# DECADES: Deeply-Customized Accelerator-Oriented Data Supply Systems Synthesis

Margaret Martonosi
Luca Carloni
David Wentzlaff

PRINCETON UNIVERSITY

COLUMBIA UNIVERSITY

# DECADES: A VERTICALLY-INTEGRATED APPROACH

**Language and Compiler Support**

Lead: Martonosi

- Enhance data locality
- Optimize spatial mapping of threads
- Enable in-memory computing

**Very Coarse-Grained Reconfigurable Tile-Based Architecture**

Lead: Carloni

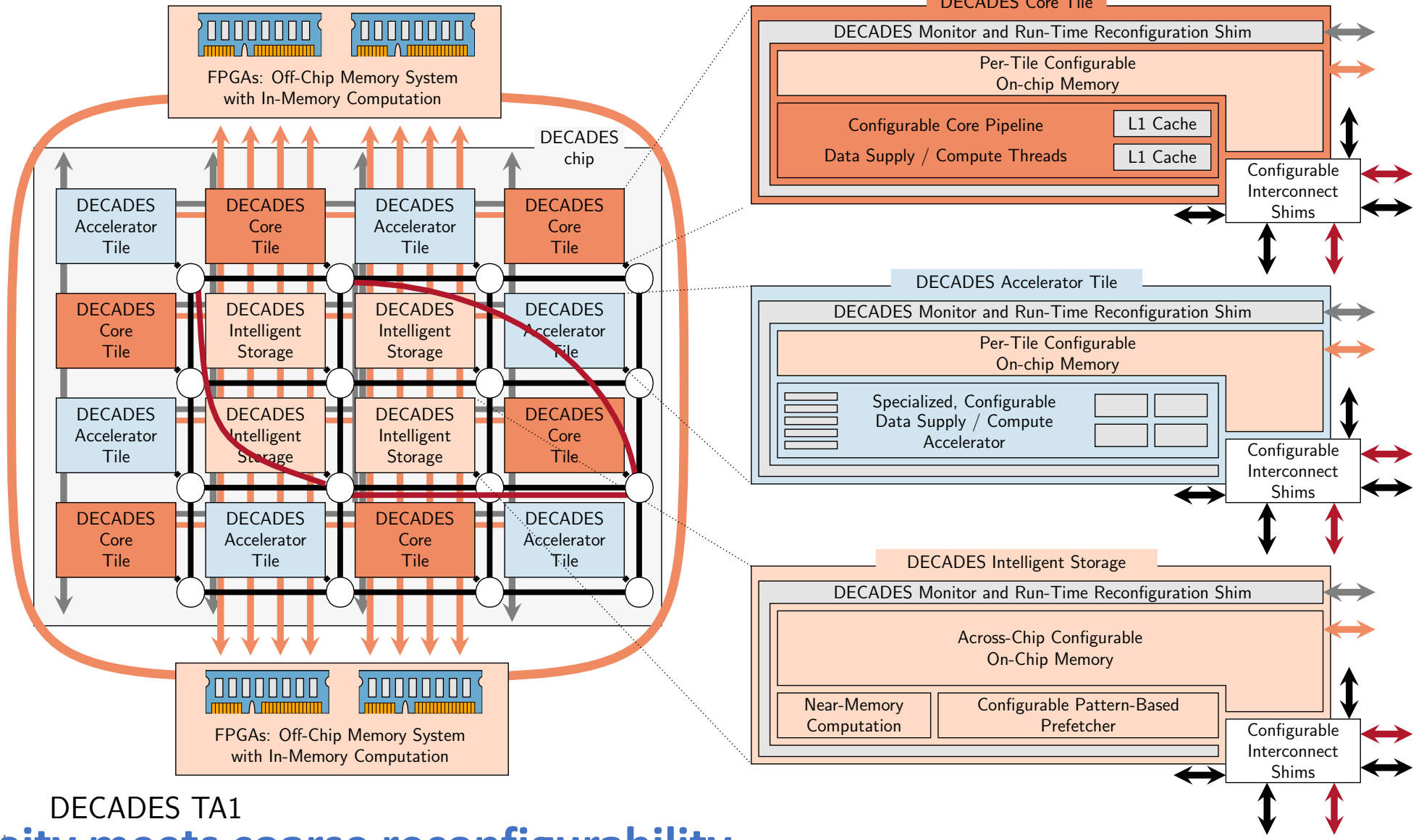
- Coarser than CGRA → VCGRTA
- 3 classes of reconfigurable tiles
- Reconfigurable interconnection network
- Reconfigurable in-memory computing

**Multi-Tiered Demonstration Strategy**

Lead: Wentzlaff

- Scalable full-system simulation
- Multi-FPGA emulation infrastructure
- 225-tile DECADES chip prototype

# DECADES PLATFORM ARCHITECTURE




DECADES TA1

**Heterogeneity meets coarse reconfigurability**

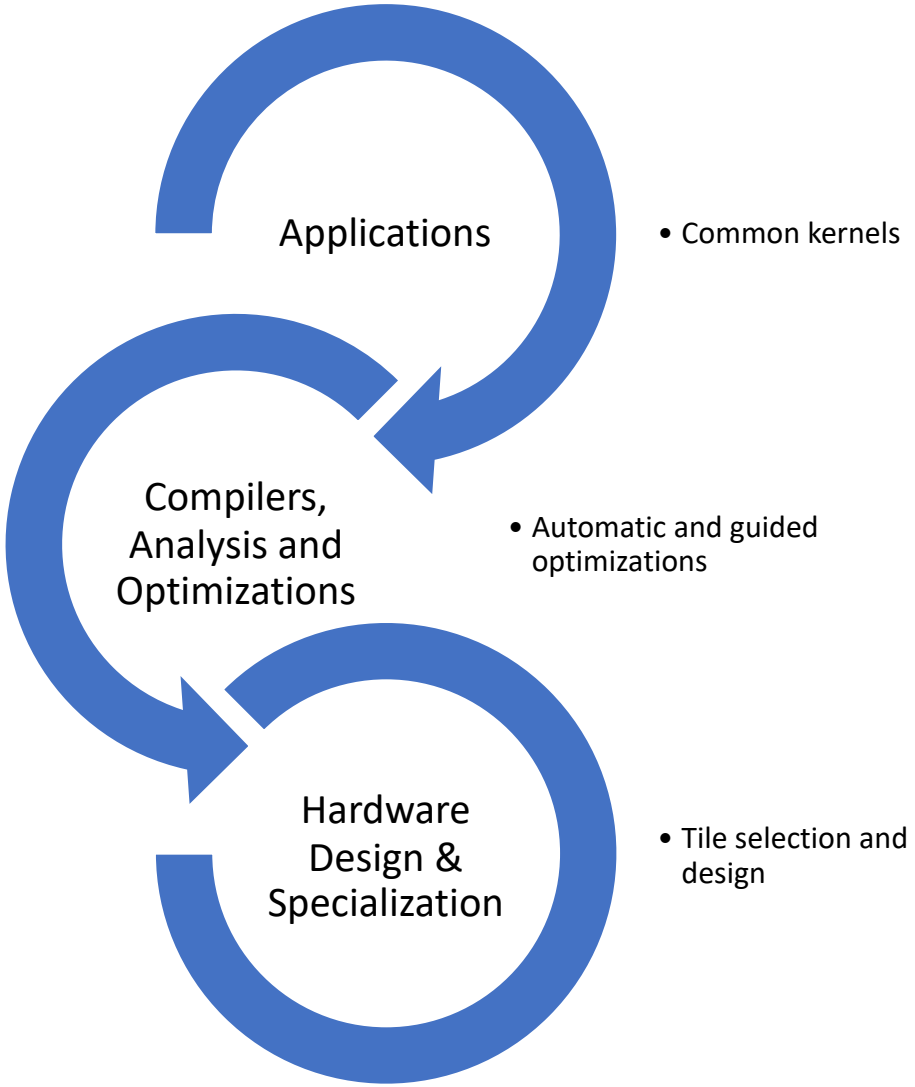
# Project Milestones

| Task Category | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| Language, Compiler & Runtime System (TA2) | 1.1: Initial Design of DECADES Language, Compiler and Runtime System | 2.1: Dynamic Adaptation in DECADES Software Systems | 3.1: Full Static/Dynamic Optimization in DECADES Language, Compiler, Runtime SYstem |
| Application Development (TA1, TA2) | 1.2: Transformation of Provided Benchmarks | 2.2: Optimization of Provided Benchmarks | 3.2: Full benchmark characterization on DECADES Hardware |
| Platform Architecture (TA2, TA2) | 1.3: Initial Design of DECADES Platform Architecture | 2.3: Full Design of DECADES Platform Architecture | 3.3: Optimization and Scaling of DECADES Platform Architecture |
| Simulation and Emulation (TA1, TA2) | 1.4: Lightweight Simulator and Emulator for DECADES Platform | 2.4: Full-system Simulator and Emulator for DECADES Platform | 3.4: Scalable Multi-FPGA-Based Emulator for DECADES Platform |
| Hardware Design (TA1) | 1.5: Test chip development | 2.5: DECADES Chip design | 3.5: Full Hardware System Demonstration |
| Technology Transfer (TA1, TA2) | 1.6: Transfer of Phase 1 Results and Deliverables | 2.6: Transfer of Phase 1 and Phase 2 Results and Deliverables | 3.6: Transfer of Results and Deliverables from all three phases |

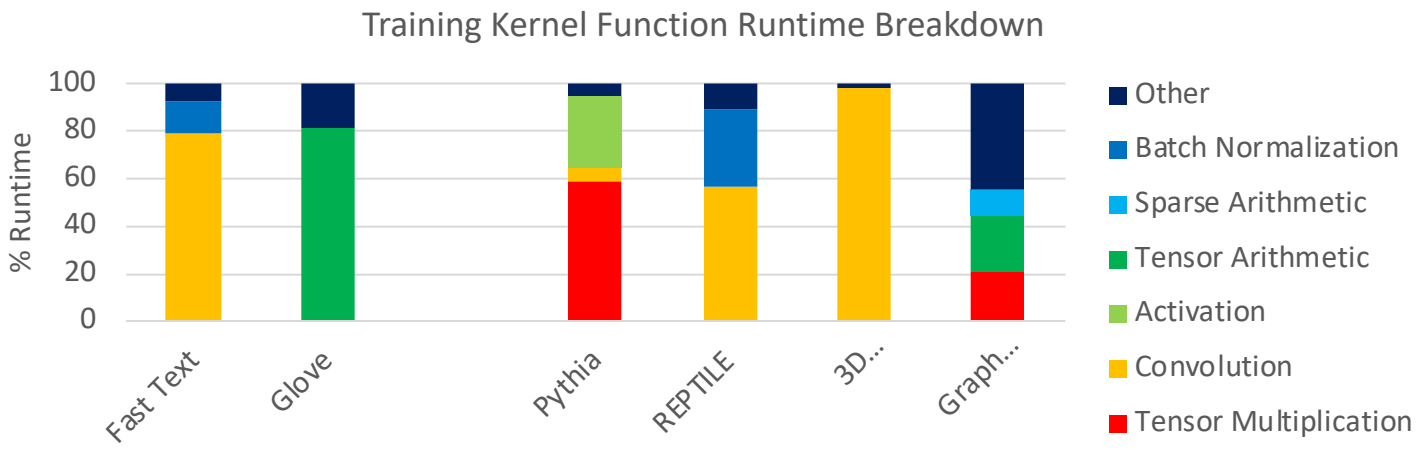
# This Talk

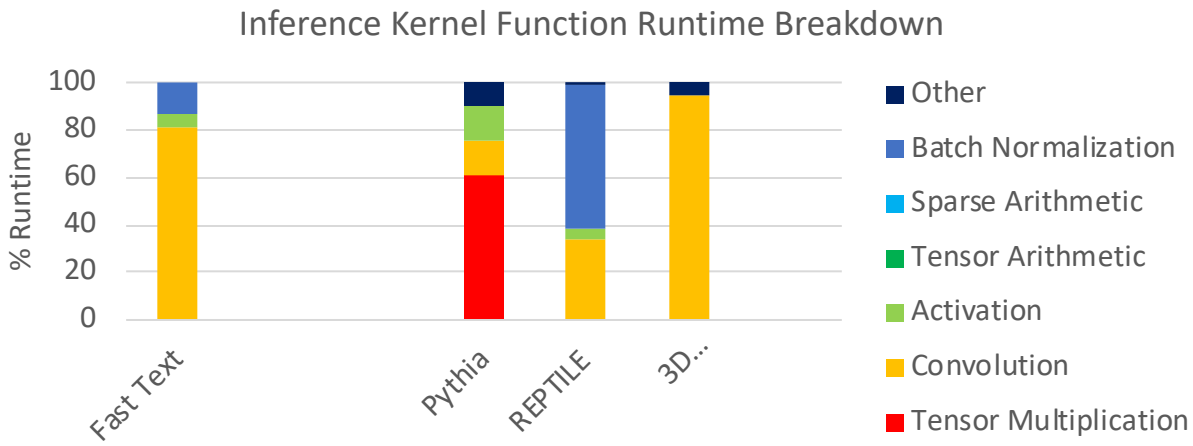
- Compiler and Chip Development
- Other Research Status Updates

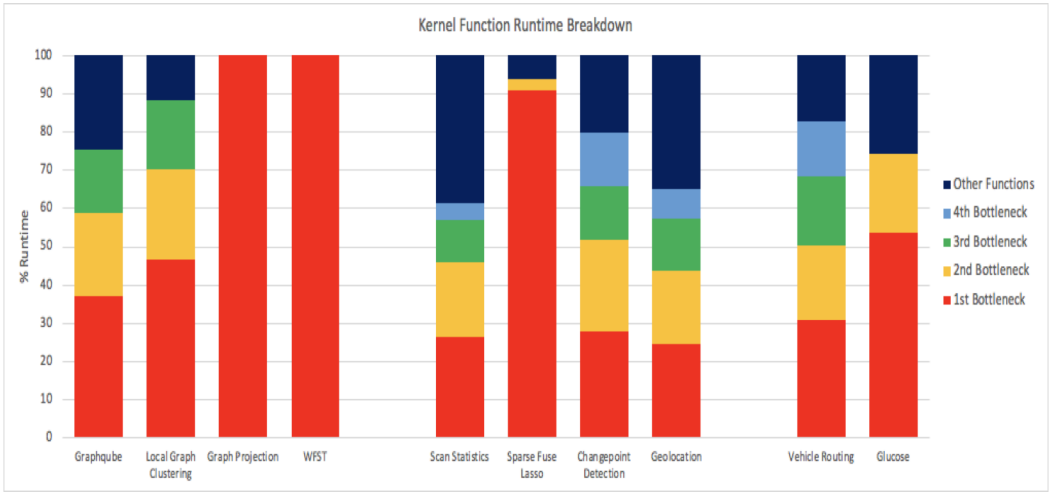
# Status & Approach



# Application Profiling: Data -> Design Plans




Graph (C/C++)

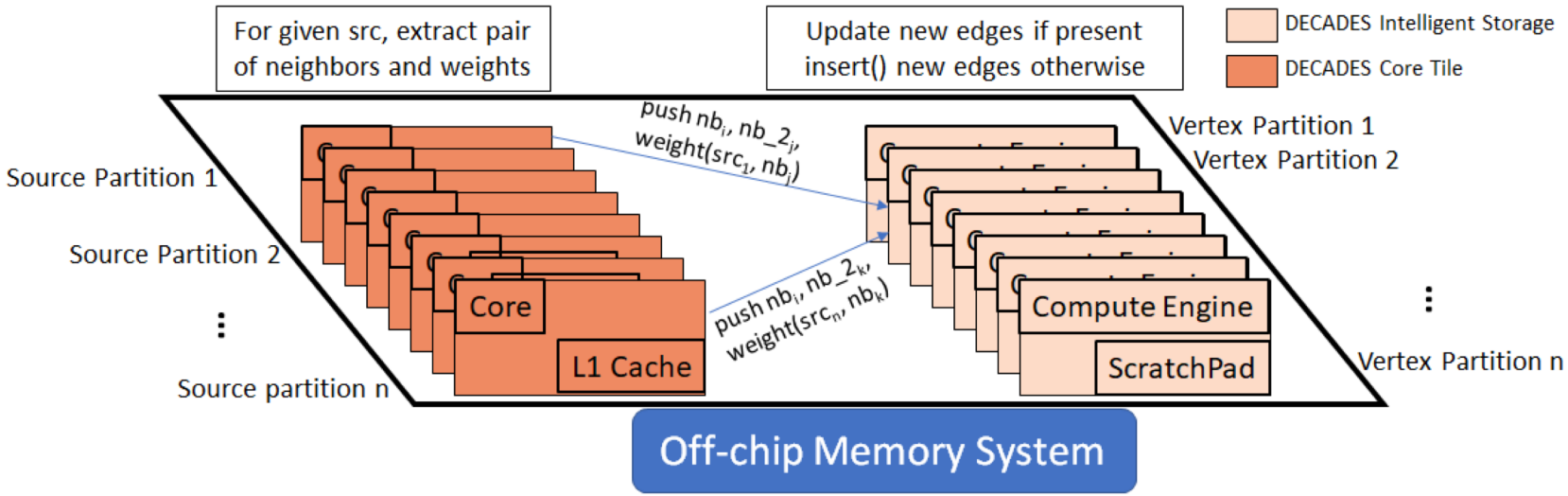
Graph (Python)

Combinatorics (C/C++)

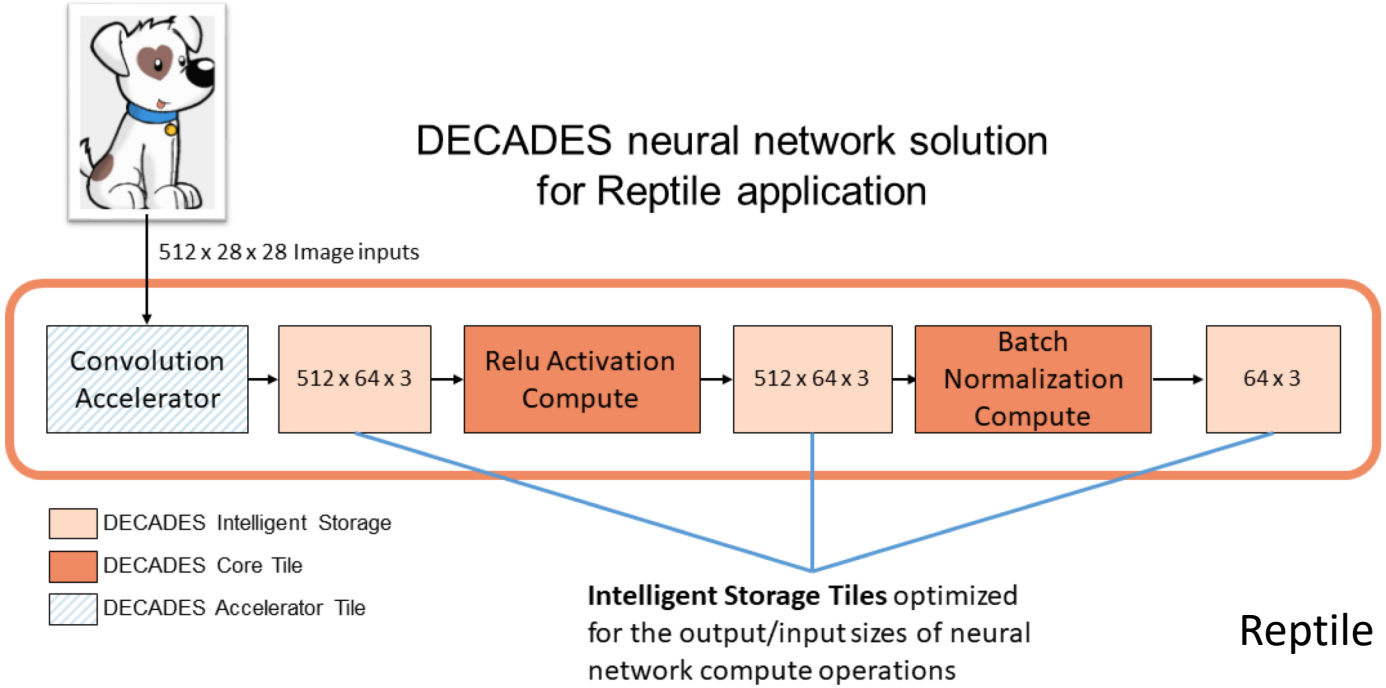
# Applications Mapping to DECADES

| Applications | DECADES features for acceleration | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accelerator tile | | | | | | | Memory tile | | | | | Processor tile |
| | convolution | vector arith | matrix arith | graph utilities | statistics | SOCP equations | map /reduce | efficient data streaming | prefetching graph data | prefetching SAT formulas | temp data for inter accelerator comm | in-memory / near-memory computation | prefetching graph data |
| 3D segmentation | X | | | | | | | X | | | | | |
| Pythia | X | X | X | | | | | X | | | | | |
| Text classification | | X | X | | | | | | | | | X | |
| Glove | | X | | | | | | X | | | | | |
| Reptile | X | X | | | | | | X | | | | | |
| Vehicle routing | | | | | | | | | | | | | X |
| Geolocation | | X | | | | | X | | | | X | | |
| Scan statistics | | | | | X | | | | X | | | X | |
| Local graph clustering | | | | X | | | | | X | | | | |
| Sparse fused lasso | | | | | | X | | | X | | | | X |
| Graph projections | | | | | | | | | X | | | X | |
| Graph classification | | X | X | | | | | | | | | | |
| WFST | | | | | | | | | X | | | X | |
| Glucose | | | | | | | | | | X | | | |
| Graph query-by-example | | | | X | | | | | X | | | X | |
| Changepoint detection | | | | | X | | | | X | | | X | |

# Example Mappings to DECADES




Graph Projections



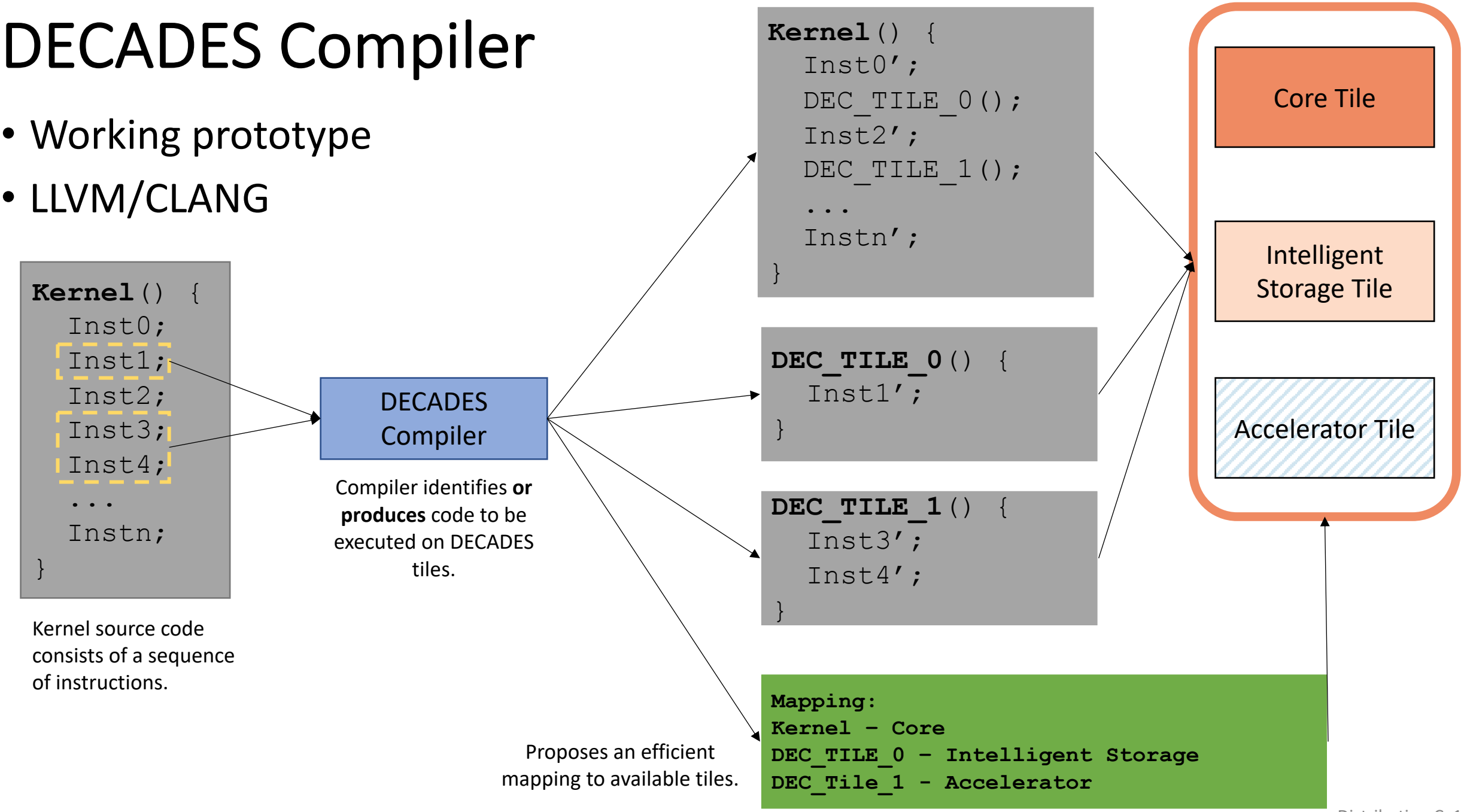

Reptile

# This Talk

- Compiler and Chip Development
- Other Research Status Updates

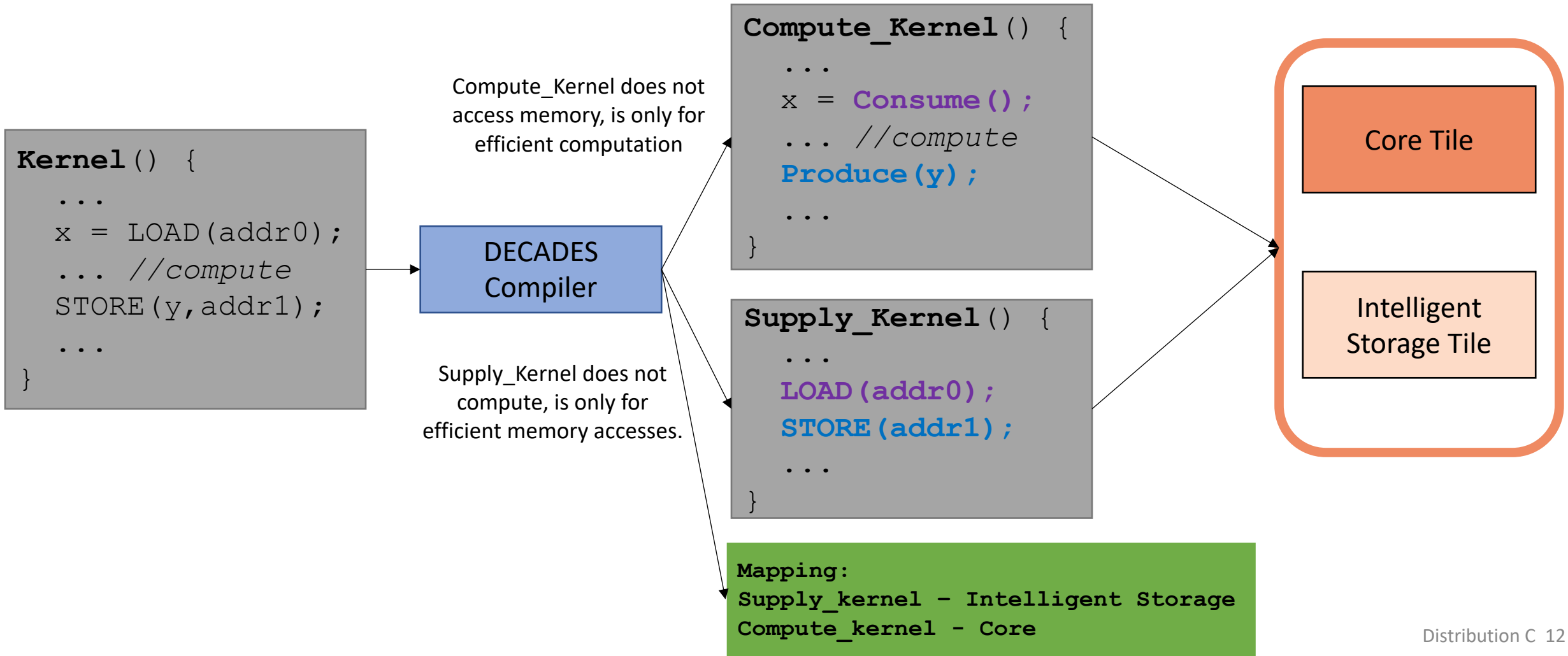
# DECADES Compiler

- Working prototype
- LLVM/CLANG

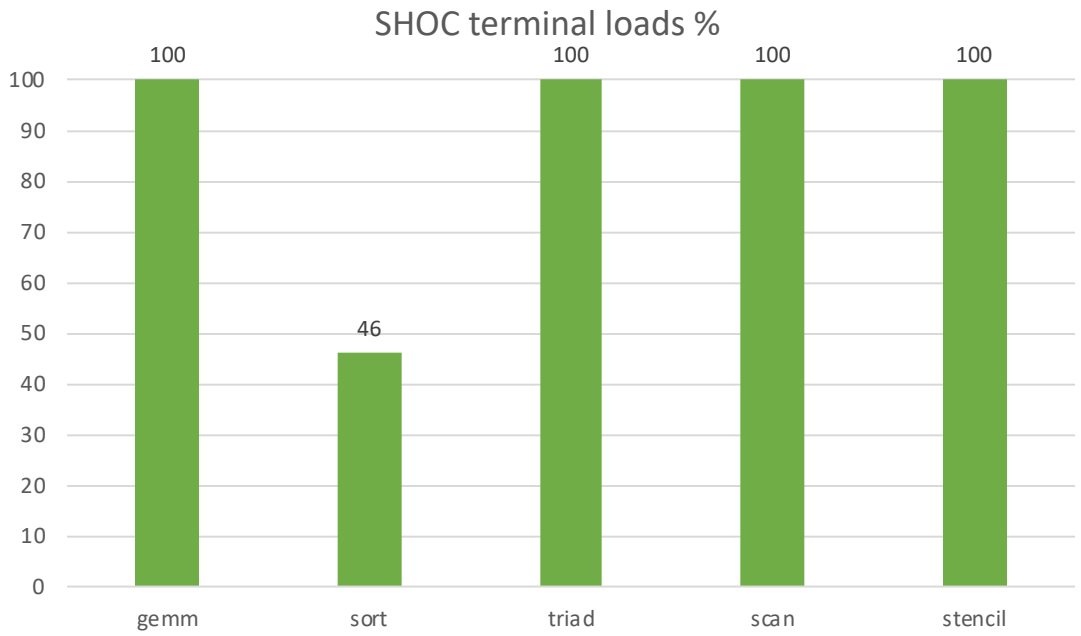


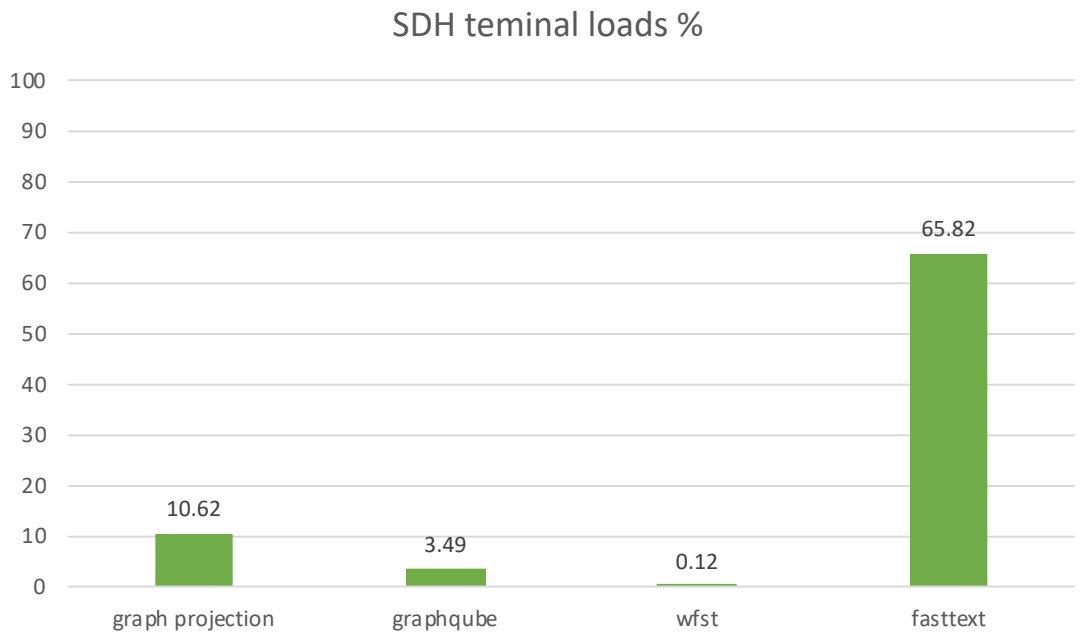
# DECADES Compiler: Decoupling Supply/Compute

Implemented in DECADES Clang/LLVM based compiler
Can automatically decouple SHOC, Parboil microbenchmarks, WFST, Graph Projections, Graphqube, FastText



# Terminal Loads & Decoupled Supply-Compute

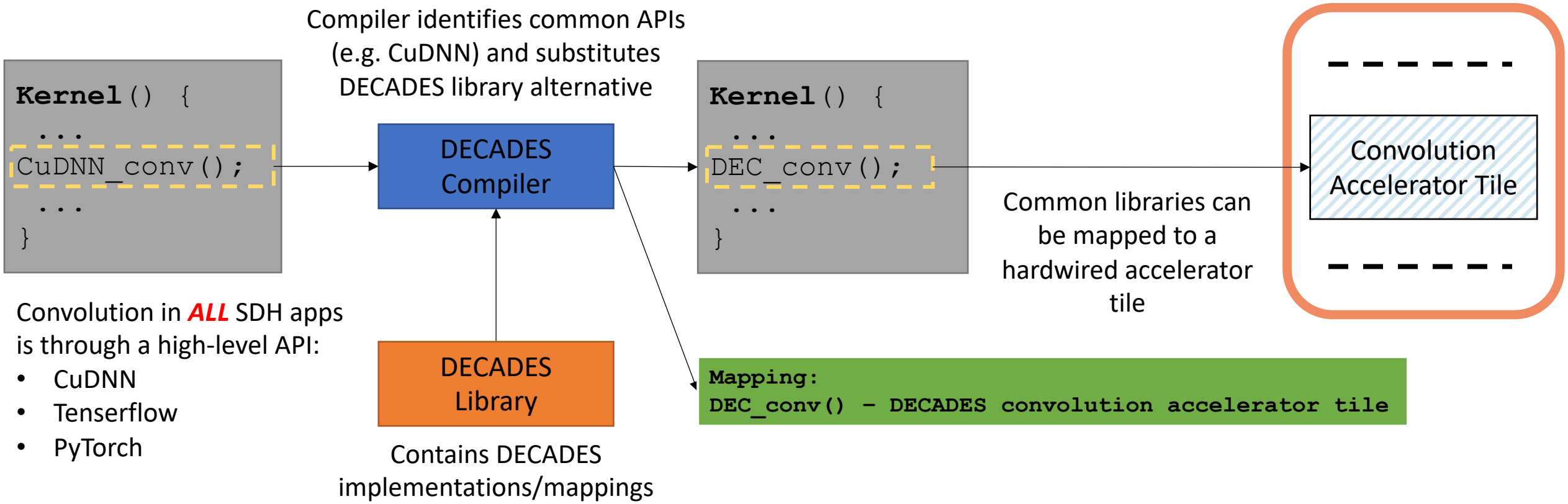



- SDH apps have fewer ***Terminal loads (***load for which supply kernel does not need load return values) than SHOC
- => Decouple at a coarser granularity:
  - Do not decouple *all* memory operations, allowing `Supply_Kernel` to have **more** *terminal loads* and execute sufficiently *ahead* of `Compute_Kernel`.

# DECADES Library Interfaces

- Building up DECADES library and API
  - Program annotations +
  - Efficient implementations/mappings of common kernels. e.g., ***convolution***.
  - Map to accelerator tile + memory wrappers

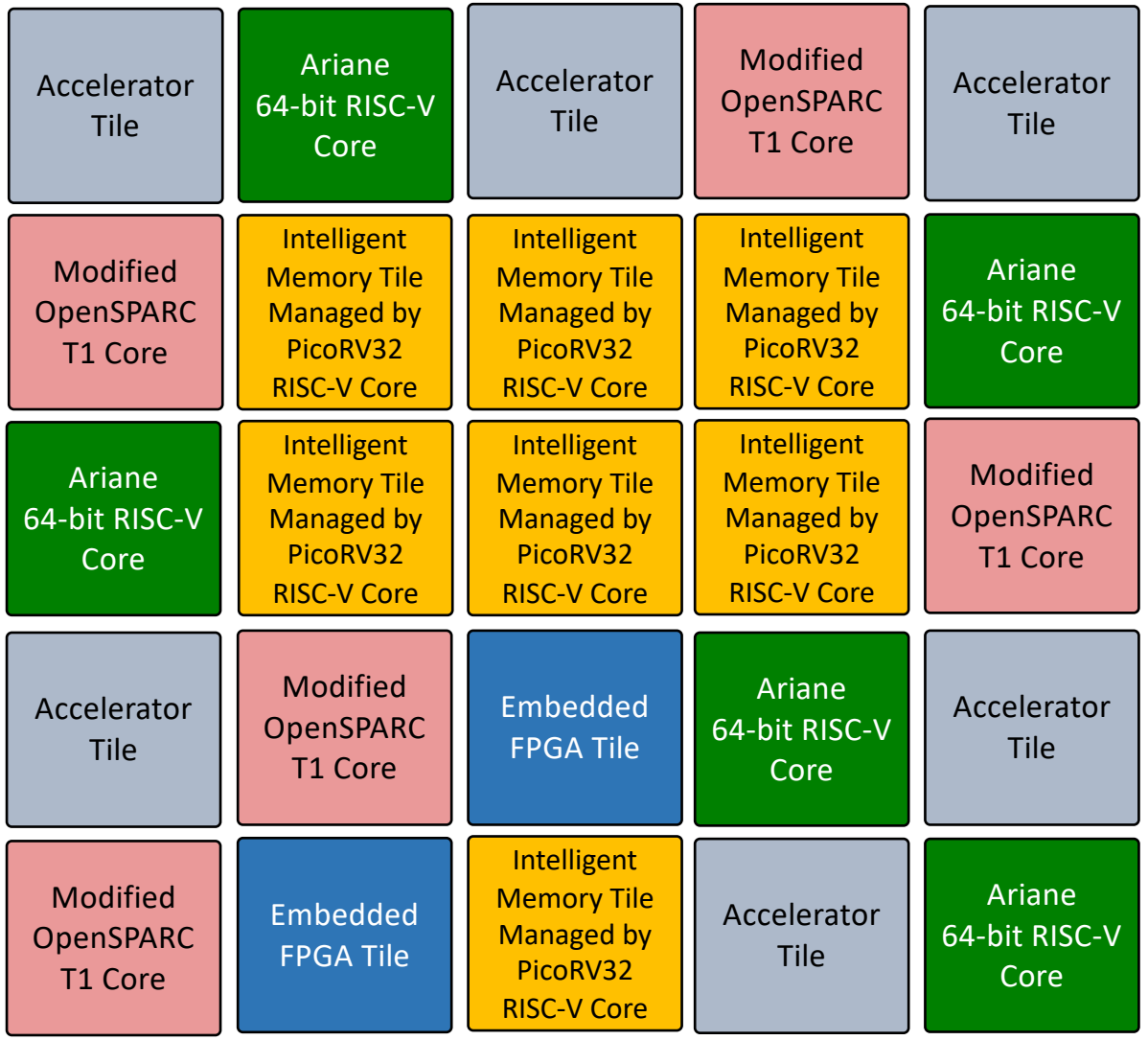


# Project Milestones

| Task Category | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| Language, Compiler & Runtime System (TA2) | 1.1: Initial Design of DECADES Language, Compiler and Runtime System | 2.1: Dynamic Adaptation in DECADES Software Systems | 3.1: Full Static/Dynamic Optimization in DECADES Language, Compiler, Runtime SYstem |
| Application Development (TA1, TA2) | 1.2: Transformation of Provided Benchmarks | 2.2: Optimization of Provided Benchmarks | 3.2: Full benchmark characterization on DECADES Hardware |
| Platform Architecture (TA2, TA2) | 1.3: Initial Design of DECADES Platform Architecture | 2.3: Full Design of DECADES Platform Architecture | 3.3: Optimization and Scaling of DECADES Platform Architecture |
| Simulation and Emulation (TA1, TA2) | 1.4: Lightweight Simulator and Emulator for DECADES Platform | 2.4: Full-system Simulator and Emulator for DECADES Platform | 3.4: Scalable Multi-FPGA-Based Emulator for DECADES Platform |
| Hardware Design (TA1 | 1.5: Test chip development | 2.5: DECADES Chip design | 3.5: Full Hardware System Demonstration |
| Technology Transfer (TA1, TA2) | 1.6: Transfer of Phase 1 Results and Deliverables | 2.6: Transfer of Phase 1 and Phase 2 Results and Deliverables | 3.6: Transfer of Results and Deliverables from all three phases |

# Planned DECADES Test Chip: Early 2020

- 9mm2 GF 14nm
- Sample of tiles
- Heterogeneous CPUs
- Tiles connected with multi-plane on-chip 2D Mesh network
- Intelligent memory tiles support optimized inter-tile data movement




Off Chip Memory System with In-Memory Computation implemented in FPGA

# DECADES Test Chip: CPU Heterogeneity

- Heterogeneity enables matching performance/energy profile to application phase
- DECADES Mixture of three core type
  - OpenSPARC T1 multithreaded core
    - Multithreaded efficiency
    - Full Stack OS
  - Ariane 64-bit RISC-V core
    - Decent performance 64-bit core
  - PicoRV32 32-bit RISC-V microcontroller
    - Area and Power-efficient/Low performance core drives intelligent storage

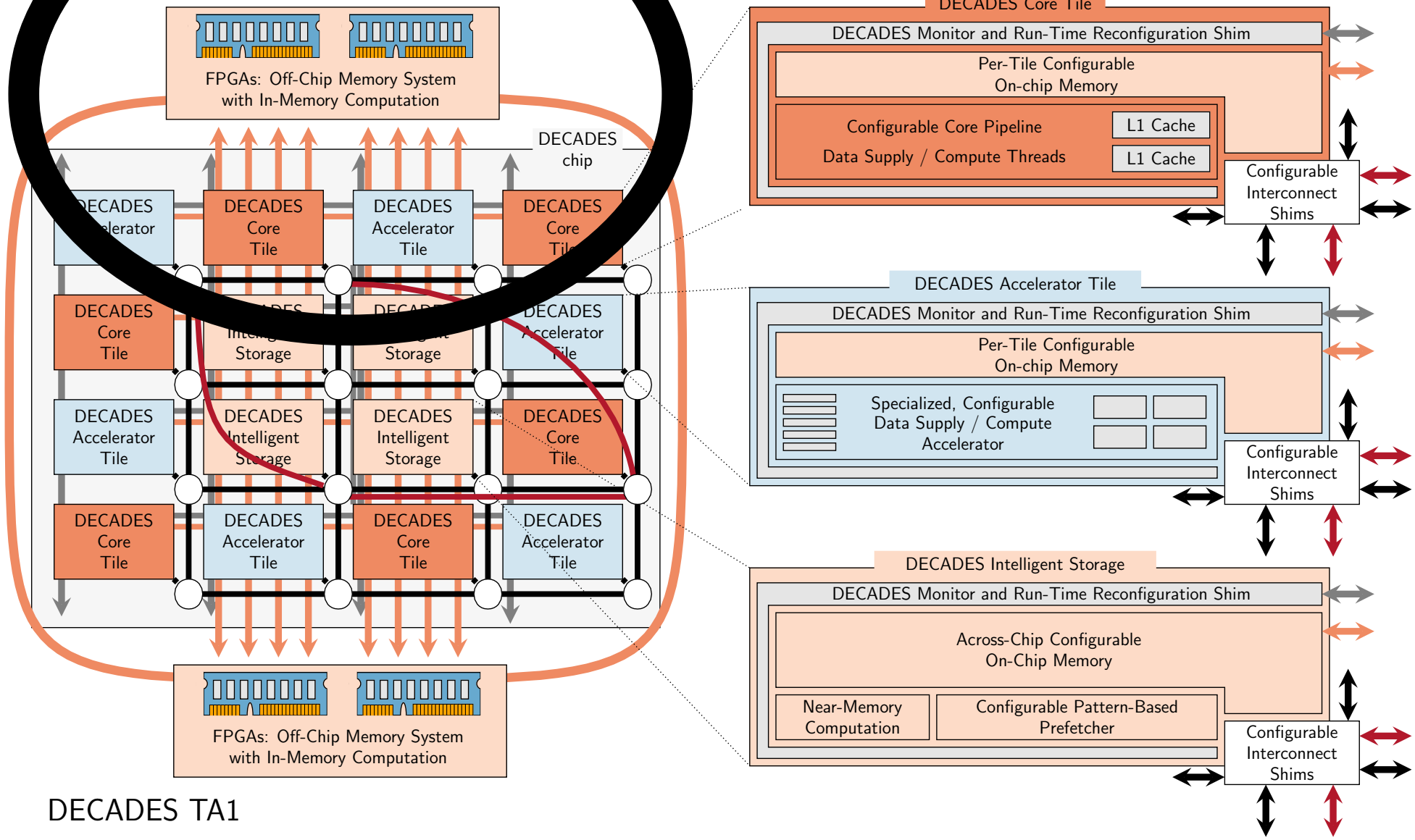
OpenPiton + Ariane 


- Specialized accelerator tiles
  - Ex: Convolution accelerator
- All cores and accelerators use common DECADES memory system with orchestrated data movement
- Leverage JuxtaPiton Expertise: Open-source, general-purpose, heterogeneous-ISA processor
  - Shared memory between 64-bit OpenSPARC T1 and 32-bit PicoRV32 cores
  - Boots Linux on OpenSPARC T1, offloads 32-bit RISC-V binaries to low-power PicoRV32 core

# This Talk

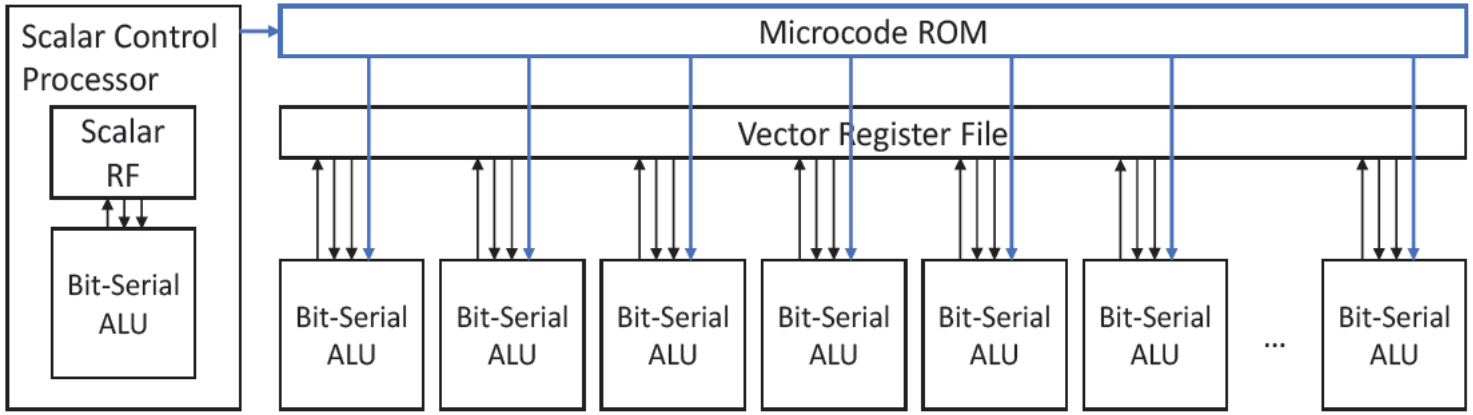
- Compiler and Chip Development
- Other Research Status Updates

# DECADES PLATFORM ARCHITECTURE




DECADES TA1

# Bit-Serial SIMD

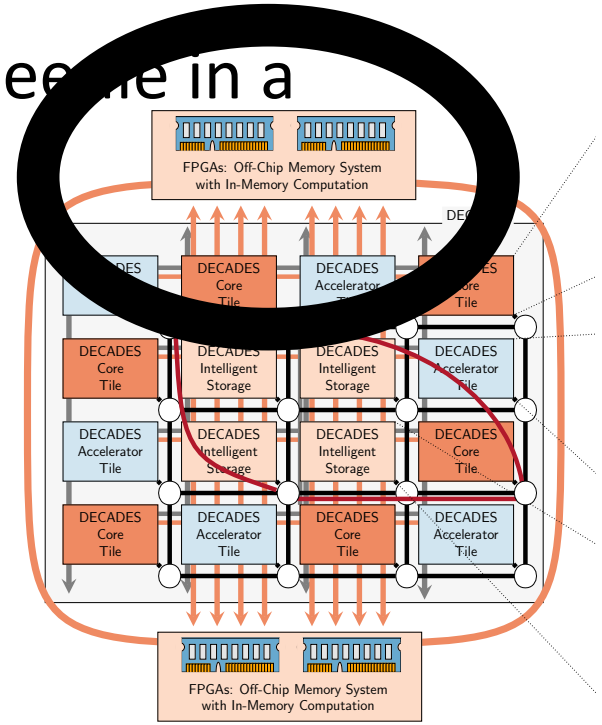



- Bit-Serial: Calculations on bit-level instead of word-level
  - Similar to breaking instructions into micro-operations
  - Dramatic reduction in datapath hardware
  - Overhead in latency and control logic
- Bit-Serial SIMD Exploit efficient datapath
  - Can fit many ALUs in a small area
    - Enable very wide SIMD parallelism – exploit data-level parallelism
  - Mitigate latency overhead by improving throughput
  - Minimize control logic overhead
  - Energy-efficient

**[Jackson & Wentzlaff]**

# Bit-Serial SIMD->Near-Memory Compute in DECADES

- Goal: Minimize data movement between memory and core by performing computation close to memory
- Opportunity: Small per-element computation on large data structure
  - FastText
  - Glucose
- Opportunity: Searching through large structures (finding a needle in a haystack)
  - Graphqube
  - MapReduce
  - Graph Projection
- Opportunity: Large numbers of indirect references
  - Glucose
  - BFS



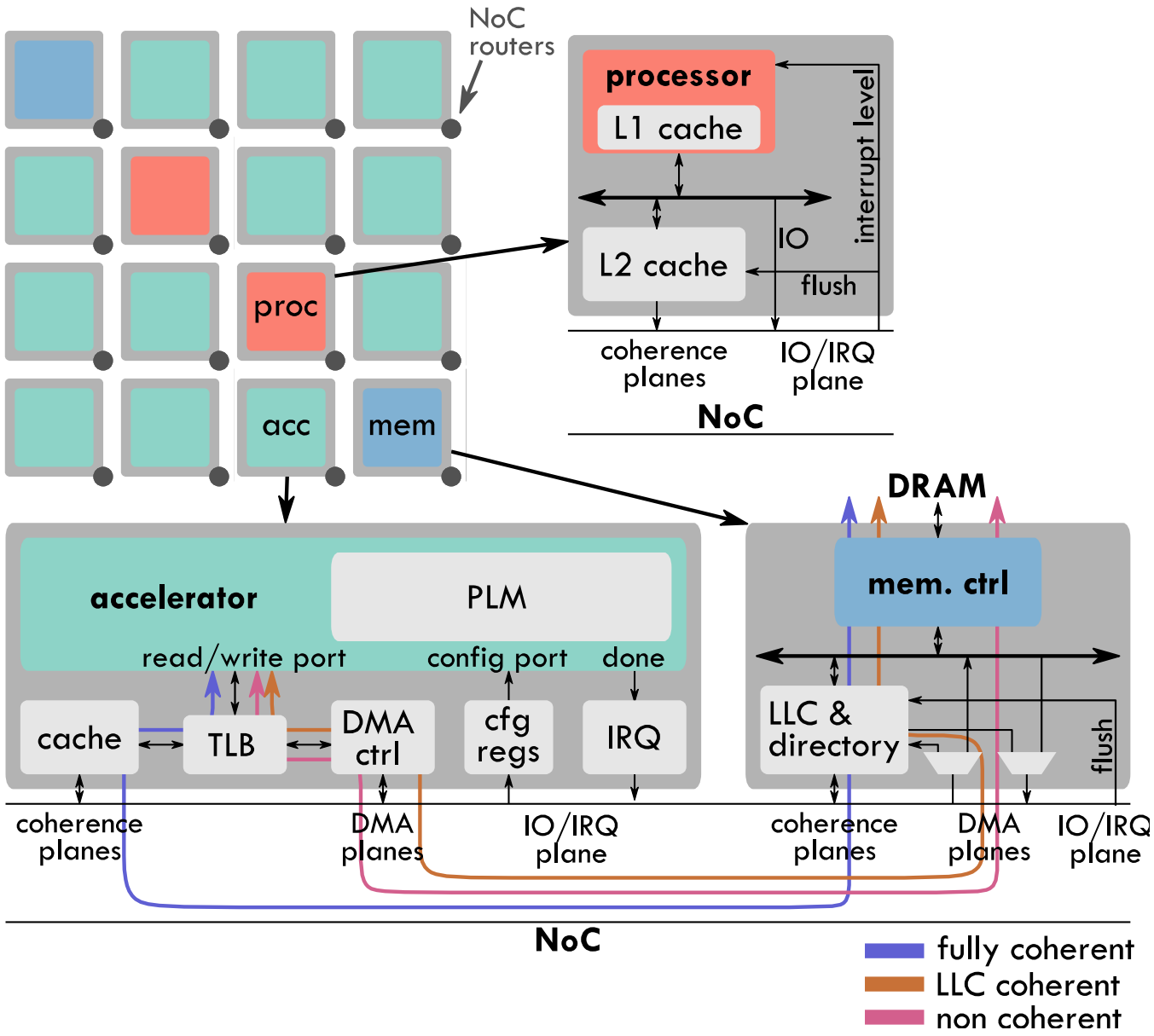

DECADES TA1

**[Jackson & Wentzlaff]**

# Accelerator Integration in DECADES Architectures

How are accelerators integrated in the DECADES architecture?

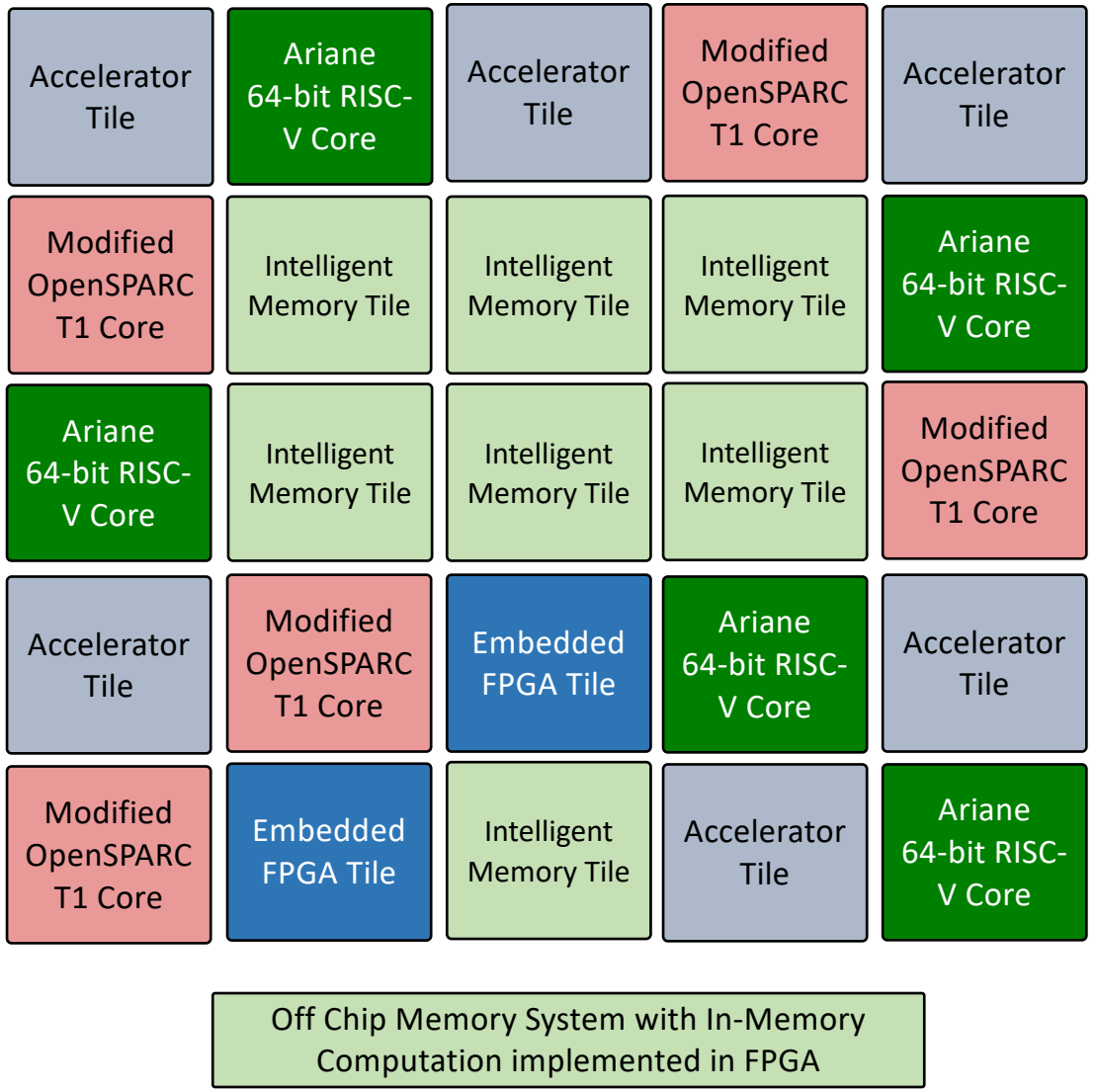
- Hardware Integration
  - Reconfigurable interface socket
- Software Integration
  - Invocation and (re)-configuration through Linux device drivers
- Interaction with Memory Hierarchy
  - Support for coexistence of 3 main heterogeneous coherence models
  - Run-time algorithm to select the optimal coherence model at each accelerator invocation




**Publications: [Giri et al., IEEE Micro'18] [Giri et al., NOCS'18] [Giri et al., ASPDAC'19]**

# Integrating Heterogeneous CPUs and Memory Consistency Models

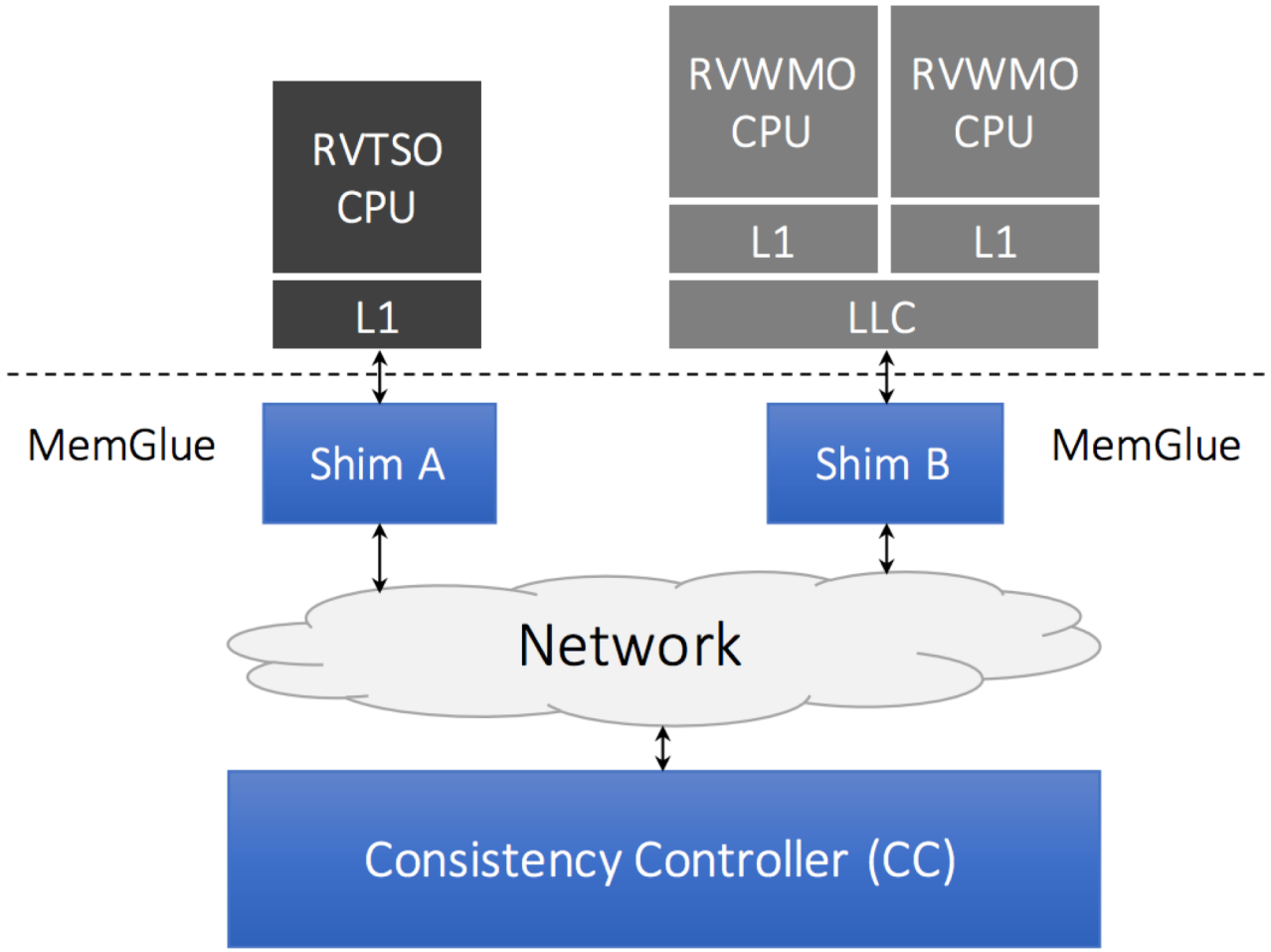
- Increasing heterogeneous parallelism → challenging to integrate hardware with heterogeneous memory models
- Key insight: hardware generally wants to support high-level language programs, e.g. C/C++
- MemGlue approach:
  - Consistency protocol designed to enforce C/C++ MCM requirements for a heterogeneously parallel system with minimal added hardware
  - Enables fine-grained communication between heterogeneous system components




**[Trippel, Manocha & Martonosi]**

# MemGlue Consistency/Coherence Shims

- Heterogeneous cluster (i.e. cluster) accelerators which share a localize
- MemGlue integrates clusters with
  - Consistency shims (i.e. shims) per-cl outside memory system
  - Consistency controller (i.e. CC) to int



# MemGlue Status and Performance Expectations

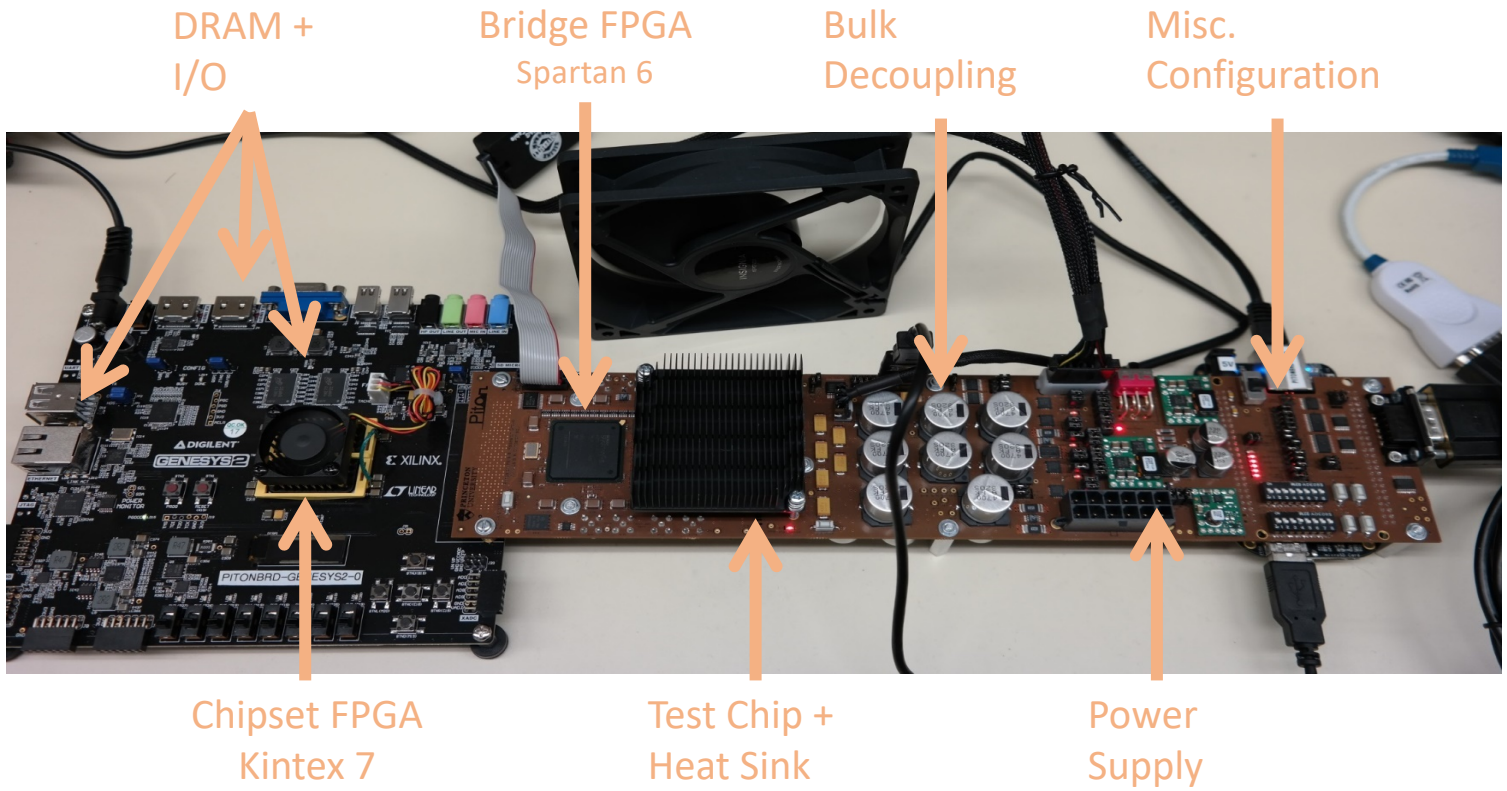
- Baseline MemGlue:
  - Timestamps (ts) instead of invalidation messages → eliminates traffic due to invalidations
  - Minimal storage requirements → ts for each cache line in the LLC of each cluster; sharer list in the CC
- MemGlue exploration and optimizations:
  - Explore methods for sending writes to CC only at sync point → minimize update traffic
  - Explore optimal buffer size → minimize traffic due to full shim buffers
- Performance expectation: achieve heterogeneous consistency performance approx. equal to the performance of the cluster with the strongest MCM
- Other goals: prove MemGlue properties using formal techniques

# Project Milestones

| Task Category | Phase 1 | Phase 2 | Phase 3 |
|---|---|---|---|
| Language, Compiler & Runtime System (TA2) | 1.1: Initial Design of DECADES Language, Compiler and Runtime System | 2.1: Dynamic Adaptation in DECADES Software Systems | 3.1: Full Static/Dynamic Optimization in DECADES Language, Compiler, Runtime SYstem |
| Application Development (TA1, TA2) | 1.2: Transformation of Provided Benchmarks | 2.2: Optimization of Provided Benchmarks | 3.2: Full benchmark characterization on DECADES Hardware |
| Platform Architecture (TA2, TA2) | 1.3: Initial Design of DECADES Platform Architecture | 2.3: Full Design of DECADES Platform Architecture | 3.3: Optimization and Scaling of DECADES Platform Architecture |
| Simulation and Emulation (TA1, TA2) | 1.4: Lightweight Simulator and Emulator for DECADES Platform | 2.4: Full-system Simulator and Emulator for DECADES Platform | 3.4: Scalable Multi-FPGA-Based Emulator for DECADES Platform |
| Hardware Design (TA1) | 1.5: Test chip development | 2.5: DECADES Chip design | 3.5: Full Hardware System Demonstration |
| Technology Transfer (TA1, TA2) | 1.6: Transfer of Phase 1 Results and Deliverables | 2.6: Transfer of Phase 1 and Phase 2 Results and Deliverables | 3.6: Transfer of Results and Deliverables from all three phases |

# Technology Transfer Plans & Status

- Outputs:
  - Software ecosystem
  - Chip design
  - FPGA emulation system
- Status:
  - Moved OpenPiton to github
  - Released two versions of OpenPiton
    - Support for JuxtaPiton (PicoRV32)
    - Support for Ariane (64-bit RISC-V core)
  - QEMU Instrumentation Plane
  - Before July 1:
    - DECADES Compiler + Pythia Simulator
    - One more OpenPiton release



# Conclusions

- Compiler:
  - Working prototype with ongoing feature additions
- Hardware:
  - Several tile designs ready.  More soon.
- Simulation/Emulation:
  - Lightweight simulator: Pythia
  - QEMU Instrumentation plane
  - FPGA Emulation