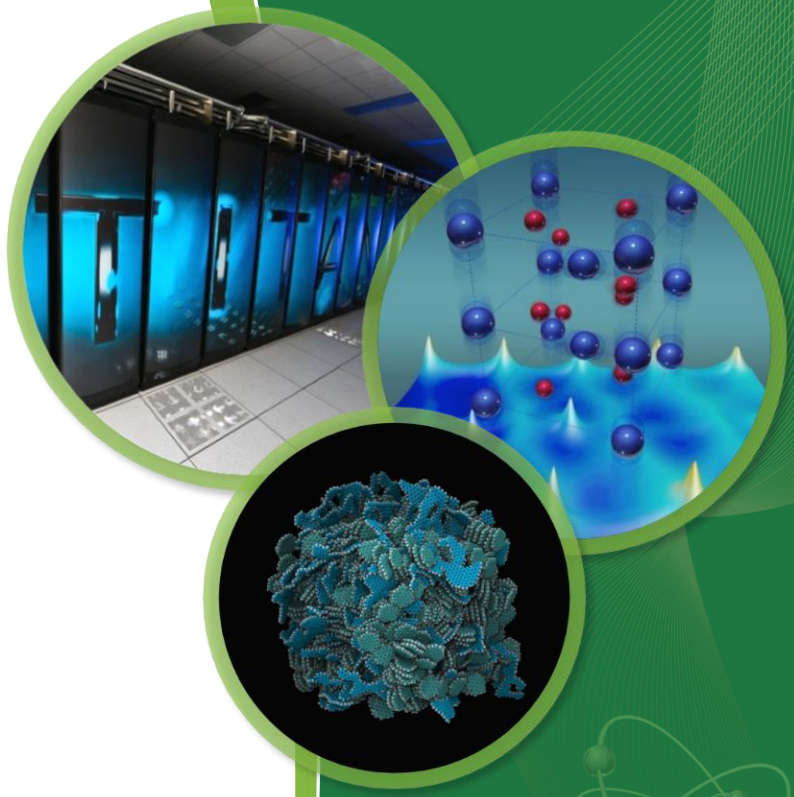
# Working Toward Performance Portability for FPGAs in High Performance Computing

Jeffrey S. Vetter

Seyong Lee

*Many contributions from FTG Group and Colleagues*



*Presented to*
**3rd International Workshop on FPGA for HPC (IWFH)**

Tokyo

Monday, 12 March 2018

Future Technologies Group ornl

http://ft.ornl.gov vetter@computer.org

OAK RIDGE National Laboratory

# Overview

- Recent trends in extreme-scale HPC paint an uncertain future
  - Contemporary systems provide evidence that power constraints are driving architectures to change rapidly
  - Multiple architectural dimensions are being (dramatically) redesigned: Processors, node design, memory systems, I/O
  - Complexity is our main challenge
- Applications and software systems are all reaching a state of crisis
  - Applications will not be functionally or performance portable across architectures
  - Programming and operating systems need major redesign to address these architectural changes
  - Procurements, acceptance testing, and operations of today's new platforms depend on performance prediction and benchmarking.
- We need performance prediction and portable programming models now more than ever!
- Programming systems must provide performance portability (beyond functional portability)!!
  - Heterogeneous systems: FPGAs are a great example
  - New memory hierarchies
- Our prototype OpenACC to FPGA compilation infrastructure shows promise

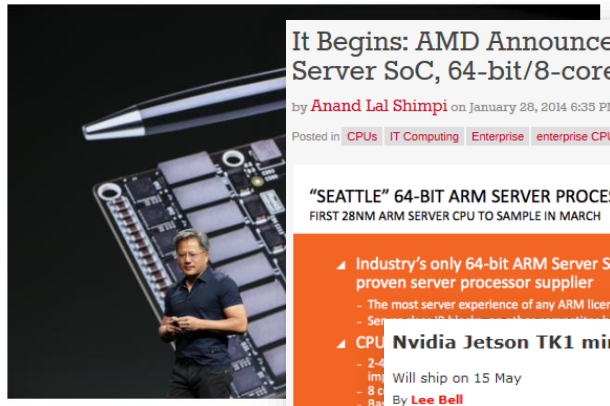
# In the news

## Nvidia and IBM create GPU interconnect for faster supercomputing

"NVLink" shares up to 80GB of data per second between CPUs and GPUs.

by Jon Brodkin - Mar 25 2014, 2:45pm EST

NETWORKING SUPERCOMPUTING 34



Nvidia CEO Jen-Hsun Huang introduces Pascal at toda

Nvidia

Nvidia and IBM have developed an interconnect
units, letting GPUs and CPUs share data five tim
The fatter pipe will let data flow between the CP
compared to 16GB per second today.

## It Begins: AMD Announces Its First ARM Based Server SoC, 64-bit/8-core Opteron A1100

123 Comments

+ Add A Comment

by Anand Lal Shimpi on January 28, 2014 6:35 PM EST

Posted in CPUs IT Computing Enterprise enterprise CPUs AMD Opteron Opteron A1100 ARM

"SEATTLE" 64-BIT ARM SERVER PROCESSOR

FIRST 28NM ARM SERVER CPU TO SAMPLE IN MARCH

AMD

- Industry's only 64-bit ARM Server SoC from a proven server processor supplier
  - The most server experience of any ARM licensee

SAMPL

Around 15 mo
2014. Less tha
A1100: a 64-b

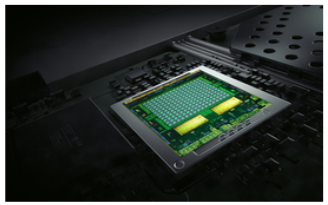
The Opteron A
talking about h
away entirely,
bets going on.

## Nvidia Jetson TK1 mini supercomputer is up for pre-order

Will ship on 15 May

By Lee Bell

Fri May 02 2014, 11:38



NVIDIA'S JETSON TK1 mini supercomputer development kit is now up for pre-order, priced at $192.

computers attempting to simulate human recognition
self-driving cars.

Speaking at the GPU Technology Conference (GTC)
Hsun Huang described it as capable of running anyth
can run, but at a slower pace.

## Intel to acquire Altera for $54 a share

Monday, 1 Jun 2015 | 8:33 AM ET

REUTERS

MarketWatch

PRESS RELEASE

## Altera and IBM Unveil FPGA-accelerated POWER Systems with Coherent Shared Memory

By

Published: Nov 17, 2014 8:00 a.m. ET

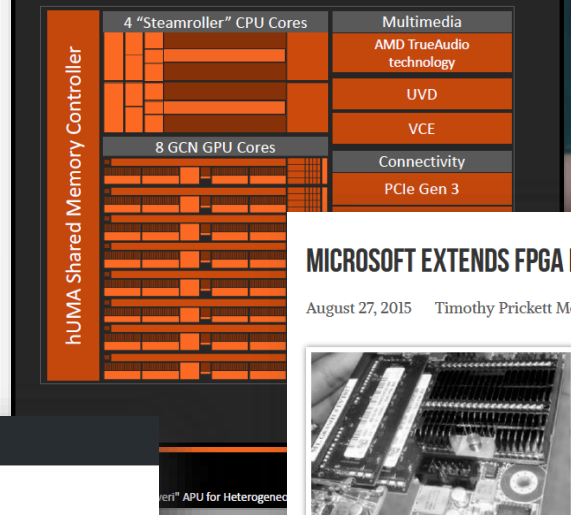
*POWER8 Systems that Leverage Reprogrammable FPGA Accelerators Gain Significant Improvements in System Performance, Efficiency and Flexibility*

NEW ORLEANS, Nov. 17, 2014 /PRNewswire/ -- **SuperComputing 2014** -- Altera Corporation ALTR, +0.00% and IBM IBM, +0.00% today unveiled the industry's first FPGA-based acceleration platform that coherently connects an FPGA to a POWER8 CPU leveraging IBM's Coherent Accelerator Processor Interface (CAPI). The reconfigurable hardware accelerator features shared virtual memory between the FPGA and processor which significantly improves system performance, efficiency and flexibility in high-performance computing (HPC) and data center applications. Altera and IBM are presenting several POWER8 systems that are coherently accelerated using FPGAs at SuperComputing 2014.

Working together through the OpenPOWER Foundation, Altera and IBM are

## A-SERIES REDEFINES COMPUTE

Kaveri

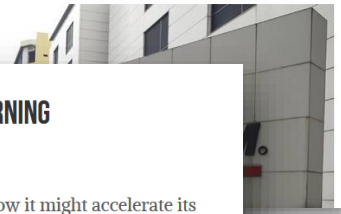



## Intel's 14nm Broadwell GPU takes shape, indicates major improvements over Haswell

By Sebastian Anthony on November 5, 2013 at 10:21 am | 16 Comments

## Avago Agrees to Buy Broadcom for $37 Billion

By MICHAEL J. de la MERCED and CHAD BRAY MAY 28, 2015



## MICROSOFT EXTENDS FPGA REACH FROM BING TO DEEP LEARNING

August 27, 2015 Timothy Prickett Morgan



After three years of research into how it might accelerate its Bing search engine using field programmable gate arrays (FPGAs), Microsoft came up with a scheme that would let it lash Stratix V devices from Altera to the two-socket server nodes in the minimalist Open Cloud Servers that it has designed expressly for its hyperscale datacenters. These CPU-FPGA hybrids were rolled out into production earlier this year to accelerate Bing page rank functions, and Microsoft started hunting around for other workloads with which to juice with FPGAs.

## Intel Mates FPGA With Future Xeon Server Chip

June 18, 2014 by Timothy Prickett Morgan


Intel is taking field programmable gate arrays seriously as a means of accelerating applications and has crafted a hybrid chip that marries an FPGA to a Xeon E5 processor and puts them in the same processor socket.



OAK RIDGE National Laboratory

# Current ASCR Computing At a Glance

| System attributes | NERSC Now | OLCF Now | ALCF Now | NERSC Upgrade | OLCF Upgrade | ALCF Upgrades | |
|---|---|---|---|---|---|---|---|
| Planned Installation | **Edison** | **TITAN** | **MIRA** | **Cori 2016** | **Summit 2017-2018** | **Theta 2016** | **Aurora 2018-2019** |
| System peak (PF) | 2.6 | 27 | 10 | > 30 | 150 | >8.5 | 180 |
| Peak Power (MW) | 2 | 9 | 4.8 | < 3.7 | 10 | 1.7 | 13 |
| Total system memory | 357 TB | 710TB | 768TB | ~1 PB DDR4 + High Bandwidth Memory (HBM)+1.5PB persistent memory | > 1.74 PB DDR4 + HBM + 2.8 PB persistent memory | >480 TB DDR4 + High Bandwidth Memory (HBM) | > 7 PB High Bandwidth On-Package Memory Local Memory and Persistent Memory |
| Node performance (TF) | 0.460 | 1.452 | 0.204 | > 3 | > 40 | > 3 | > 17 times Mira |
| Node processors | Intel Ivy Bridge | AMD Opteron Nvidia Kepler | 64-bit PowerPC A2 | Intel Knights Landing many core CPUs Intel Haswell CPU in data partition | Multiple IBM Power9 CPUs & multiple Nvidia Voltas GPUS | Intel Knights Landing Xeon Phi many core CPUs | Knights Hill Xeon Phi many core CPUs |
| System size (nodes) | 5,600 nodes | 18,688 nodes | 49,152 | 9,300 nodes 1,900 nodes in data partition | ~3,500 nodes | >2,500 nodes | >50,000 nodes |
| System Interconnect | Aries | Gemini | 5D Torus | Aries | Dual Rail EDR-IB | Aries | 2nd Generation Intel Omni-Path Architecture |
| File System | 7.6 PB 168 GB/s, Lustre® | 32 PB 1 TB/s, Lustre® | 26 PB 300 GB/s GPFS™ | 28 PB 744 GB/s Lustre® | 120 PB 1 TB/s GPFS™ | 10PB, 210 GB/s Lustre initial | 150 PB 1 TB/s Lustre® |

Binkley, ASCAC, April 2016

Complexity α T

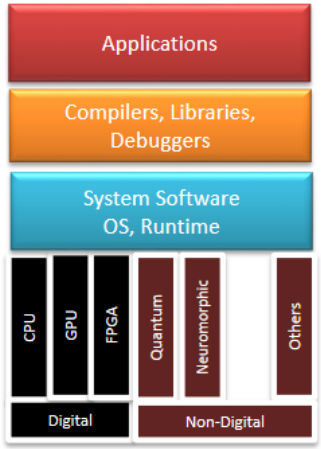
# DOE Workshop on Extreme Heterogeneity

23-25 Jan 2018

# Extreme Heterogeneity Workshop

## Jan. 23-25, 2018, in Gaithersburg, MD

- **POC: Lucy Nowell (Lucy.Nowell@science.doe.gov)**
- **Goal: Define challenges that extreme heterogeneity presents to the software stack and programming environment and identify related Computer Science priority research directions** that are essential to making extremely heterogeneous systems useful, usable and secure for science applications and DOE mission requirements in the 2025-2035 timeframe.
- 148 expected participants: DOE Labs, academia, & industry
- ~20 observers from DOE and other federal agencies (DoD, NSF, NASA)
- Pre-workshop report is being edited and will be posted by Jan. 1, 2018
- 105 white papers were received by the Dec. 4 deadline
  - After review, these resulted in 26 new invitations to Lab people and 20 to non-Lab people, including academics, industry and people from Europe and Japan.
- Agenda is being finalized, based in part on white paper content

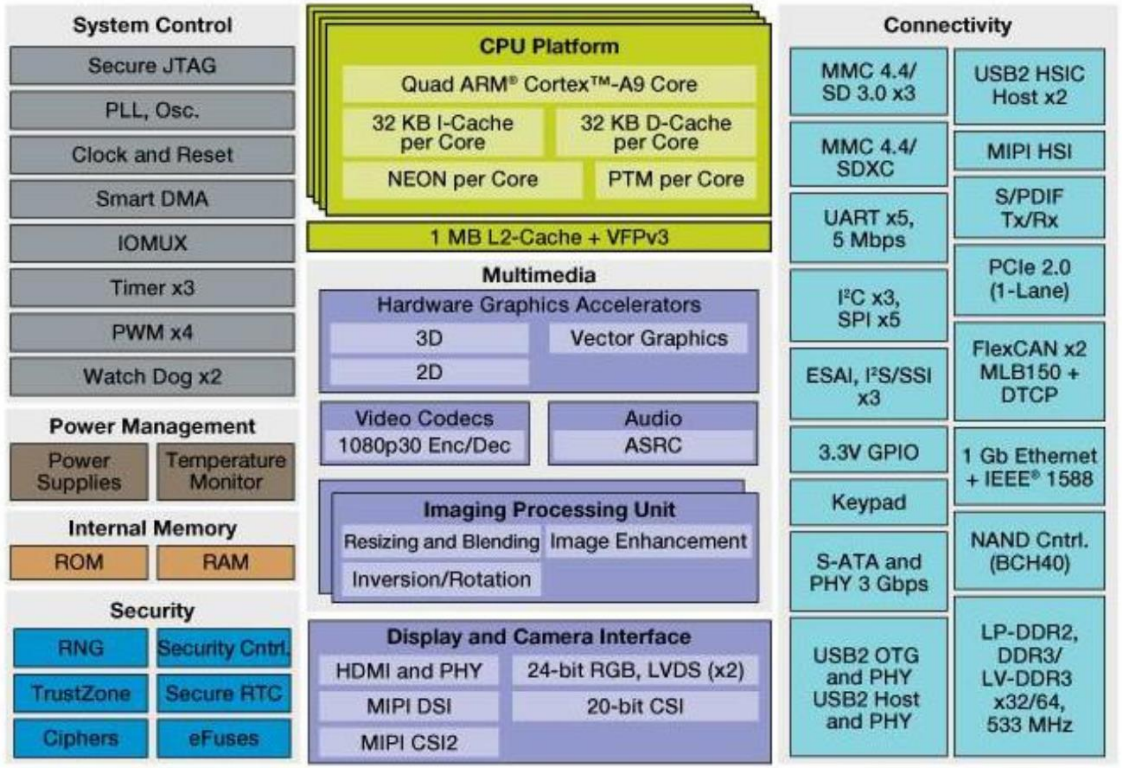





## Tuesday, January 23, 2018

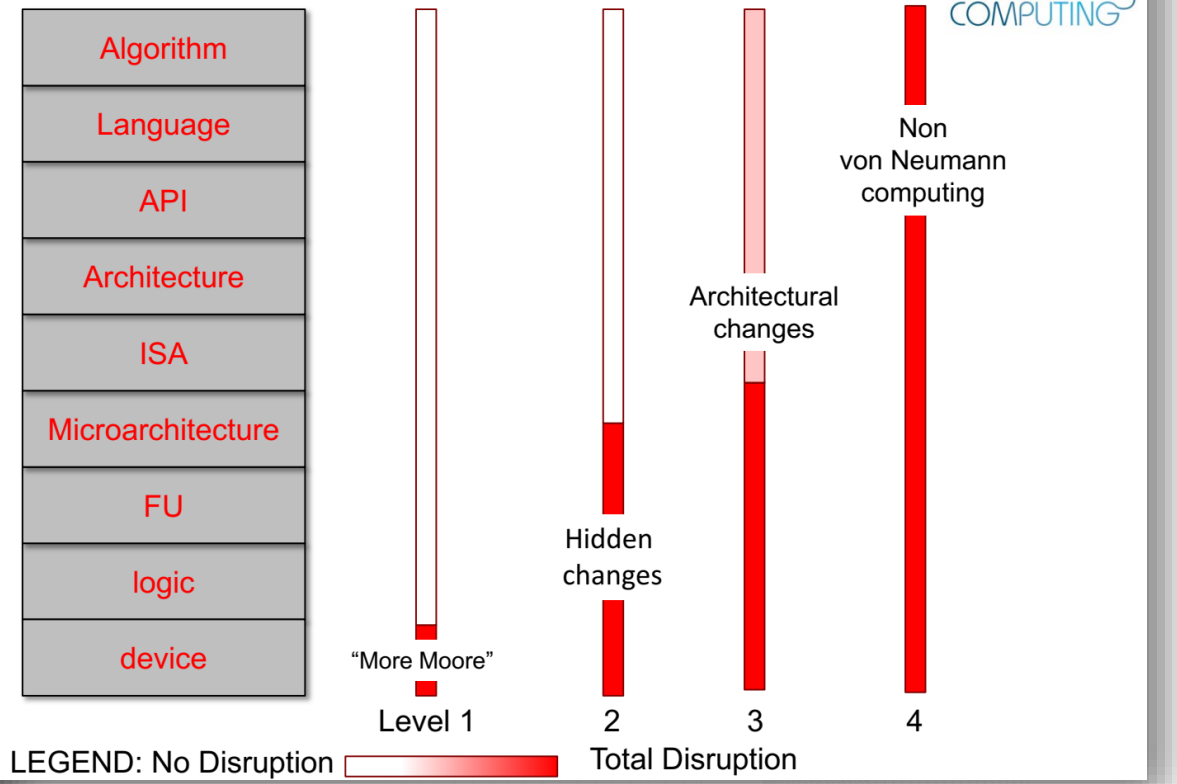
| | |
|---|---|
| 10:00 – 10:15 | Introductions: Lucy Nowell and Jeffrey Vetter |
| 10:15 – 10:35 | Welcome and ASCR Update – Barbara Helland, Director, Advanced Scientific Computing Research |
| 10:35– 11:05 | View from ASCR Research Division – Steve Lee, Acting Division Director |
| 11:05 – 11:35 | Invited Plenary Talk: IEEE Rebooting Computing - Tom Conte |
| 11:35– 11:45 | Break |
| 11:45 – 12:15 | Invited Talk: Architectural Trends and System Design Issues - Bob Colwell |
| 12:15– 1:30 | FSD Introduction to Extreme Heterogeneity – Jeffrey Vetter, John Shalf, and Maya Gokhale + FSD section owners |
| 1:45 – 2:30 | Break for lunch |
| 2:30 – 3:15 | Invited Talk: Report on the ASCAC future of computing study - Maya Gokhale |
| 3:15 – 4:30 | Panel on Issues Raised by Extreme Heterogeneity - Moderator Ron Brightwell |
| | Usability, Understandability and Programmability – Salman Habib |
| | Operating and Runtime Systems – Ron Brightwell |
| | Data Analytics – Wes Bethel |
| | EH Workflow Management – Ewa Deelman |
| | Open Q&A |

## Heterogeneity is already here: smartphone SoC ca 2016




Bob Colwell, Jan 2018

## Potential Approaches vs. Disruption in Computing Stack




Tom Conte, Jan 2018

# Workshop Organization and Topics

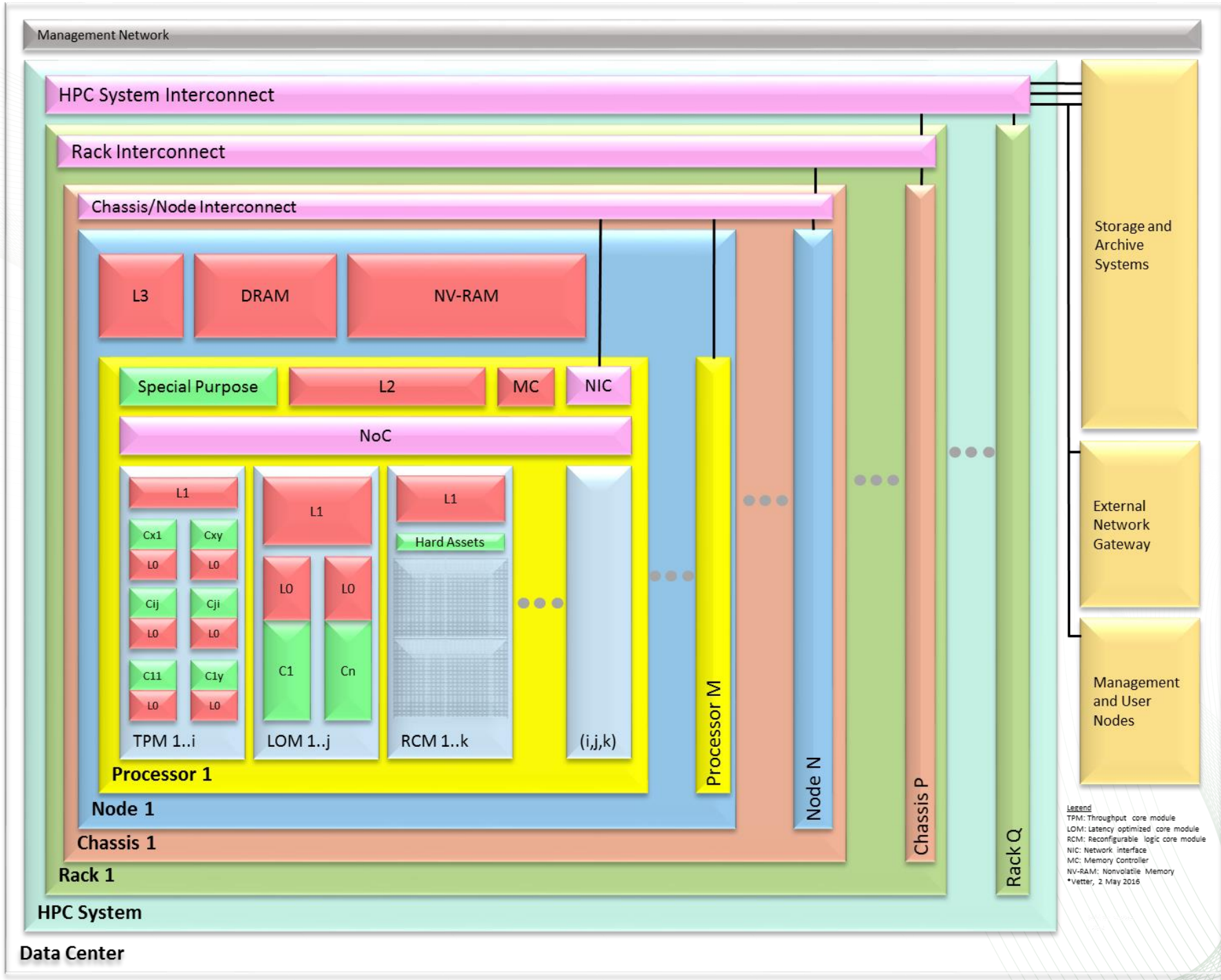
- Gov shutdown forced cancellation of physical meeting
- Moved to virtual meeting
  - Kept to original agenda (with some minor changes for timezones)
  - Approximately  200 participants viewed plenary sessions!!
- Breakout groups converged on priority research directions

| BOG ID | B/O Topic | Tu PM | Wed PM-1 | Wed PM-2 | Th AM | Th PM |
|---|---|---|---|---|---|---|
| 1 | Prog Env: Abstractions, Models, and Languages | Aiken/McCormick | | | McCormick | |
| 2 | Data Management and I/O | Ross/Byna | | | | Ross |
| 3 | Data Analytics and Workflows | Tom P./Yoo | | Christine S./Bethel | | |
| 4 | OS/RM: global, composition, workflow | | Brightwell | | | |
| 5 | Software Development Methodologies | | Li/Bernholdt | | | |
| 6 | Crosscut: Modeling and Simulation | | Chien/Donofrio/Leidel | | Wilke/Lan/Gokhale | |
| 7 | Prog Env: Compilers, Libraries, and Runtimes | | | Strout/Chapman | | |
| 8 | System Management, Admin, Job Scheduling | | | Peltz/Hartman-Baker | | |
| 9 | Crosscut: productivity, composition, interoperability | | | Lucas | | |
| 10 | OS/RM: local, prog env support | | | | Lang | |
| 11 | Crosscut: Portability, code reuse, performance portability | | | | Dubey/Li | |
| 12 | Prog Env: Debugging and Correctness, autotuning, specialization | | | | | Hall/Mellor-Crummey |
| 13 | Crosscut: resilience, power | | | | | Cappello/Cameron |

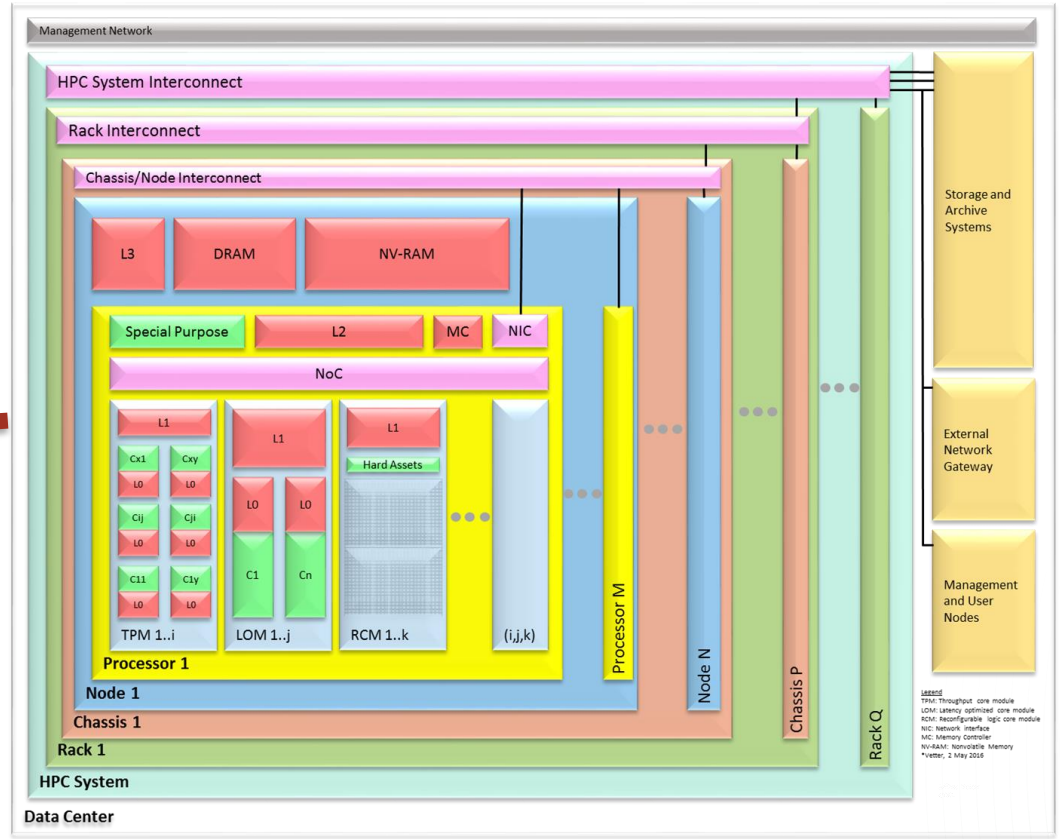
# Status

- Initial Priority Research Directions (Categories)
  - Programmability and Software Development Productivity
  - Execution, Scheduling in Runtime and OS
  - Reproducibility including Correctness, Debugging, Resilience
  - Modeling and Simulation for Performance, Power

- Working on report now… Stay tuned



# …Yields Complex Programming Models




- This approach is not scalable, affordable, robust, elegant, etc.
- Not performance portable

**System**: MPI, Legion, HPX, Charm++, etc

- Low overhead
- Resource contention
- Locality

**Node**: OpenMP, Pthreads, U-threads, etc

- SIMD
- NUMA, HBM

**Cores**: OpenACC, CUDA, OpenCL, OpenMP4, …

- Memory use, coalescing
- Data orchestration
- Fine grained parallelism
- Hardware features

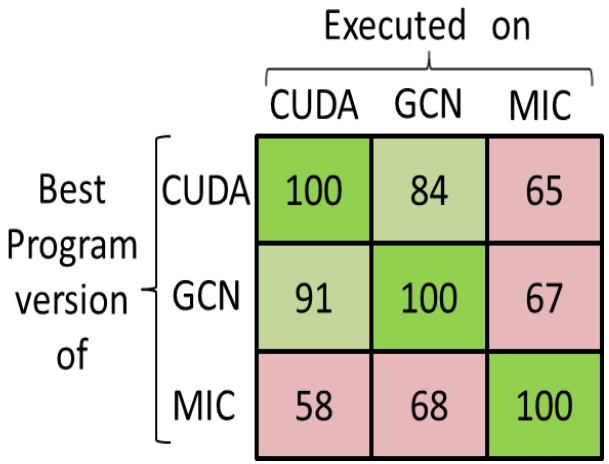
# Our Approach – Try to map FPGAs into an Existing Programming Framework

# Standard, Portable Programming Models for Heterogeneous Computing

- OpenCL
  - Open standard portable across diverse heterogeneous platforms (e.g., CPUs, GPUs, DSPs, Xeon Phis, FPGAs, etc.)
  - Much higher than HDL, but still complex for typical programmers.
- Directive-based accelerator programming models
  - OpenACC, OpenMP4, etc.
  - Provide higher abstraction than OpenCL.
  - Most of existing OpenACC/OpenMP4 compilers target only specific architectures; none supports FPGAs.

# Performance Portability of High-level Programming Models for Contemporary Heterogeneous Systems

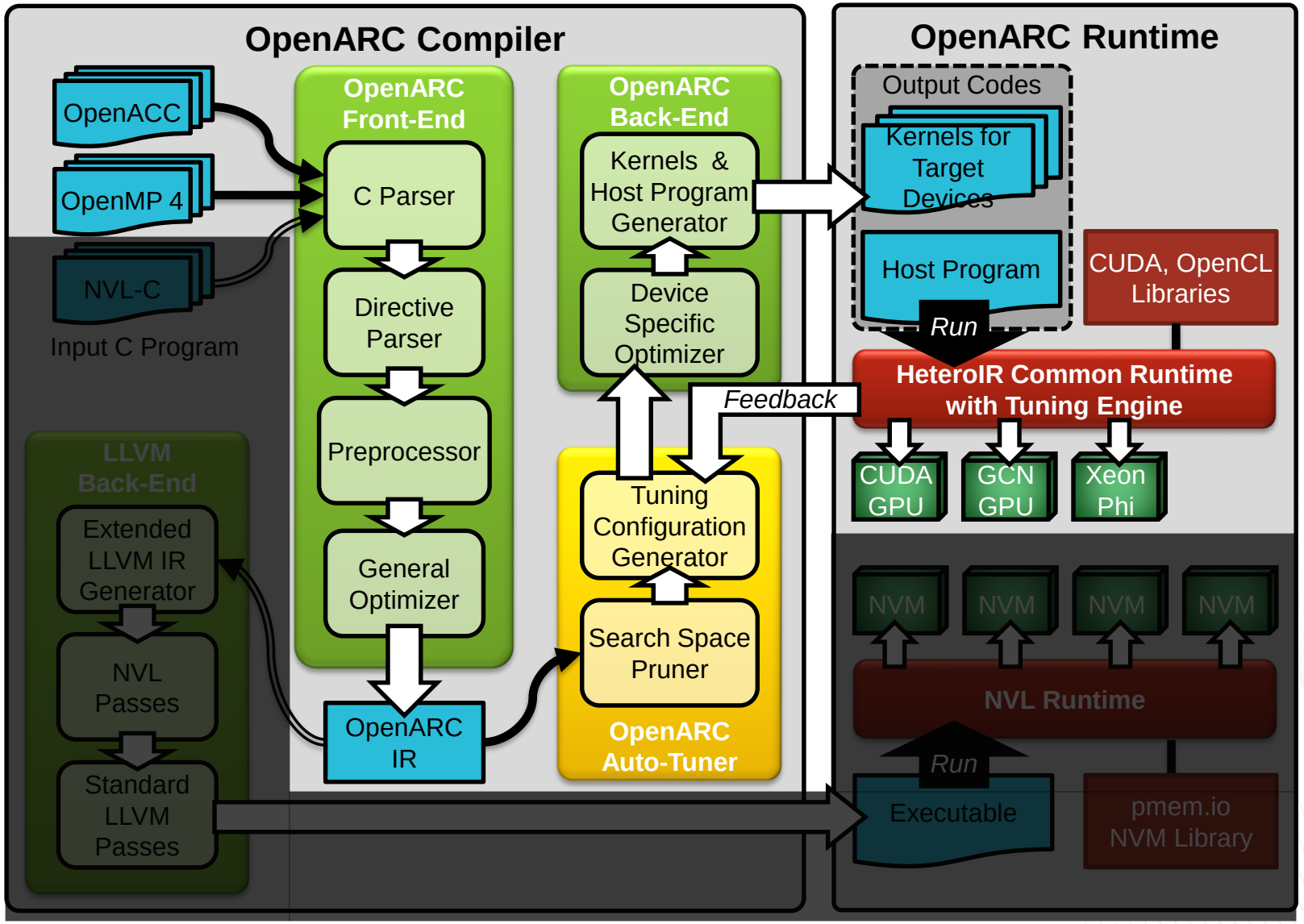
- Problem
  - Directive-based, high-level accelerator programming models such as OpenACC provide code portability.
    - How does it fare on performance portability?
    - And what architectural features/compiler optimizations affect the performance portability? And how much?
- Solution
  - Proposed a high-level, architecture-independent intermediate language (HeteroIR) to map high-level programming models (e.g., OpenACC) to diverse heterogeneous devices while maintaining portability.
  - Using HeteroIR, port and measure the performance portability of various OpenACC applications on diverse architectures.




- Results
  - Using HeteroIR, OpenARC ported 12 OpenACC applications to diverse architectures (NVIDIA CUDA, AMD GCN, and Intel MIC), and measured the performance portability achieved across all applications.
  - HeteroIR abstracts out the common architecture functionalities, which makes it easy for OpenARC (and other compilers) to support diverse heterogeneous architectures.
  - HeteroIR, combined with rich OpenARC directives and built-in tuning tools, allows OpenARC to be used for various tuning studies on diverse architectures.

A. Sabne, P. Sakdhnagool *et al.*, *"Understanding Portability of a High-Level Programming Model on Contemporary Heterogeneous Architectures," IEEE Micro, 35(4):48-58, 2015, 10.1109/MM.2015.73.*

# OpenARC System Architecture



# Intelligent selection of optimizations based on target architecture

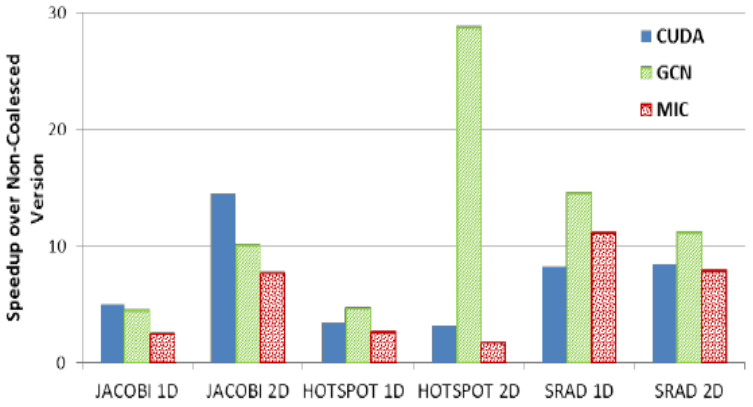



Figure 5: Memory Coalescing Benefits on Different Architectures : MIC is impacted the least by the non-coalesced accesses

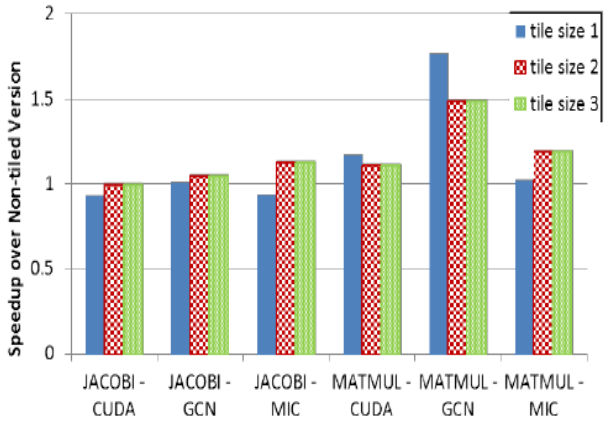



Figure 7: Impact of Tiling Transformation : *MATMUL* shows higher benefits than *JACOBI* owing to more contiguous accesses

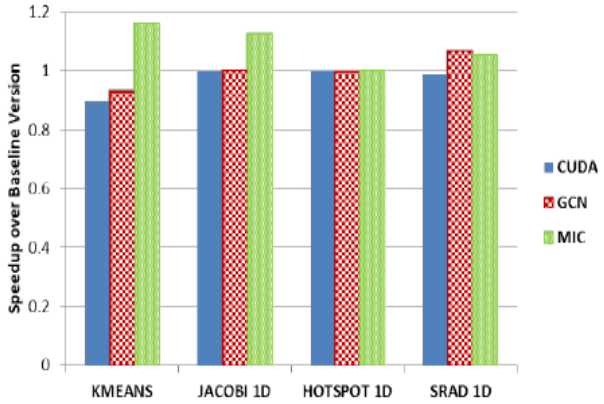



Figure 9: Effects of Loop Unrolling - MIC shows benefits on unrolling

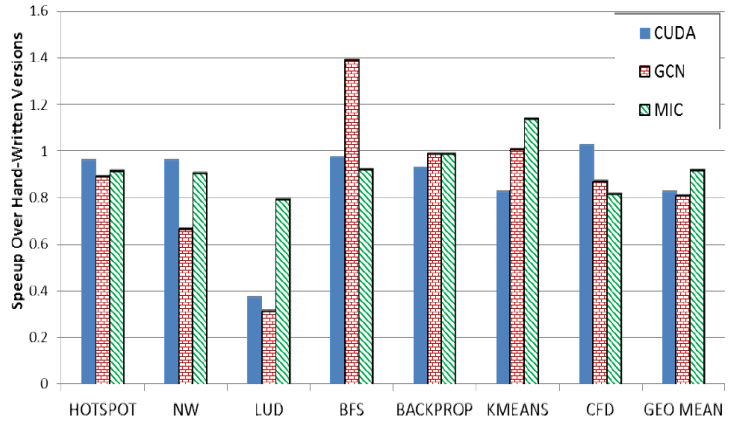



Fig. 11: Comparison of hand-written CUDA/OpenCL programs against auto-tuned OpenARC code versions : Tuned OpenACC programs perform reasonably well against hand-written codes

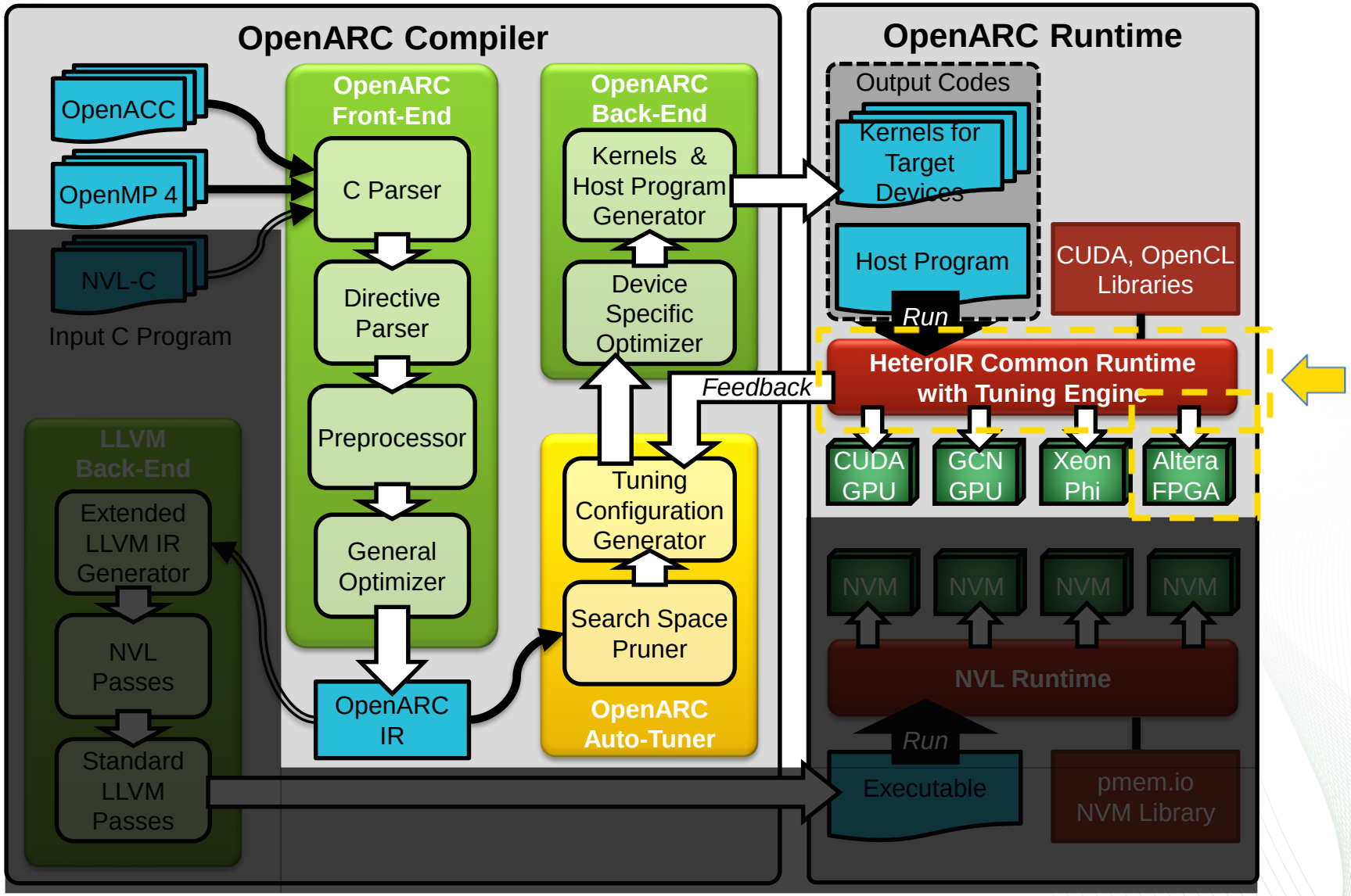
# Approach for FPGA support

- Design and implement an OpenACC-to-FPGA translation framework, which is the first work to use a standard and portable directive-based, high-level programming system for FPGAs.
- Propose FPGA-specific optimizations and novel pragma extensions to improve performance.
- Evaluate the functional and performance portability of the framework across diverse architectures (Altera FPGA, NVIDIA GPU, AMD GPU, and Intel Xeon Phi).



S. Lee, J. Kim, and J.S. Vetter, "OpenACC to FPGA: A Framework for Directive-based High-Performance Reconfigurable Computing," Proc. IEEE International Parallel & Distributed Processing Symposium (IPDPS), 2016, 10.1109/IPDPS.2016.28.

# OpenARC System Architecture



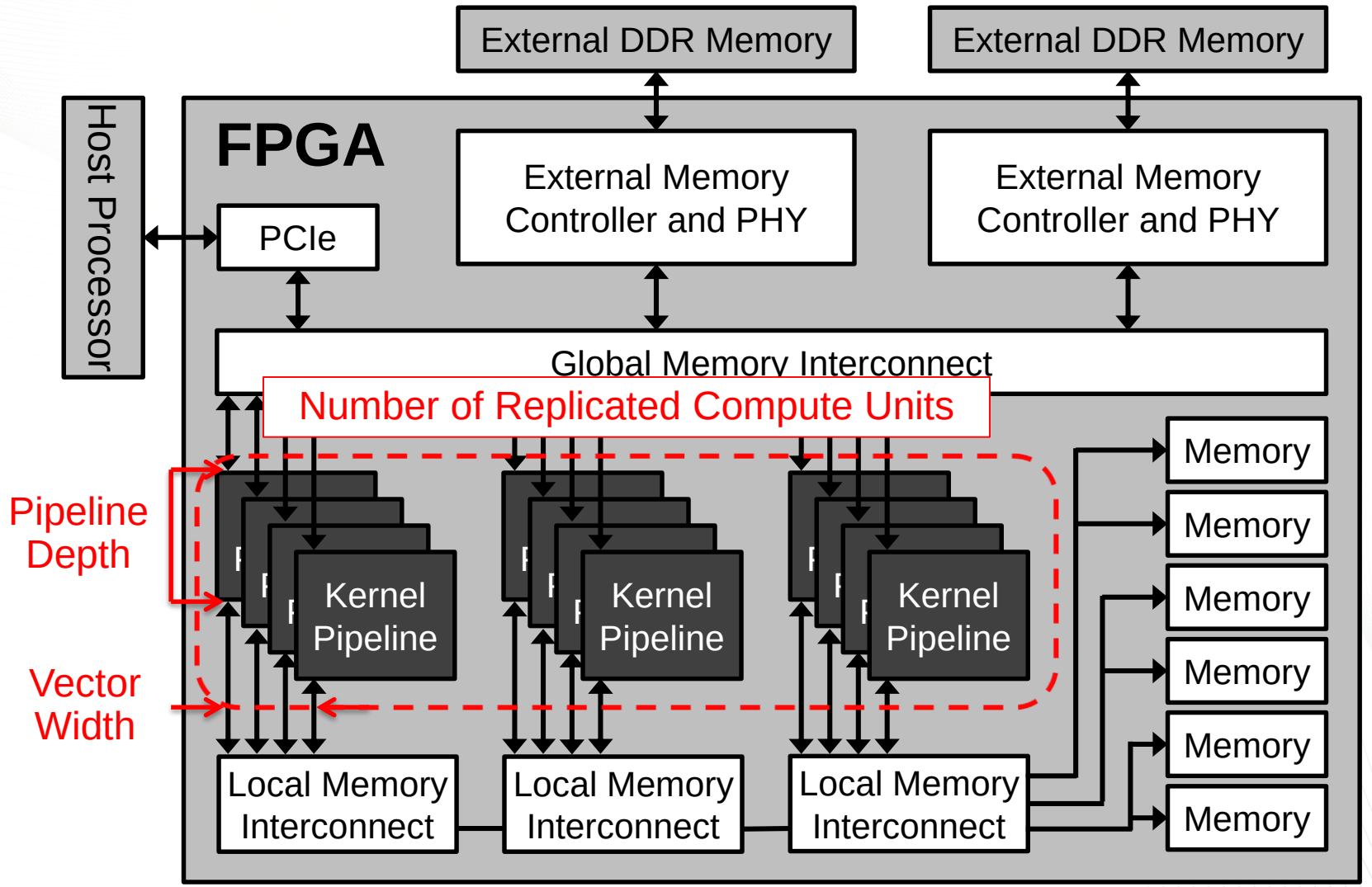
# Baseline Translation of OpenACC-to-FPGA

- Use OpenCL as the output model and the Altera Offline Compiler (AOC) as its backend compiler.
- Translates the input OpenACC program into a host code containing HeteroIR constructs and device-specific kernel codes.
  - Use the same HeteroIR runtime system of the existing OpenCL backends, except for the device initialization.
  - Reuse most of compiler passes for kernel generation.

# OpenARC Extensions and Optimizations for Efficient FPGA Programming

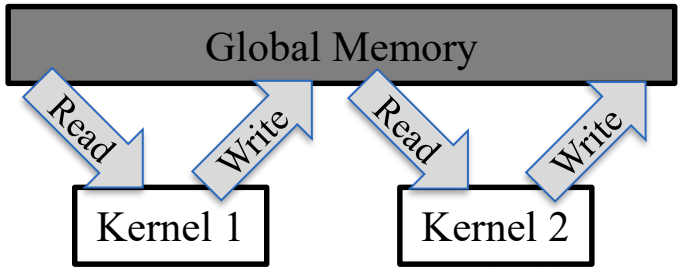
- Key benefit of using FPGAs is that they support wide, heterogeneous, and deeply pipelined parallelism customized for the input program.
- In FPGA programming with OpenCL, the OpenCL compiler synthesizes all the hardware logic for the input program.
  - The efficiency of the compiler is critical.
- We extend OpenARC to generate output OpenCL codes in a manner friendly to the underlying AOC OpenCL backend compiler.

# FPGA OpenCL Architecture

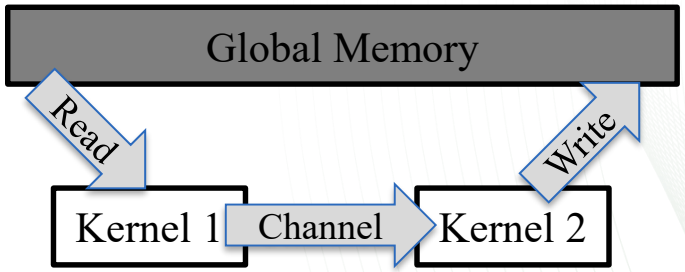


# Kernel-Pipelining Transformation Optimization

- Kernel execution model in OpenACC
  - Device kernels can communicate with each other only through the device global memory.
  - Synchronizations between kernels are at the granularity of a kernel execution.
- Altera OpenCL channels
  - Allows passing data between kernels and synchronizing kernels with high efficiency and low latency




Kernel communications through global memory in OpenACC

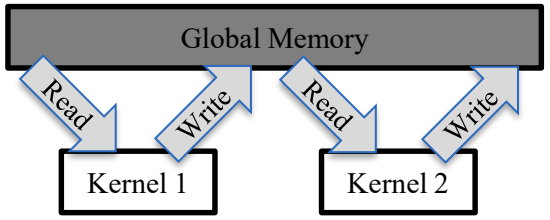



Kernel communications with Altera channels

# Kernel-Pipelining Transformation Optimization (2)

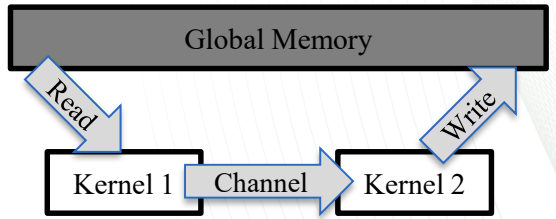
(a) Input OpenACC code

```
#pragma acc data copyin (a) create (b) copyout (c)
{
   #pragma acc kernels loop gang worker present (a, b)
   for(i=0; i<N; i++) { b[i] = a[i]*a[i]; }
   #pragma acc kernels loop gang worker present (b, c)
   for(i=0; i<N; i++) {c[i] = b[i]; }
}
```




(b) Altera OpenCL code with channels

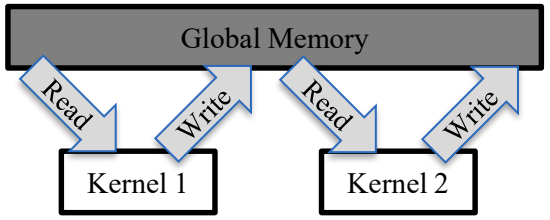
```
channel float pipe_b;
__kernel void kernel1(__global float* a) {
   int i = get_global_id(0);
   write_channel_altera(pipe_b, a[i]*a[i]);
}
__kernel void kernel2(__global float* c) {
   int i = get_global_id(0);
   c[i] = read_channel_altera(pipe_b);
}
```



# Kernel-Pipelining Transformation Optimization (3)

(a) Input OpenACC code

```
#pragma acc data copyin (a) create (b) copyout (c)
{
   #pragma acc kernels loop gang worker present (a, b)
   for(i=0; i<N; i++) { b[i] = a[i]*a[i]; }
   #pragma acc kernels loop gang worker present (b, c)
   for(i=0; i<N; i++) {c[i] = b[i]; }
}
```

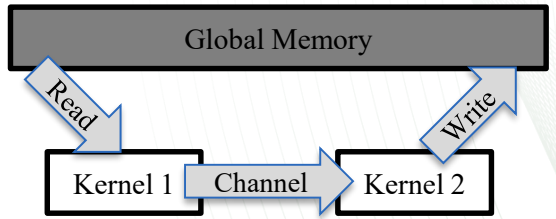



Kernel-pipelining transformation

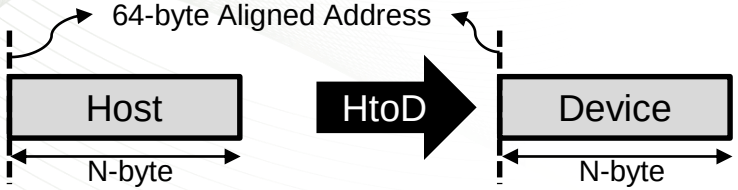
Valid under specific conditions

(c) Modified OpenACC code for kernel-pipelining

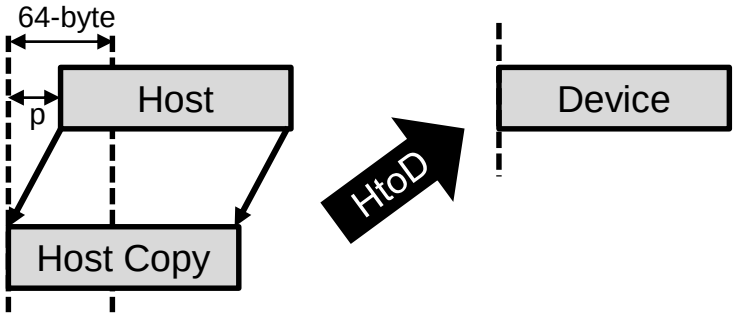
```
#pragma acc data copyin (a) pipe (b) copyout (c)
{
   #pragma acc kernels loop gang worker pipeout (b) present (a)
   For(i=0; i<N; i++) { b[i] = a[i]*a[i]; }
   #pragma acc kernels loop gang worker pipein (b) present (c)
   For(i=0; i<N; i++) {c[i] = b[i];}
}
```



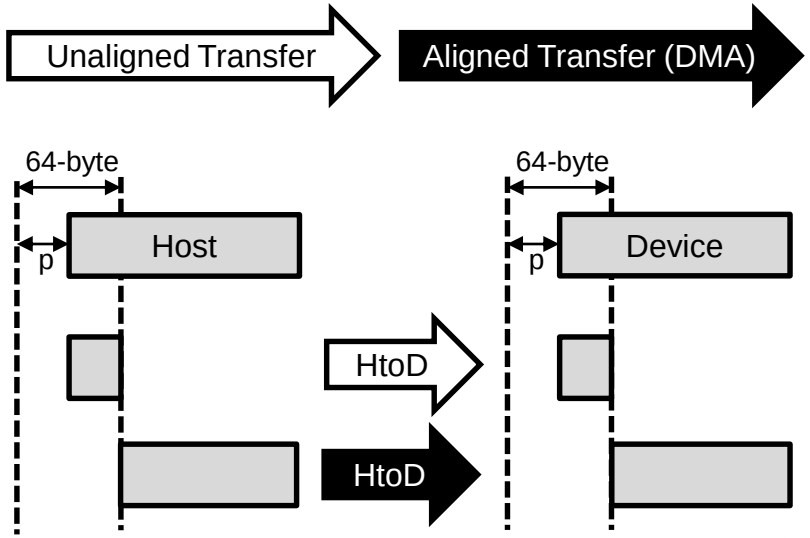
# Dynamic Memory-Transfer Alignment Optimization (2)




(a) Aligned-host & Aligned-device

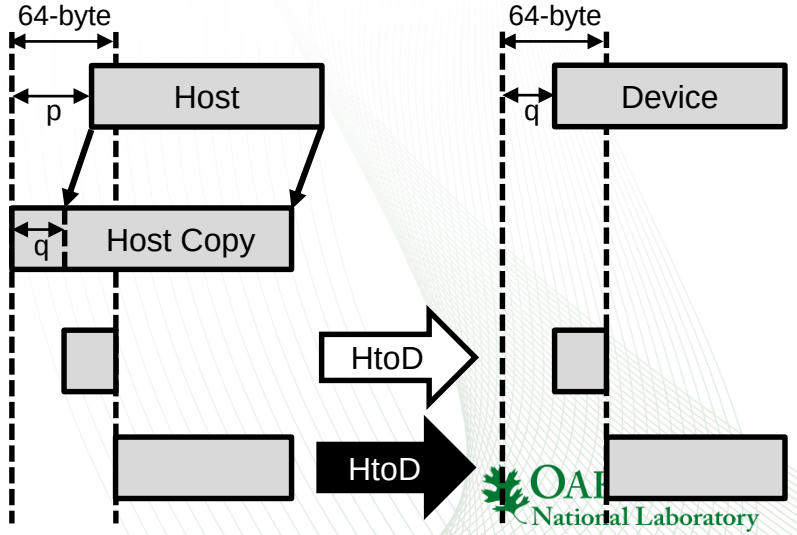



(b) Unaligned-host with Offset p (0 < p < 64) & Aligned-device




(c) Unaligned-host & Unaligned-device with the Same Offset p (0 < p < 64)

(d) Aligned- or Unaligned-host with Offset p ($0 \leq p < 64$) & Unaligned-device with Offset q ($0 < q < 64$)



# Application Used

OpenARC Compiler Suite | Rodinia Benchmark Suite | Altera SDK for OpenCL

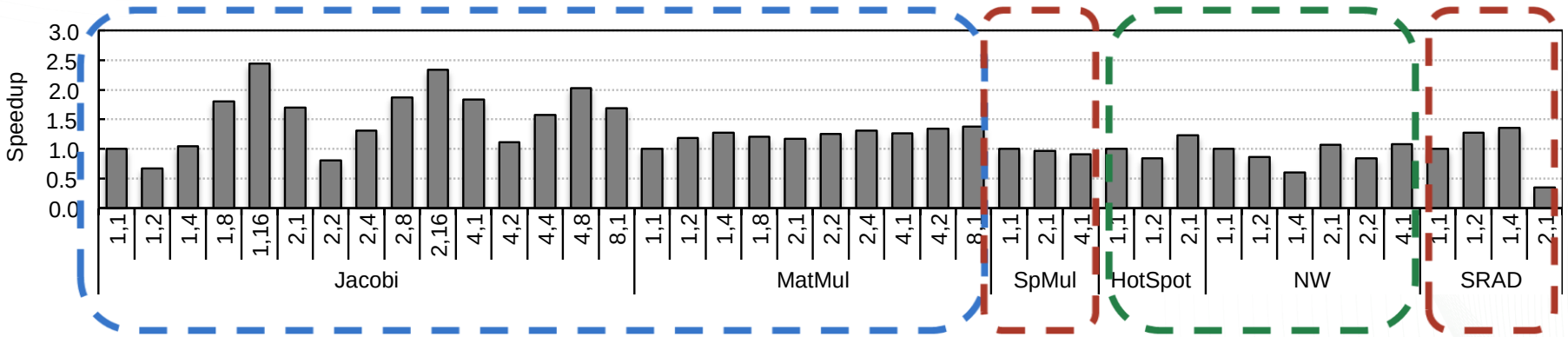
| Application | Description | Input | A | B | C | D | E |
|---|---|---|---|---|---|---|---|
| Jacobi | Jacobi iterative method | 8192x8192, 10 iters | X | | X | | |
| MatMul | Dense matrix multiplication | 2048x2048 | X | | X | | |
| SpMul | Sparse matrix multiplication | 2063494 x 2063494 | X | X | | | |
| HotSpot | Compact thermal modeling | 1024x1024, 1000 iters | X | | | | |
| NW | Needleman-Wunsch algorithm | 8192x8192 | | | | | |
| SRAD | Speckle reducing anisotropic diffusion | 8192x8192 | X | | | | |
| FFT-1D | 1D radix-4 complex fast Fourier transform | 4096, 100 iters | X | | | X | |
| FFT-2D | 2D radix-4 complex fast Fourier transform | 256x256 | X | | | X | X |

A: Boundary check elimination, B: Work-item ID-dependent backward branching, C: Loop unrolling, D: Single work-item kernel, E: Kernel pipelining

# Speedup over CU, SIMD (1,1)

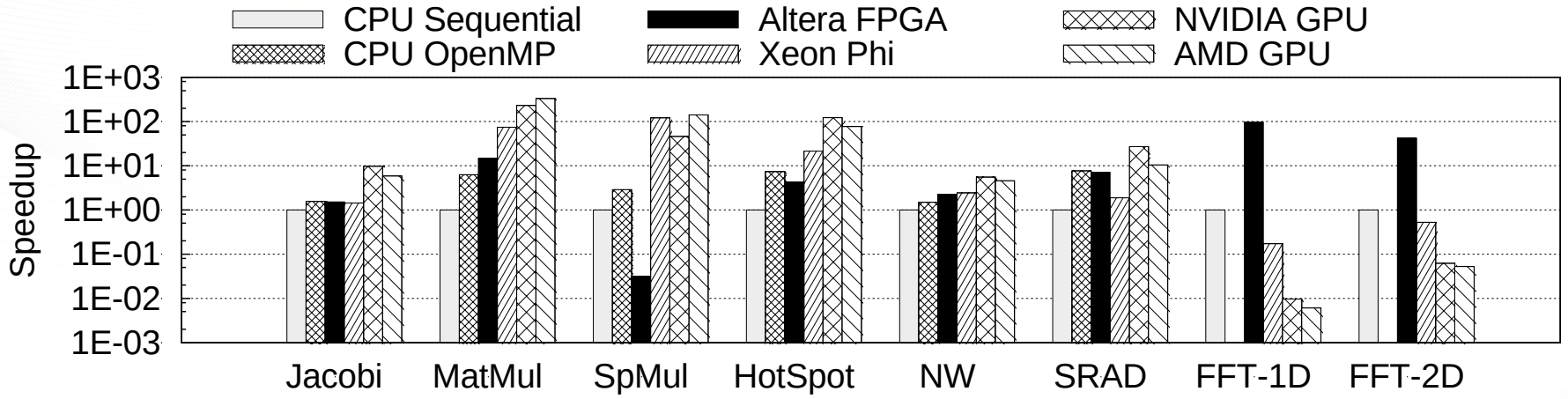
Jacobi and MatMul show better performance with increase in CU and SIMD, thanks to regular memory accesses.

SpMul and SRAD perform worse with multiple CUs, mainly due to memory contention.




Performance of HotSpot and NW increases with multiple CUs, but decreases with vectorization.

# Overall Performance




FPGAs prefer applications with deep execution pipelines (e.g., FFT-1D and FFT-2D), performing much higher than other accelerators.

For traditional HPC applications with abundant parallel floating-point operations, it seems to be difficult for FPGAs to beat the performance of other accelerators, even though FPGAs can be much more power-efficient.

- Tested FPGA does not contain dedicated, embedded floating-point cores, while others have fully-optimized floating-point computation units.

Current and upcoming high-end FPGAs are equipped with hardened floating-point operators, whose performance will be comparable to other accelerators, while remaining power-efficient.

# Hardware Resource Utilization (%)

Hardware resource utilization (%) depending on the number of the replicated compute units (CUs) and SIMD width in the kernel vectorization

| App | Number of the replicated CUs, SIMD width in the kernel vectorization | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1,1 | 1,2 | 1,4 | 1,8 | 1,16 | 2,1 | 2,2 | 2,4 | 2,8 | 2,16 | 4.1 | 4,2 | 4,4 | 4,8 | 4,16 | 8,1 | 8,2 |
| Jacobi | 29 | 33 | 37 | 41 | 49 | 36 | 43 | 51 | 59 | 74 | 48 | 62 | 78 | 95 | 124 | 73 | 101 |
| MatMul | 28 | 34 | 45 | 67 | 109 | 35 | 46 | 68 | 110 | 195 | 48 | 69 | 112 | 197 | 367 | 72 | 115 |
| SpMul | 35 | - | - | - | - | 46 | - | - | - | - | 69 | - | - | - | - | 114 | - |
| HotSpot | 56 | 79 | 124 | 214 | 443 | 89 | 134 | 224 | 445 | 863 | 154 | 245 | 467 | 866 | 1704 | 285 | 518 |
| NW | 35 | 46 | 68 | 112 | 200 | 46 | 68 | 112 | 200 | 377 | 69 | 113 | 201 | 377 | 730 | 115 | 202 |
| SRAD | 54 | 65 | 80 | 110 | 170 | 84 | 106 | 136 | 197 | 317 | 145 | 189 | 249 | 370 | 621 | 266 | 354 |
| FFT-1D | 80 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |
| FFT-2D | 56 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - | - |

# of CU affects the resource utilization more than the SIMD width.

If a resource utilization is larger than 100%, the compiler cannot generate kernel execution file.

# Additional optimizations to exploit FPGA architectural features within this directive-based framework

- *Pipe directive to reduce accesses to global memory*
- Collapse directive to put more work into deep pipelines
- Use shift registers
  - Sliding window
  - Reductions

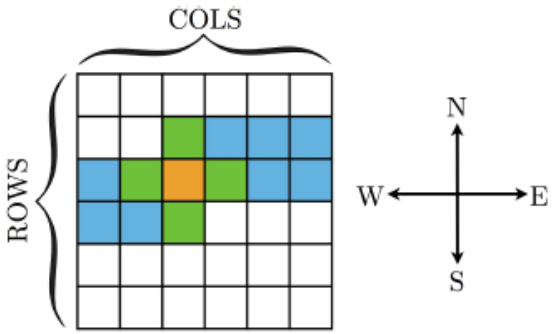



Fig. 1: Sliding-window-based stencil computation example. The blue cells represent the sliding window, the green cells represent the constant access points, and the orange cell represents the target element.

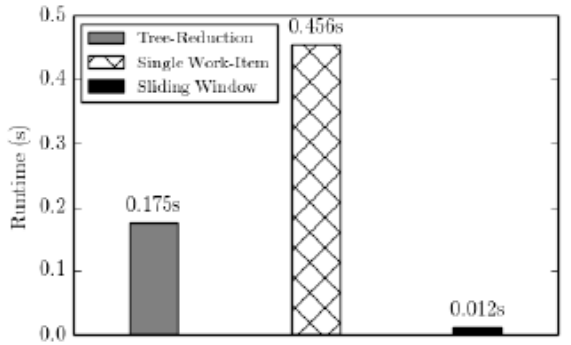



Fig. 3: SRAD performance comparison of tree reduction, basic single-threaded reduction, and single-threaded shift register reduction.

# Recap

- Recent trends in extreme-scale HPC paint an uncertain future
  - Complexity is our main challenge
- Applications and software systems are all reaching a state of crisis
  - Applications will not be functionally or performance portable across architectures
- Programming systems must provide performance portability (beyond functional portability)!!
  - Reconfigurable systems are the ultimate challenge ☺
- Extending OpenACC to target FPGAs
  - Extend standard programming model to include FPGA targets
  - Addressing shortcomings with extensions to OpenACC and compiler infrastructure
  - Promising results with initial prototypes
- Push these improvements into OpenACC/OpenMP

# Acknowledgements


- Contributors and Sponsors
  - Future Technologies Group: http://ft.ornl.gov
  - US Department of Energy Office of Science
    - DOE Vancouver Project: https://ft.ornl.gov/trac/vancouver
    - DOE Blackcomb Project: https://ft.ornl.gov/trac/blackcomb
    - DOE ExMatEx Codesign Center: http://codesign.lanl.gov
    - DOE Cesar Codesign Center: http://cesar.mcs.anl.gov/
    - DOE Exascale Efforts: http://science.energy.gov/ascr/research/computer-science/
  - Scalable Heterogeneous Computing Benchmark team: http://bit.ly/shocmarx
  - US National Science Foundation Keeneland Project: http://keeneland.gatech.edu
  - US DARPA
  - NVIDIA CUDA Center of Excellence