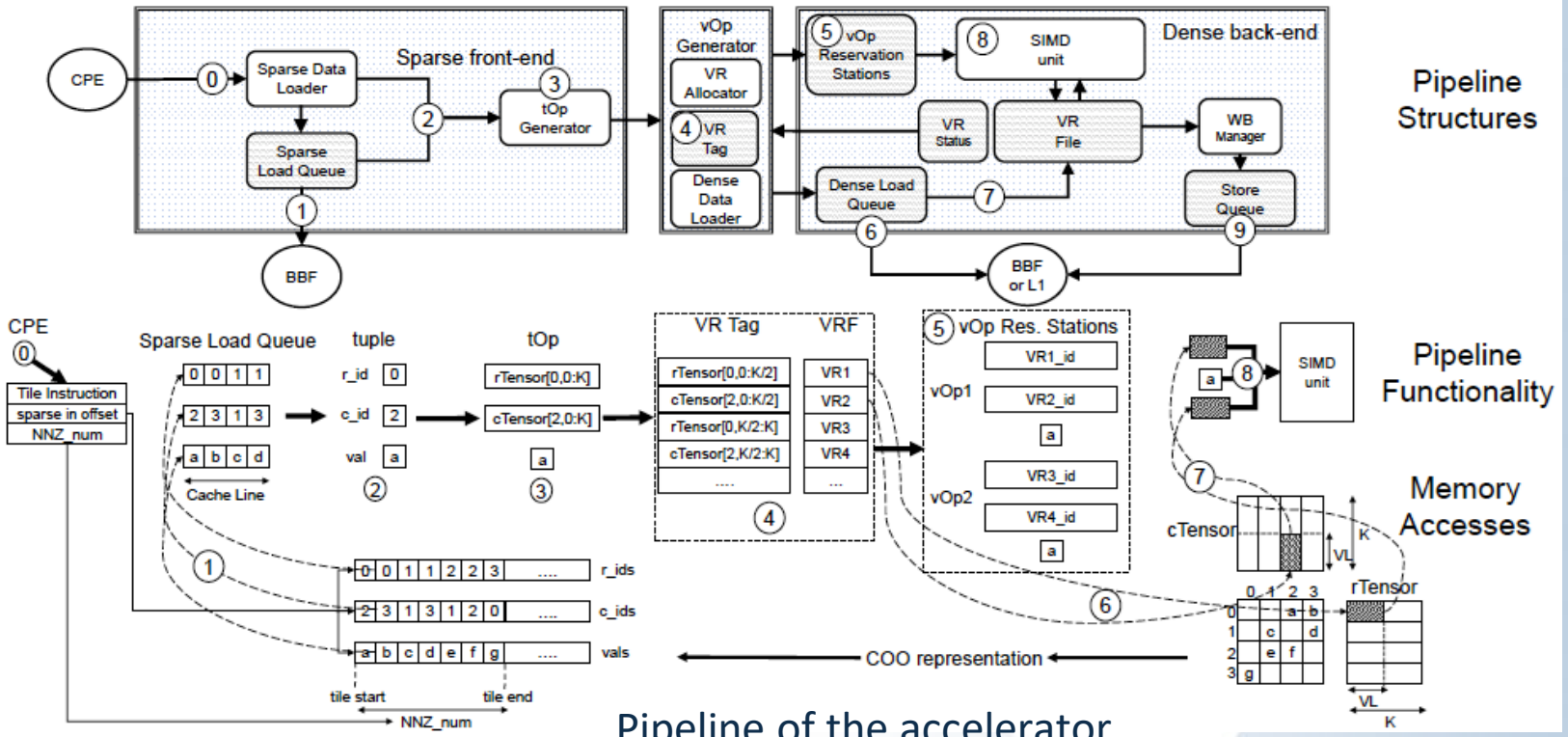


- FPGA-based overlay architectures to transparently accelerate distributed cloud services
- Offload "datacenter tax" operations to distributed reconfigurable accelerators:
  - Communication, memory allocation, compression, serialization-deserialization, and encryption

Flexible & Scalable Accelerator for Sparse Computations



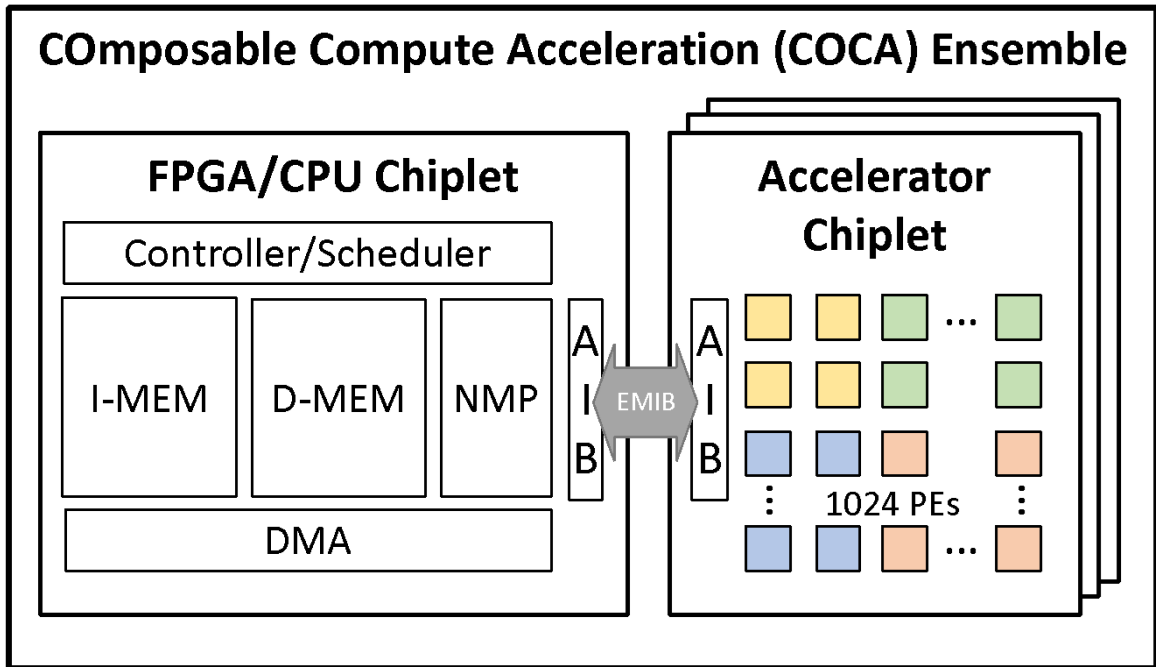
Accelerator embedded in a core



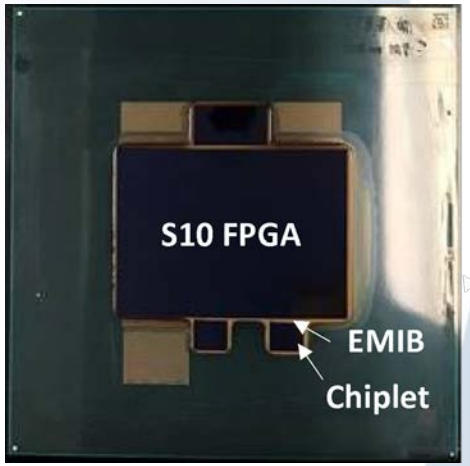
Pipeline of the accelerator

- To minimize data transfers between CPU-accelerator: embed accelerators in the CPU cache hierarchy
- To adapt to different input sparse matrices: accelerator has high-level *tile-based* ISA

COmposable Compute Acceleration (COCA)



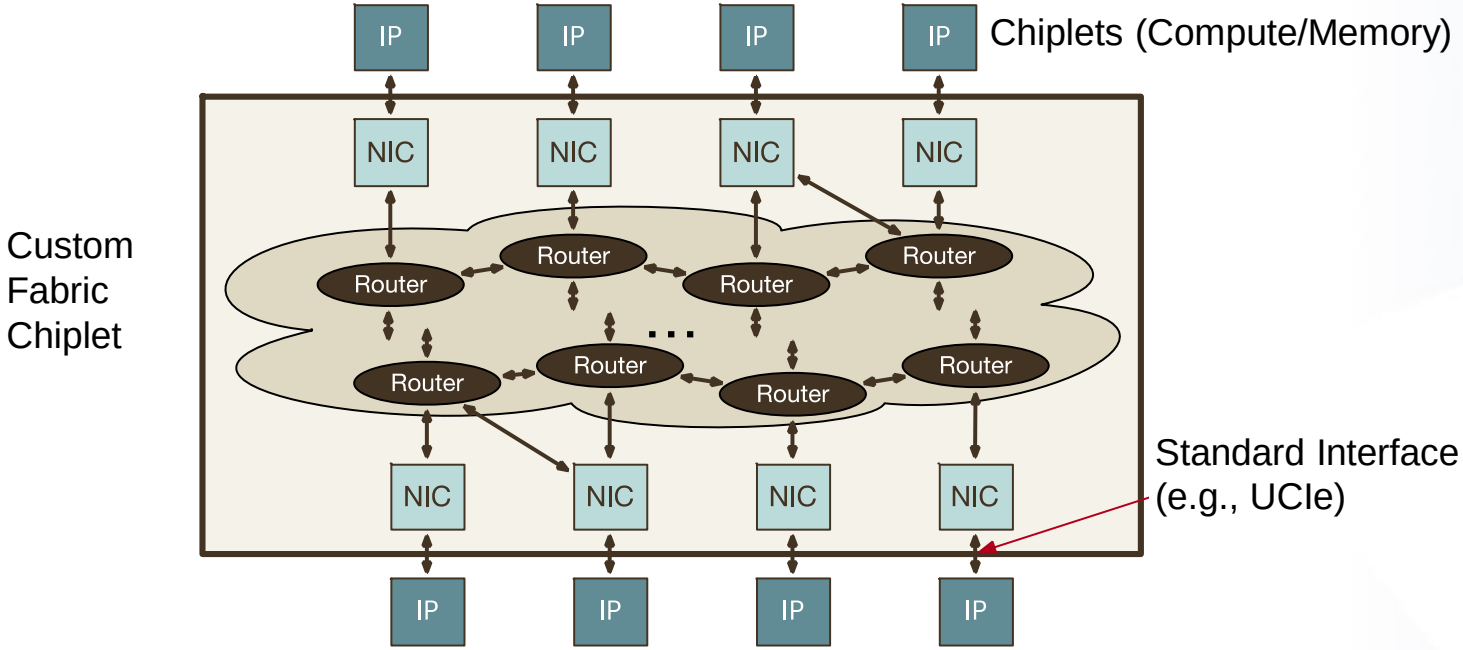
Conceptual illustration



Silicon example

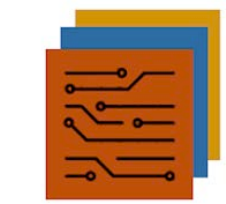
- Design a composable platform with heterogeneous chiplets
- Composed systems can evolve over time
  - Off-line: new designs may include new combinations of chiplets
  - On-line: reconfigure CGRAs and overlays, dynamically adapting to the applications running

Designing a Fabric for Scalable Connectivity

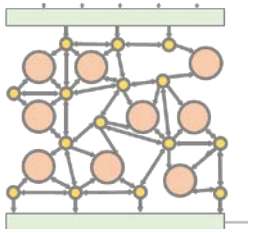


- Generator of a network-on-package architecture for heterogeneous chiplet platform
  - Generator written in Verilog / HLS / Chisel
  - Standardized interfaces for protocol layer and physical layer
  - Creates 2.5D/3D topology that is packaging-technology aware (MCM vs Interposer vs Wafer-scale)

Dataflow Mapping for Chiplet-Based Accelerators

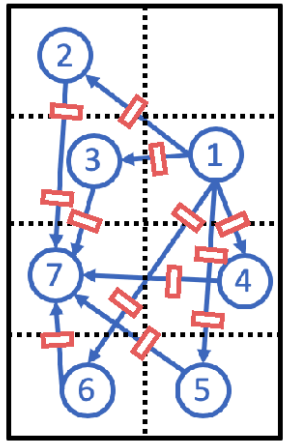


Individual kernels

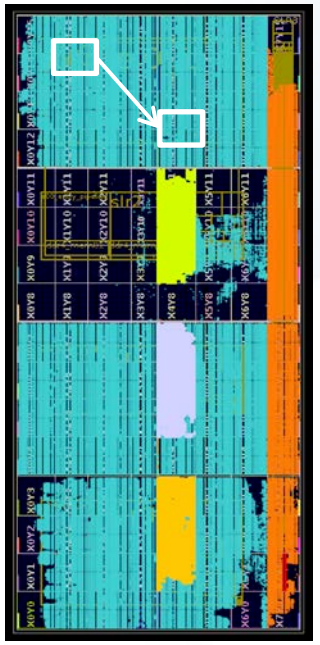


Logical topology
(dataflow graph)

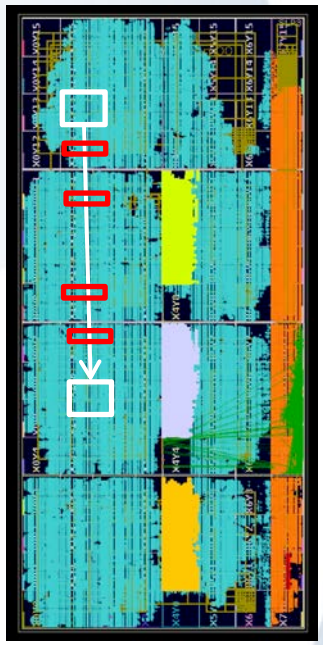
Physical-aware Mapping & Pipelining



Partitioned multi-die implementation



Default @ ~200 MHz on a multi-die FPGA
(High local congestion)



Physical-aware mapping @ 300+ MHz
(Long wire pipelined)

- Developing a tool-flow to map large dataflow designs into accelerators based on multiple chiplets
  - Language and compilation support

Semiconductor Research Corporation

CENTER FOR EVOLVABLE COMPUTING

~~Compute~~

Theme 2: Distributed Evolvable Memory and Storage

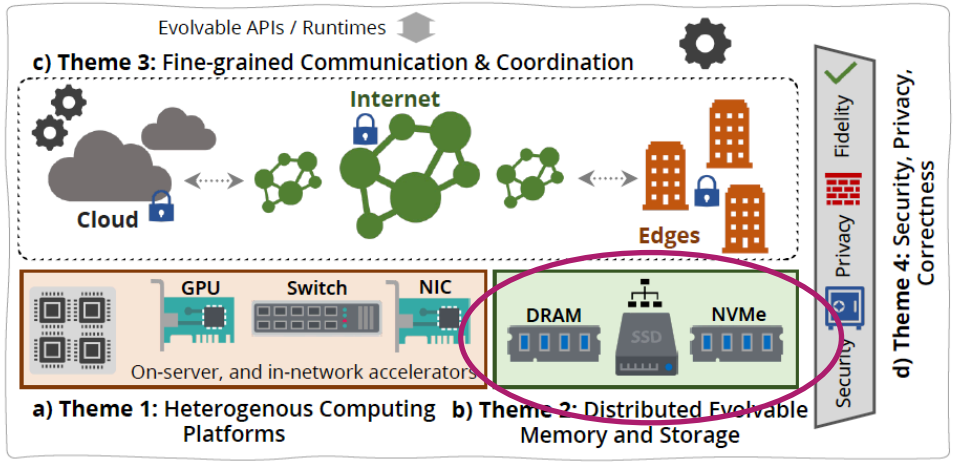
Communication

Security

Demonstrators

Theme 2: Distributed Evolvable Memory & Storage

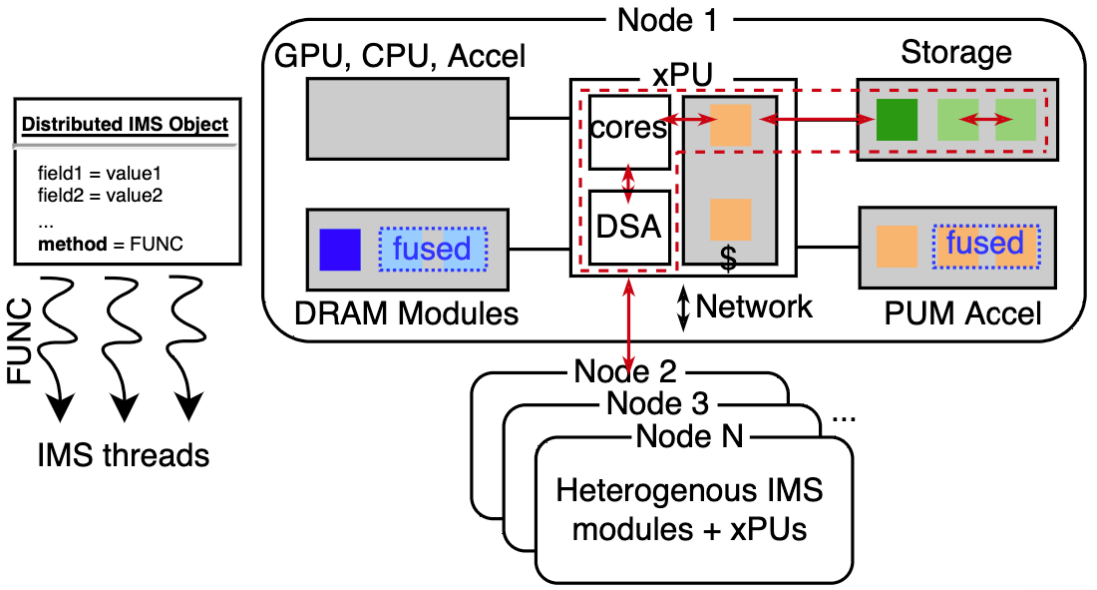
- Highly heterogenous, distributed memory and storage hierarchy
- Memory reachable by cores as local memory is expanded across an entire rack
- Handle memory wall: novel processor structures and gracefully-degrading coherence
- Abstractions allow applications to select the type of memory/storage assets needed
- Algorithms to apportion these assets among 1000s competing apps in a datacenter
- Distributed intelligent memory and storage blocks harnessed for coordinated operation



SRC ACE CENTER FOR EVOLVABLE COMPUTING

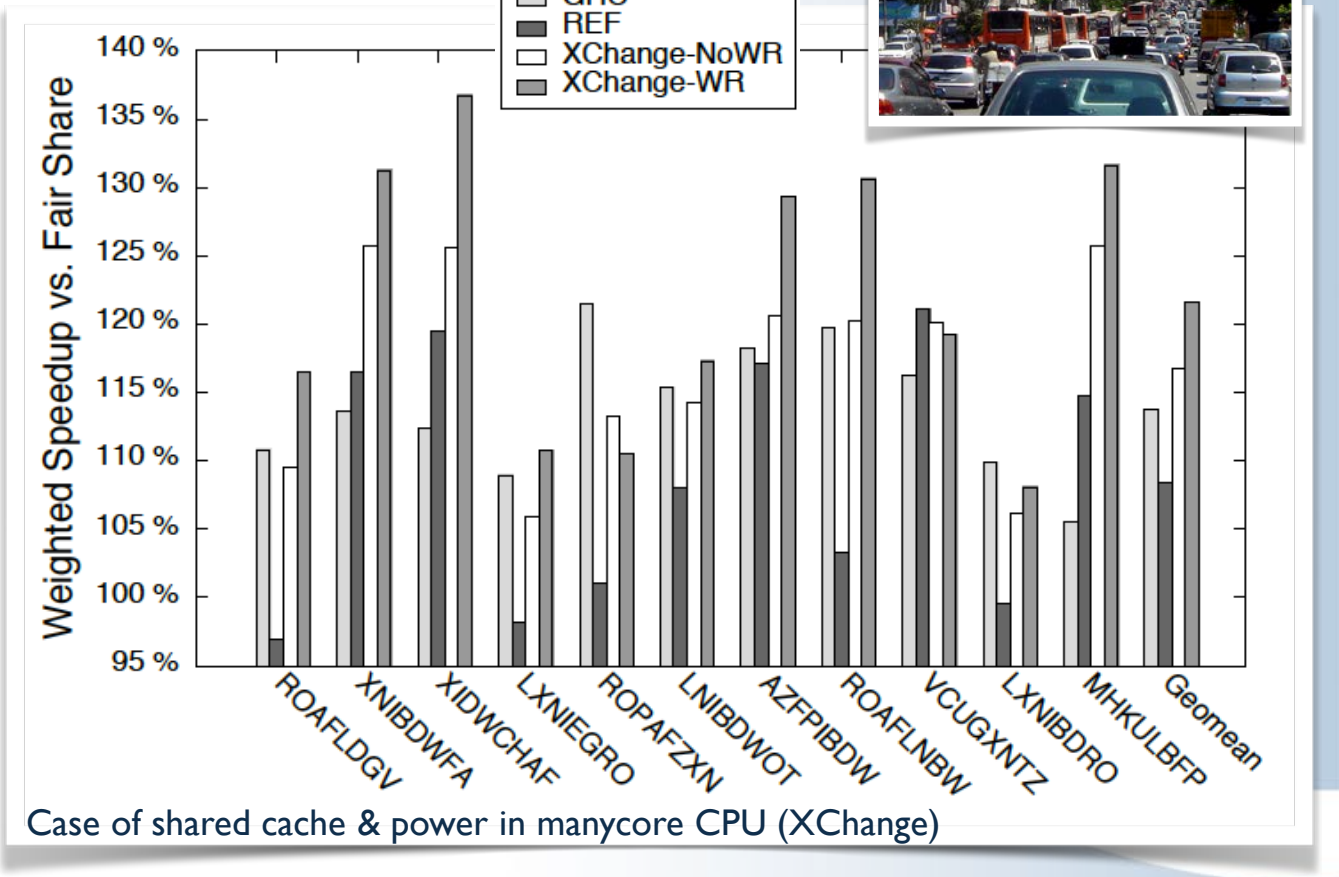
Near- and In-memory/storage (IMS) Acceleration

- Ubiquitous IMS blocks in memory hierarchy, network switches, SSDs
- Harness distributed IMS blocks to operate with coordination
- Cooperation with JUMP2.0 PRISM Center



Market-Based Distributed Resource Allocation

- Resource are disaggregated
- Billions of allocation requests
- Applications endowed with
  - Virtual currency
  - Independent resource profiler
- Resource broker reconciles supply and demand
  - Hardware resources priced dynamically according to demand
  - Applications bid according to pricing *and* own resource sensitivity
  - Seeks competitive market equilibrium



Case of shared cache & power in manycore CPU (XChange)

Semiconductor Research Corporation

CENTER FOR EVOLVABLE COMPUTING

~~Compute~~

~~Storage~~

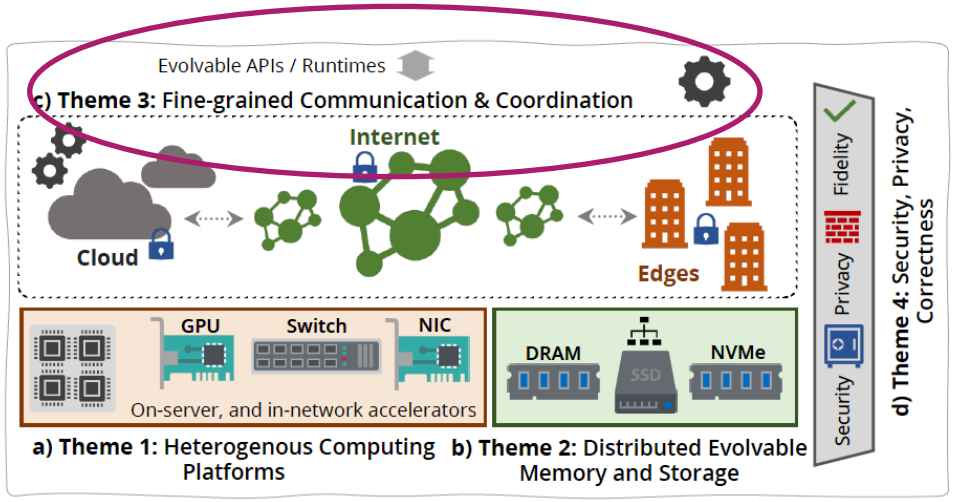
Theme 3: Fine-grained Communication and Coordination

Security

Demonstrators

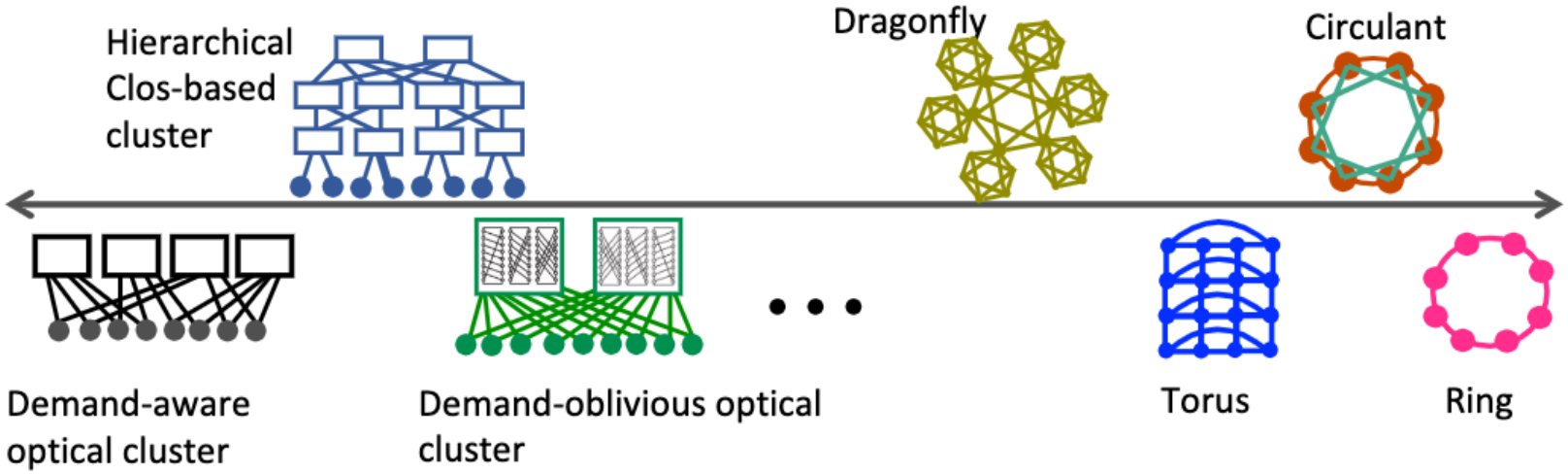
Theme 3: Fine-grained Communication and Coordination

- Million-accelerator datacenters with reconfigurable network topologies
- Flexible communication software stacks specialized to the accelerators available
- Nimble runtimes that bundle computation in tiny buckets and ship them where the data lives
- Accelerators in network switches and SmartNICs offload processor tasks
- Geo-distributed databases with Exabytes of data that distribute it at tens TB/s

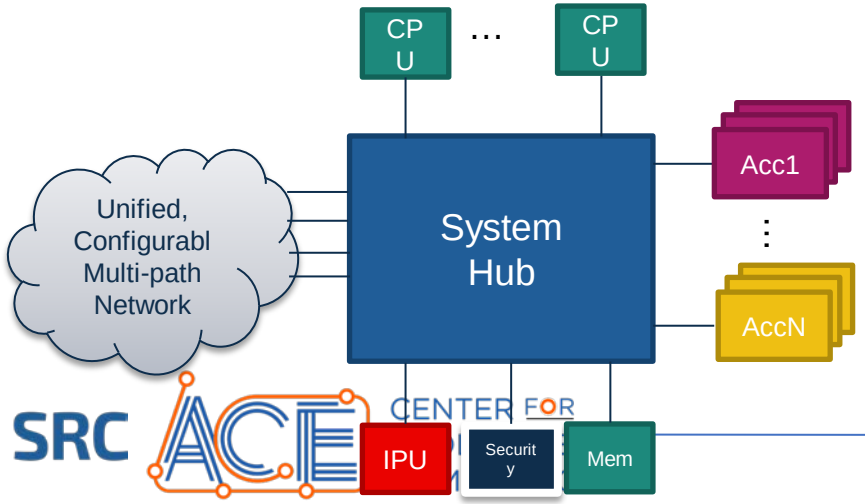


SRC ACE CENTER FOR EVOLVABLE COMPUTING

Network Topology Design for Accelerator-Rich Datacenters



- Goal: build a single physical network that gradually evolves to cover emerging workloads and newer accelerators.

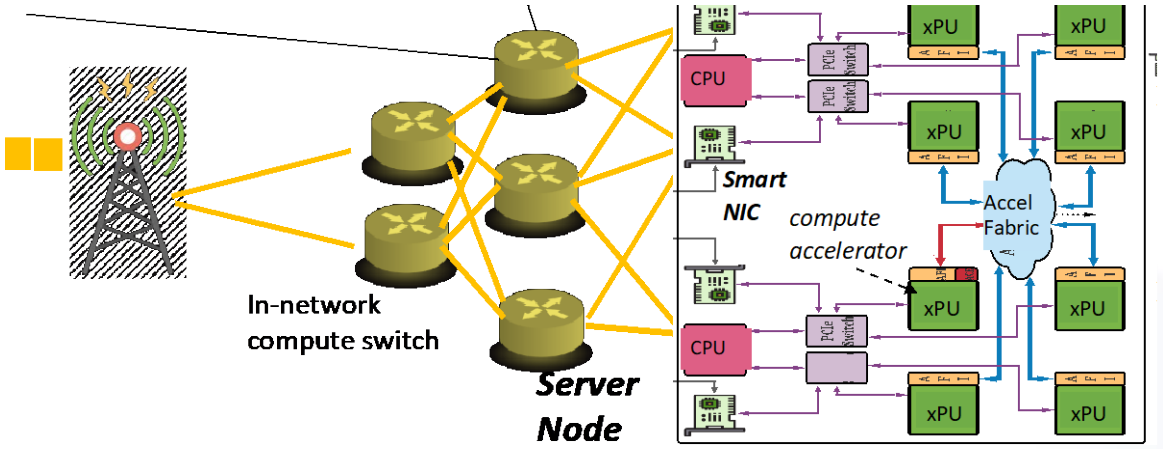


System hub:

- Bridge multiple interconnect domains within each server:
  - Memory interconnects (e.g., UPI, Infinity, NVLink, or CXL)
  - I/O interconnects (e.g., PCIe)
  - Network connections (Ethernet)

Task 3.4: In-network Computing

- Network switches and SmartNICs perform some computations more efficiently than cores
- Example: Straggler identification



Semiconductor Research Corporation

CENTER FOR EVOLVABLE COMPUTING

~~Compute~~

~~Storage~~

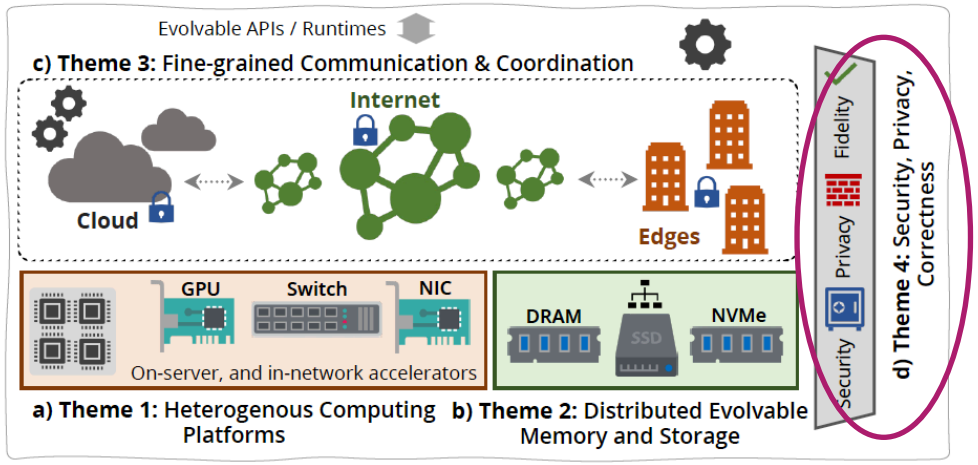
~~Communication~~

Theme 4: Security, Privacy, and Correctness

Demonstrators

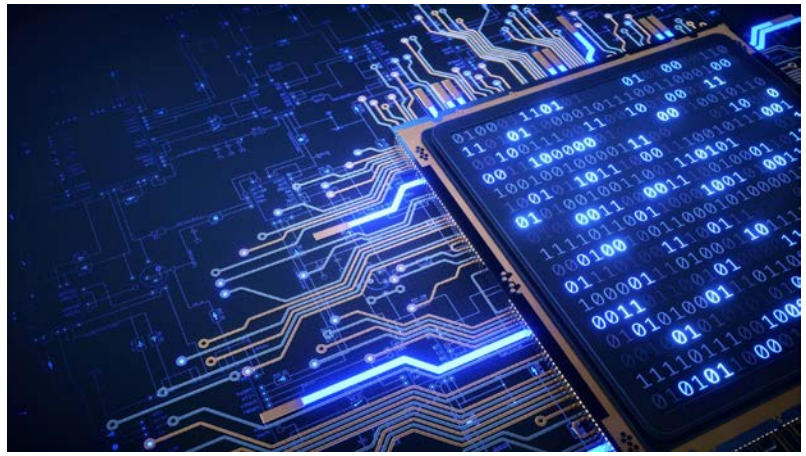
Theme 4: Security, Privacy, and Correctness

- Accelerators built from ground up with security/correctness mechanisms
- New security paradigms
- Customize trusted execution environments (TEEs) to the particular accelerators, and automatically generate them with tools
- Order-of-magnitude faster techniques to verify correctness of accelerator design



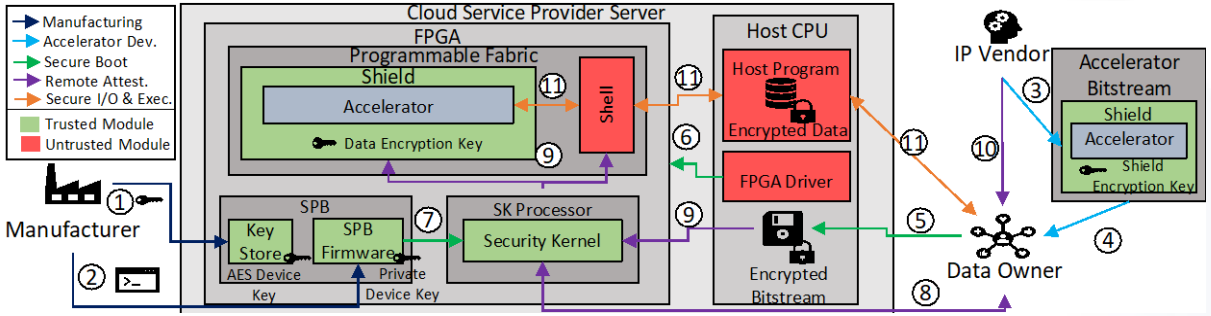
Data-centric Security

- Large distributed systems will increasingly process sensitive data and safety-critical systems
- Existing security frameworks tie security properties to users, applications, or hardware/software systems → resulted in many attacks
- We propose a new *Data-Centric* security paradigm that ties the security properties to data



Domain-specific Trusted Execution Environments (TEEs) on Heterogeneous Accelerators

- A future ecosystem rich in accelerators will need first class support for secure computing
- Trusted Execution Environment (TEE) designs will expand to a multitude of accelerators
- We are developing TEE compilers and frameworks that generate TEEs customized to accelerators, threat models, and application needs



Semiconductor Research Corporation

ACE CENTER FOR EVOLVABLE COMPUTING

~~Compute~~

~~Storage~~

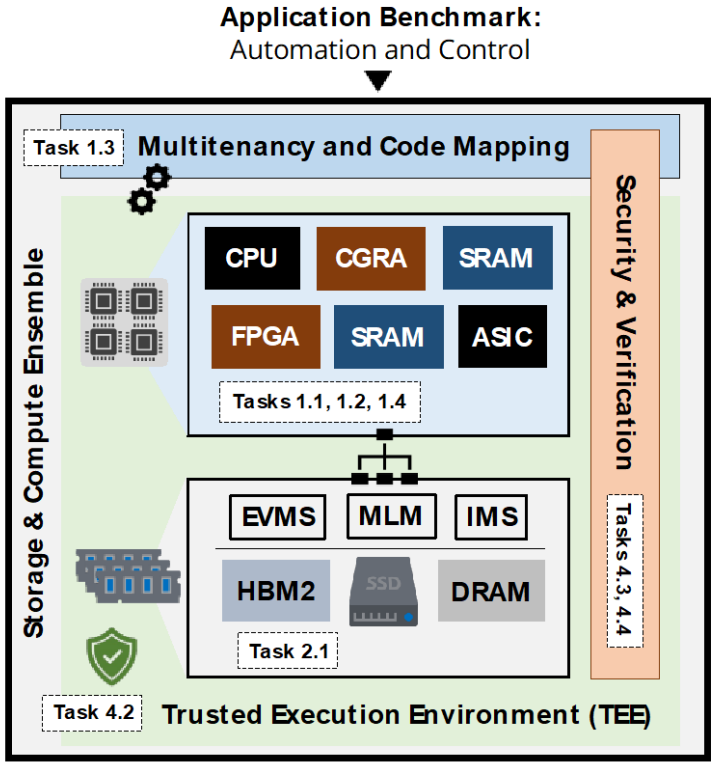
~~Communication~~

~~Security~~

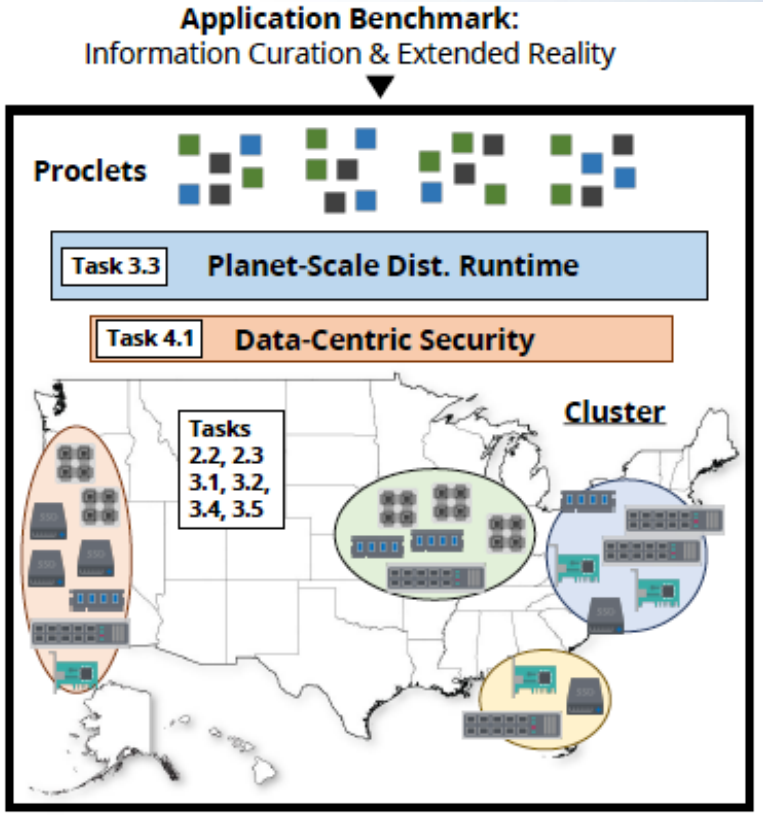
Theme 5: Demonstrators

Theme 5: Demonstrators

Design, prototype and evaluate two demonstrators to showcase combined operation of technologies

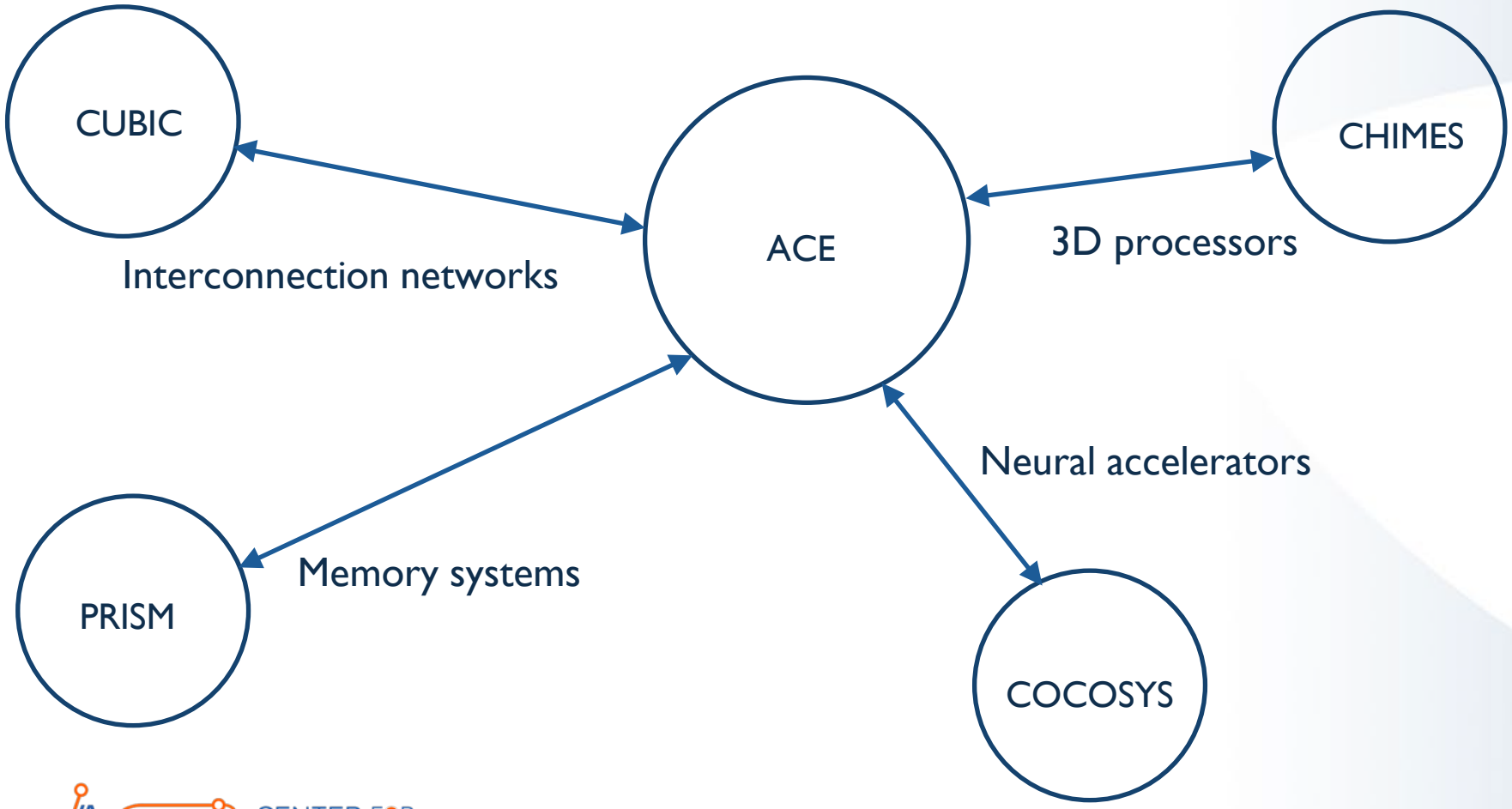


Reconfigurable Multi-accelerator Compute Ensemble

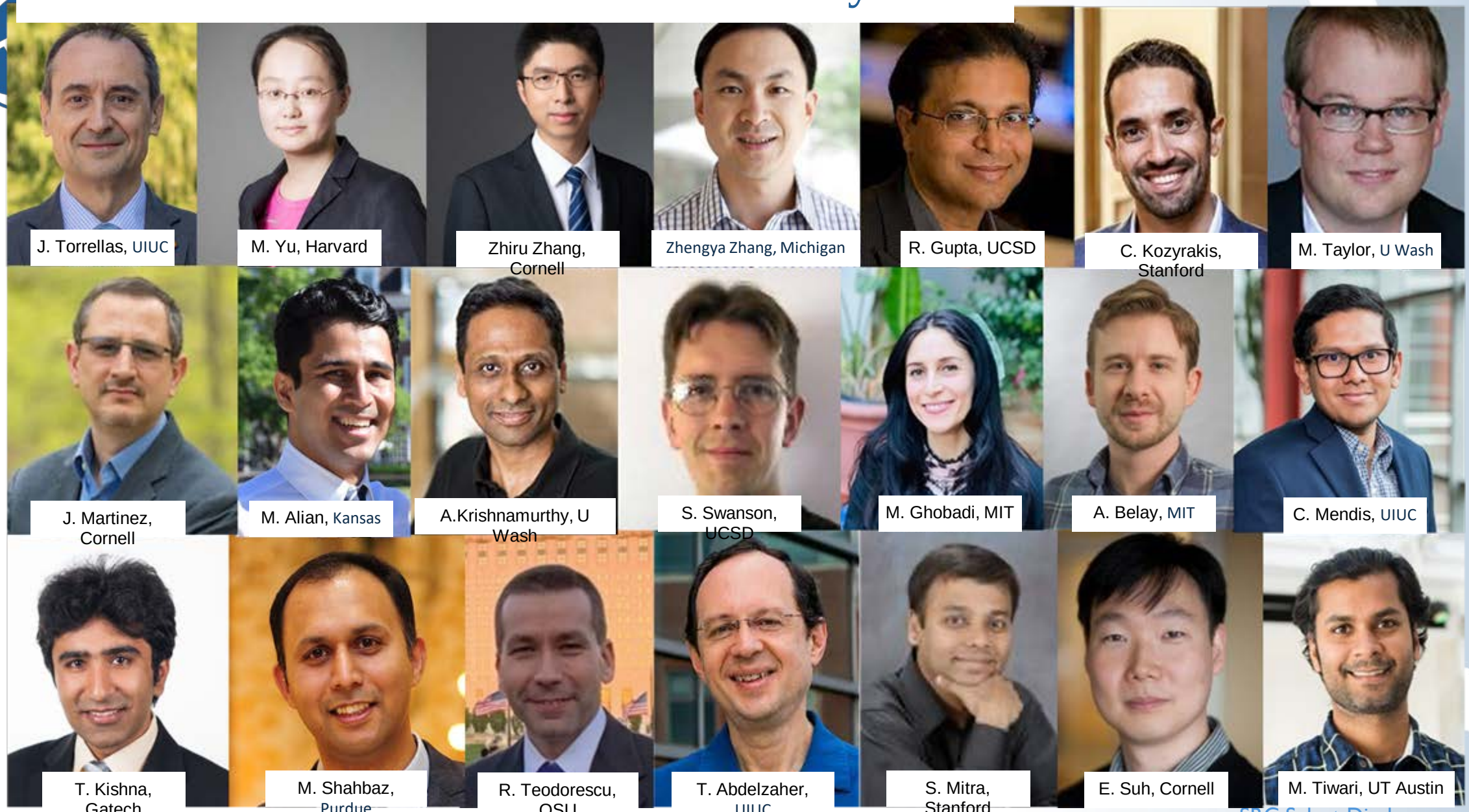


A Heterogeneous Large Cluster with Specialized Intelligence

Collaboration with Other JUMP 2.0 Centers



Our Mission: Make this Vision a Reality



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

Minlan Yu

- Gordon McKay Professor of Harvard University
- Research interests:
  - Data center networks and software-defined networking
  - Accelerating network communications for large-scale distributed systems
  - Programmable switches and smartNICs
- Relevance to the ACE center:
  - Theme 3: Fine-grained communication and coordination
  - Support evolving application requirements by redesigning network stack
  - Improve performance and energy efficiency by leveraging programmable switches and smartNICs

Manya Ghobadi

- MIT EECS
- Research interests:
  - Large-scale distributed systems
  - Efficient systems for machine learning and AI
  - Reconfigurable networks
- Relevance to the ACE center:
  - Theme 3: Fine-grained communication and coordination
  - Efficient network infrastructures are crucial to support the demand for current and emerging application.
  - My work is about building scalable, cost-effective systems to provide high-bandwidth, low end-to-end latency, and high availability.

Networking is Critical in Distributed Systems

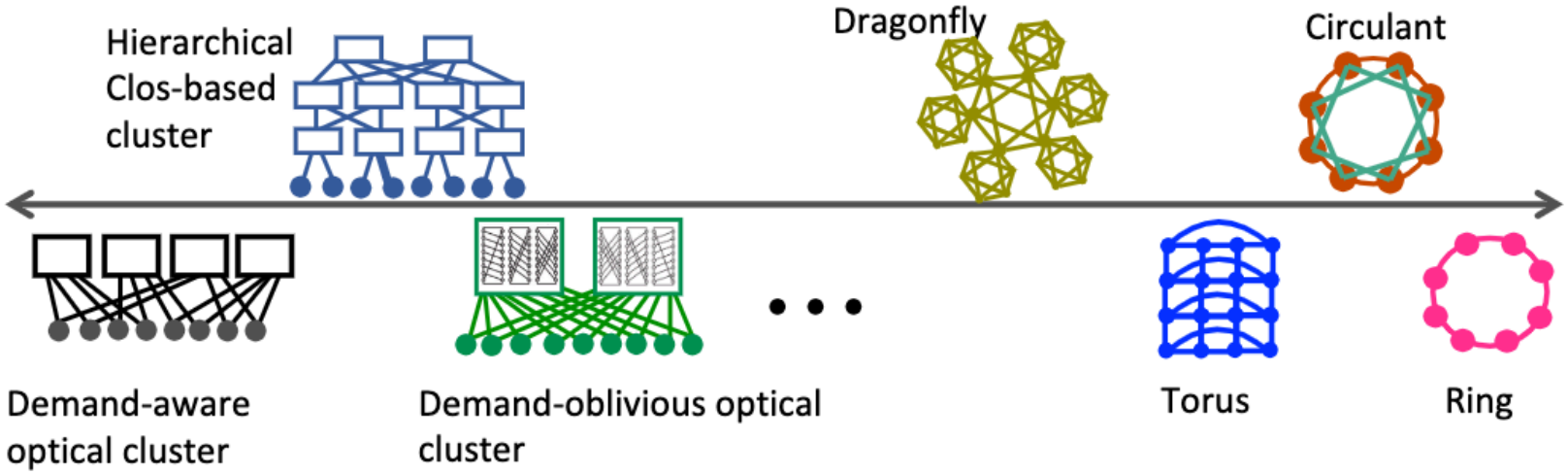
- Trends of future distributed systems
  - Large scale, high throughput, low (tail) latency, diverse requirements
- E.g., machine learning:
  - Ever growing ML models + semiconductor scaling slowdown ==> scalable ML systems
  - If accelerators give us 10x – 100x in FLOPS/$, we need a 10x – 100x in Gbps/$
- E.g., cloud systems
  - New applications invented daily with widely different workloads and requirements
  - Reduce resource stranding: Need networks to shift compute to data or vice versa

Challenges and Vision

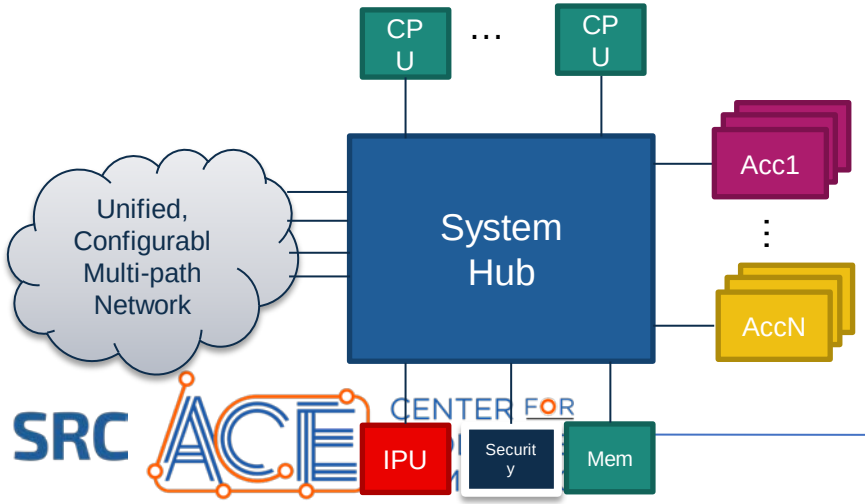
- Key challenges
  - Increasing needs for high throughput and low latency
  - Evolving application requirements and diverse accelerators
  - Stranded resources with high cost

- Our vision
  - Application-specific evolvable design: Automatically adapt network fabric and stack
  - Fine-grained scheduling and migration of data and compute
  - End-to-end optimizations across fabric and hosts

Task 3.1: Network Topology Design for Accelerator-Rich Datacenters



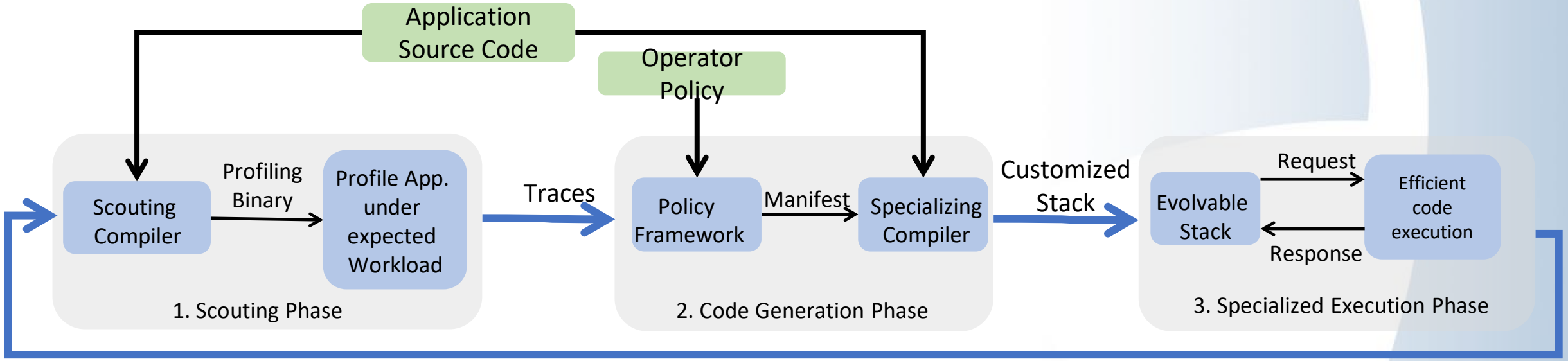
- Goal: build a single physical network that gradually evolves to cover emerging workloads and newer accelerators.



System hub:

- Bridge multiple interconnect domains within each server:
  - Memory interconnects (e.g., UPI, Infinity, NVLink, or CXL)
  - I/O interconnects (e.g., PCIe)
  - Network connections (Ethernet)

Task 3.2 Developing Evolvable Stacks



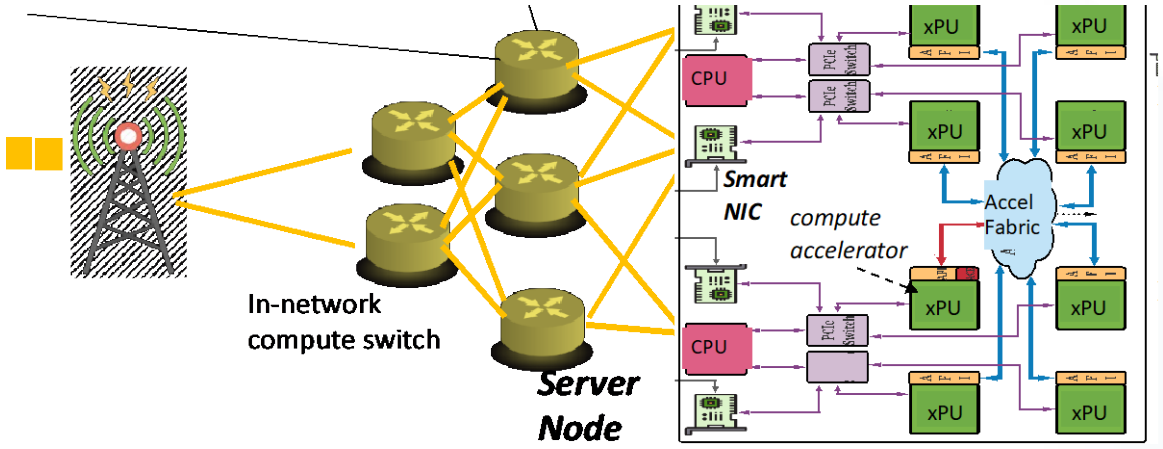
- The three phases that generate evolvable stacks:
  1. Scouting phase: automatically monitors the application to identify the workload properties and application requirements.
  2. Code-generation phase: determines which parts of the stack can be customized for each accelerator
  3. Specialized execution phase: executes our specialized stack for any request that enters the system.

Task 3.3: Self-balancing, planet-scale distributed runtime

- Accelerators should not be waiting for data to arrive
- Also, data movement should be minimized
- Bundle the computation in small buckets and ship it remotely to where the data lives

Task 3.4: In-network Computing

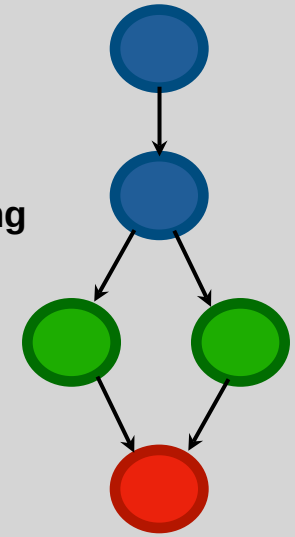
- Network switches and smartNICs
  - High throughput, low latency, energy efficiency for some types of computation
  - Offload operations traditionally done by cores
- Coordination across switches, NICs, hosts, accelerators
  - Programming and resource constraints at switches and NICs
  - Division of labor between network and hosts



Task 3.5: Hardware-supported Intelligent Distributed Data Stores

- Geo-distributed databases replicate information across distributed sites
- Optimize the storage and accelerators to store Exabytes of data and serve them at tens of TB/s

App's Data Access Specification



Data Processing Operations

Data Access SLOs & Reqs

SmartNIC

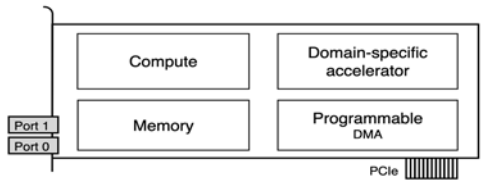
Server

Programmable Switch

Network Accelerator Card

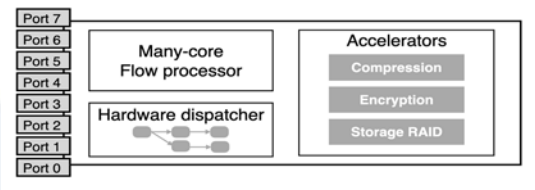
SmartNIC

Device Tray (cSSDs)



SmartNIC
- Reliability management
- Consistency, durability
- Storage bandwidth mgmt

Programmable switch
- Provide QoS and fairness
- Deadline-driven delivery
- Simple data xforms



Network accelerator
- Complex data xforms
- Compression, crypto, RAID

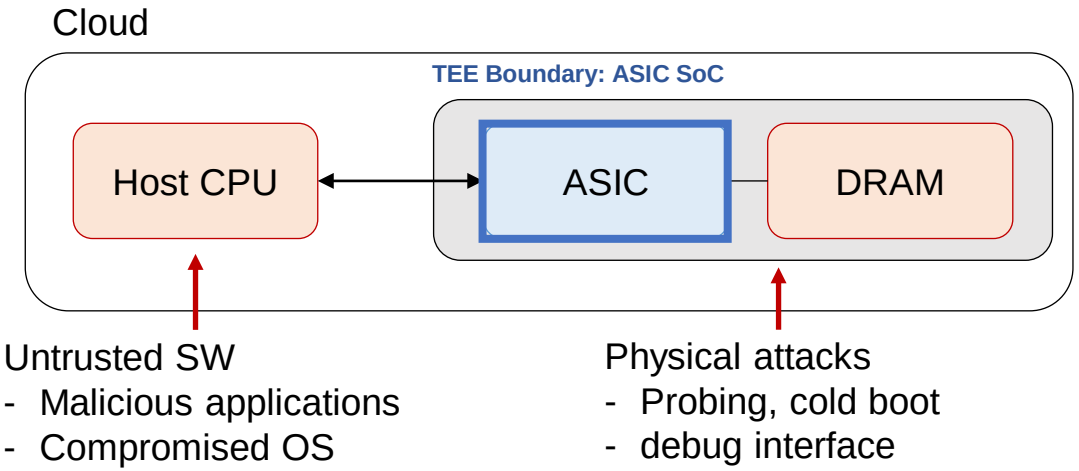
Radu Teodorescu

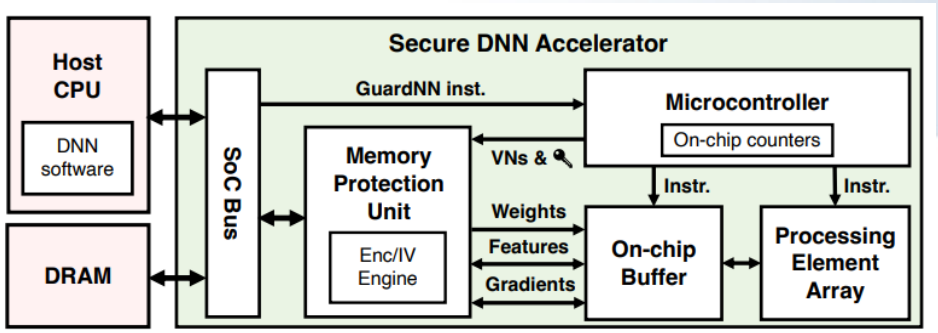
- Professor of CSE at Ohio State University
- Research Interests
  - Computer Architecture
  - Security: transient execution attacks, security verification, robust ML
  - Energy-efficient architectures, process variation, reliability
- Role in ACE Center
  - Leading Theme 4: Security, Privacy and Correctness
  - Focused on developing a new security framework for an accelerator-rich future

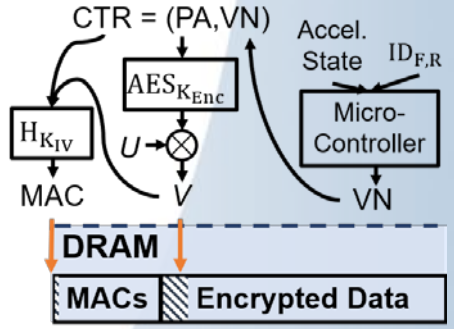
Radu Teodorescu

- Key challenges
  - Hardware accelerators deployed at all levels of computing stack
  - Accelerators are heterogenous, rapidly evolving, performance-optimized
  - How do we secure these systems?
- Hey insights
  - Accelerators provide new performance vs. security tradeoffs
  - Multi-tenancy introduces new attack vectors
  - Need to re-think security verification and develop new tools
  - Evolvable accelerators will benefit from verifiable designs

Trusted Execution Environments for Heterogeneous Accelerators







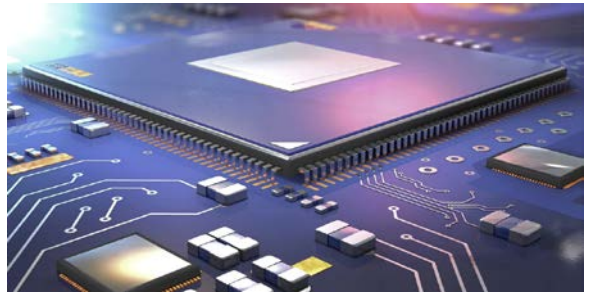
Develop high-performance, high-assurance TEEs for machine learning (ML) accelerators

- Leverage physical isolation, minimize trusted code base
- Customize protection leveraging regular/coarse-grained data movements, nearly overhead-free encryption

- Extend to all types of accelerators, using FPGAs, GPUs, SmartNICs, etc.
  - Compiler-based framework to generate customized TEEs in a programmer-friendly manner

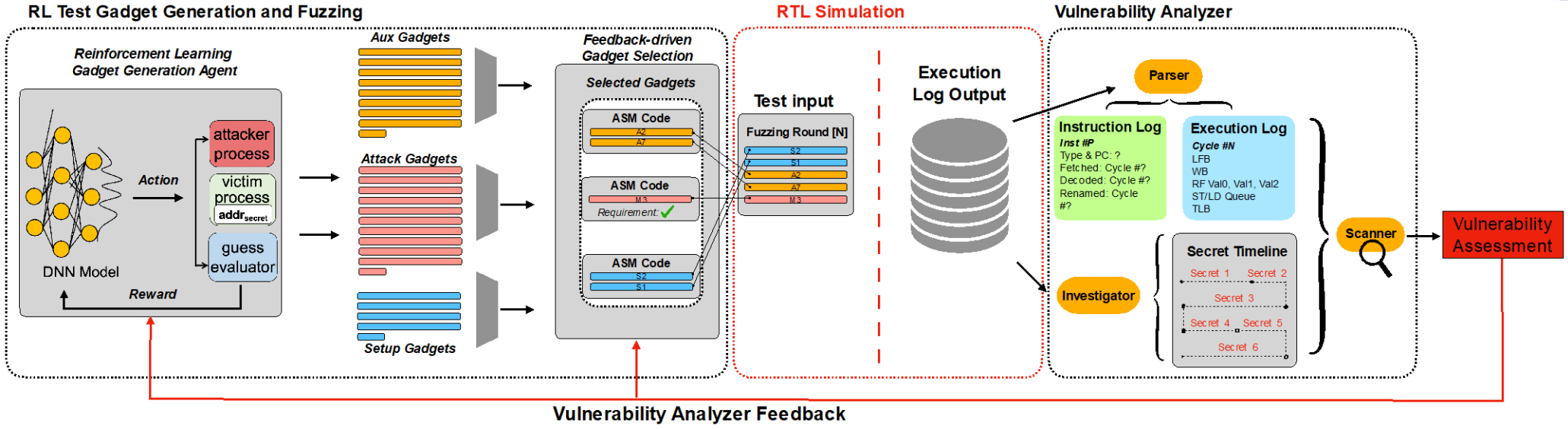
Secure Multi-Tenant Accelerators

- **Problem**: multi-tenant accelerators for efficiency and throughput – challenge for ensuring isolation and data integrity
- Proposed solutions:
  - Characterize leakage severity in multi-tenant accelerators, a new attack vector
  - Accelerator architectures designed for fine-grain, leakage-aware partitioning
  - Hardware-level **information flow control** to enforce and verify timing-sensitive isolation
- We will leverage multi-tenant designs developed in this and other JUMP 2 centers



Security and Privacy Verification and Assurance

- We need to rethink tools and methods for security verification of accelerators
  - Automatic RTL-level instrumentation of accelerator designs, to capture detailed execution state for verification
  - Model-guided fuzzing tools will automatically generate test vectors and attack gadgets
  - RTL simulation used to generate detailed execution logs
  - Automatic security verification and vulnerability analysis of execution logs
  - Prototype framework was used to verify TEE software/hardware revealing new vulnerabilities in open-source CPUs



Design for Verification of Evolvable Hardware Accelerators

- Rapidly evolving accelerators and short design-to-deployment timelines will require fast and complete verification
- Propose new formal verification approach called Accelerator Quick Error Detection (A-QED)
- Propose new Design-for-Verification (DFV) principles