Princeton SDH DECADES Architecture

David Wentzlaff, Princeton University; Margaret Martonosi, Princeton University;
Luca Carloni, Columbia University

ELECTRONICS RESURGENCE INITIATIVE SUMMIT & MTO Symposium 2020 | Seattle, WA | August 18-20

Software Defined Hardware (SDH)

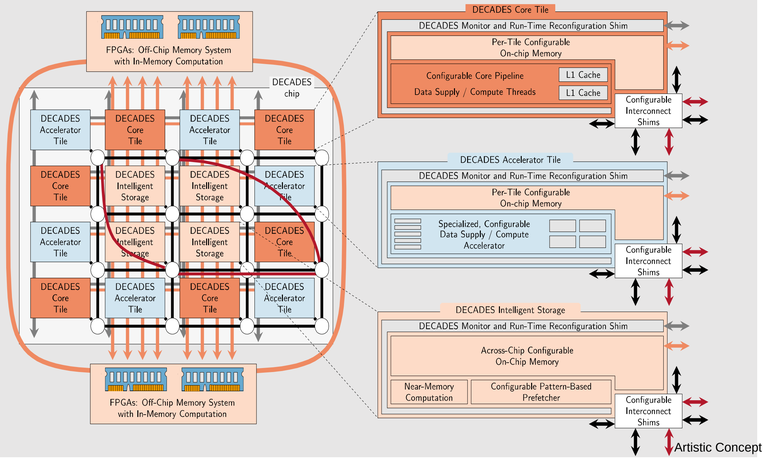
Artificial Intelligence

Background

Target Challenge: Data Supply is the Fundamental Bottleneck in Accelerator-Rich Computing Systems

- Hardware accelerators make data supply bottlenecks dominate runtime
- Key bottlenecks lie in supplying specialized accelerators with data
- Different accelerators and applications have different data supply needs
- Accelerators lack general-purpose latency-tolerance mechanisms
- Accelerator-rich computing requires big increases in memory bandwidth
- Targets machine learning and complex graph applications

DECADES Architectural Overview

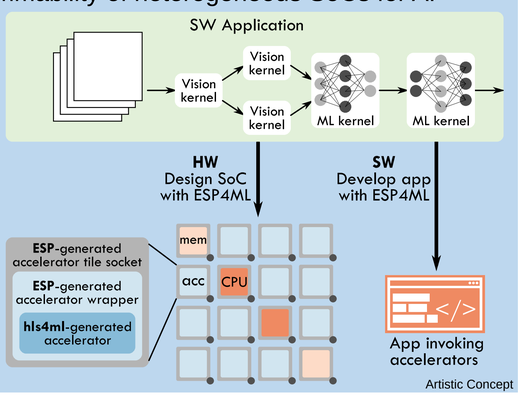


Artistic Concept

ESP4ML (AI accelerator portion of DECADES)
An Open Source Design Flow for Embedded AI Applications

Simplify design and programmability of heterogeneous SoCs for AI

- Generate accelerators from machine learning models given in Keras, Pytorch, ONNX
- Automate accelerator integration in the SoC
- Seamless accelerator programmability from target applications



Artistic Concept

Transition Paths: Open-Source Release Contributions

MosaicSim: A cycle-driven, LLVM-based simulator for heterogeneous systems
- https://github.com/PrincetonUniversity/MosaicSim

DEC++: LLVM-based compiler and runtime; supports C/++, and Python
- https://github.com/PrincetonUniversity/DecadesCompiler

MosaicSim and DEC++ Support: Docker/Documentation/Tutorial
- https://hub.docker.com/repository/docker/princetondecades/decades
- https://github.com/amanocha/DECADES_Applications

OpenPiton: General purpose, multithreaded manycore RISC-V processor
- https://github.com/PrincetonUniversity/openpiton

ESP: Open-source research platform for heterogeneous SoC design
- https://github.com/sld-columbia/esp

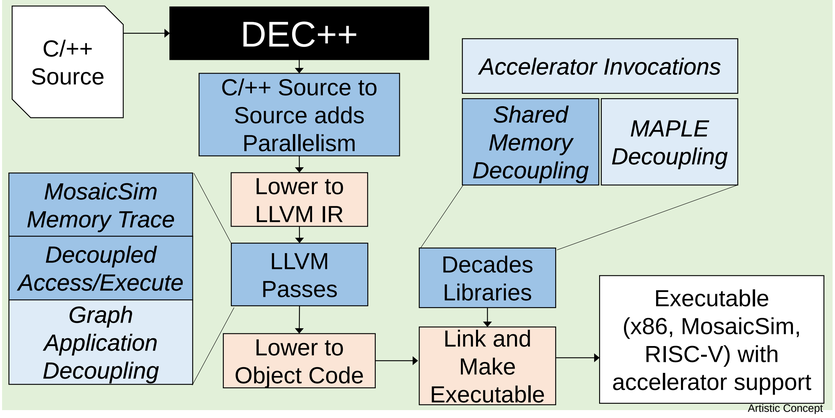
Approach

DECADES is a Vertically-Integrated Software/Hardware approach that combines Language and Compiler support to map complex graph and Machine Learning applications to a novel, heterogeneous, accelerator-rich manycore architectures.

DECADES Key Innovations:

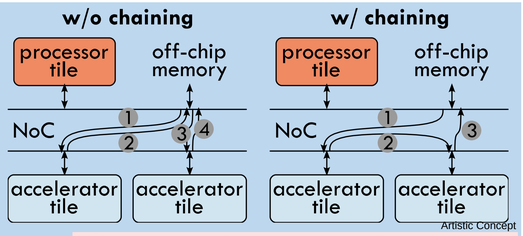
- Intelligent Storage tiles orchestrate on-chip data movement between accelerators and accelerators, accelerators and core, and core to core
- Best-of-breed pluggable accelerator socket and High-Level Synthesis flow ease accelerator integration (ESP and ESP4ML)
- Rich compiler (DEC++) and language infrastructure automatically slices applications and maps graph applications onto accelerators and cores
- DECADES architecture contains both near memory and in-memory computation to reduce energy of data movement (ComputeDRAM)
- Strong commitment to open source release of software and hardware

DECADES DEC++ Compiler and Accelerator Invocation Flow



Artistic Concept

ESP4ML Innovations

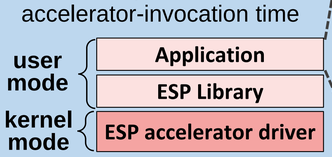


Artistic Concept

1) Accelerator Chaining
- Avoid memory roundtrips
- Fine-grained accelerators synchronization

2) Accelerator invocation API

A 3-functions API for accelerator invocation from user applications
- Automatically generated Linux device drivers
- No data copies needed at accelerator-invocation time

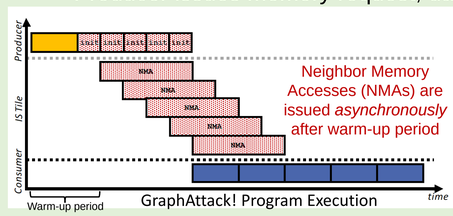



```
// ESP accelerators replace
// software kernels 2 and 4
{
  int *buffer = esp_alloc(size);
  for (...) {
    kernel_1(buffer,...);
// cfg_k2 has the accelerator
// configuration parameters
    esp_run(buffer, cfg_k2);
    kernel_3(buffer,...);
    esp_run(buffer, cfg_k4);
  }
  esp_free(buffer);
}
```

Artistic Concept

Results and Impact

GraphAttack!: HW/SW co-design to **hide long latencies** of indirect *Neighbor Memory Accesses (NMAs)* that bottleneck **graph applications**
- DEC++ Producer/Consumer program slicing where Producer issues NMAs and Consumer performs computation with their data
- Intelligent Storage Tile asynchronously performs NMAs
  - Producer issues memory request; data provided to Consumer

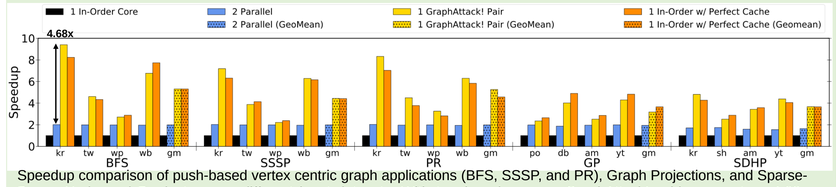



```
for node in frontier:
  val = process_node(node)
  for neib in G.neighbors(node):
    update = update_neib(node_vals, val, neib)
    if(add_to_frontier(update)):
      new_frontier.push(neib)
```

NMAs are pointer indirect and therefore have long latencies

Iterative, frontier-based graph application template

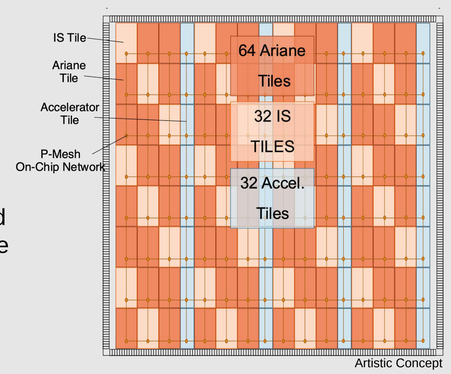
Artistic Concept



Speedup comparison of push-based vertex centric graph applications (BFS, SSSP, and PR), Graph Projections, and Sparse-Dense Hadamard Product across different input datasets. With two in-order cores, GraphAttack! achieves up to a 4.68x speedup (geomean 2.48x) over traditional do-all parallelism and can outperform an in-order core with a perfect cache.

DECADES Testchip 1

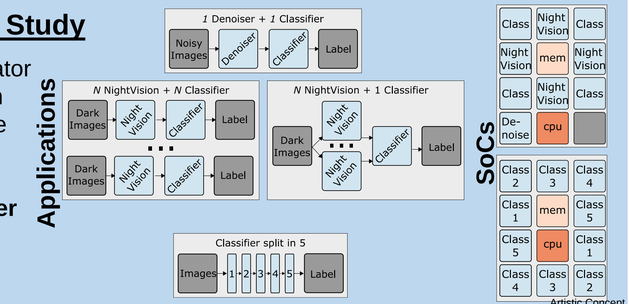
- Enables testing hardware and software innovation
- Over 100 tiles
- Core Tiles (cache coherent)
  - RISC-V 64-bit Ariane
- Intelligent Storage Tiles
  - Programmatically controlled data movement and storage
- Accelerator Tiles
  - Specialized hardware
- Over 1B transistors
- 1.5GHz target frequency



Artistic Concept

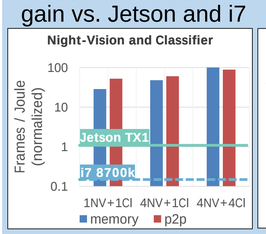
ESP4ML Case Study

Two multi-accelerator SoC prototypes on FPGA with multiple accelerators
- **Night-vision**
- **Image classifier**
- **Denoiser**
- **Autoencoder**

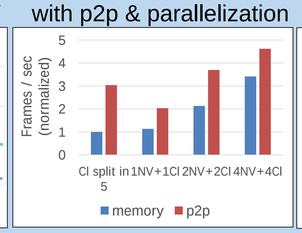


Artistic Concept

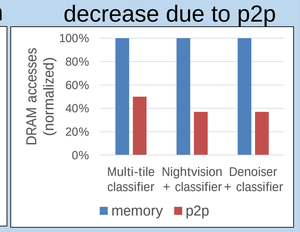
Energy efficiency: 100x gain vs. Jetson and i7



Performance: 4.5x gain with p2p & parallelization



Memory accesses: 3x decrease due to p2p



This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA).
The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.
Distribution Statement A – Approved for Public Release, Distribution Unlimited