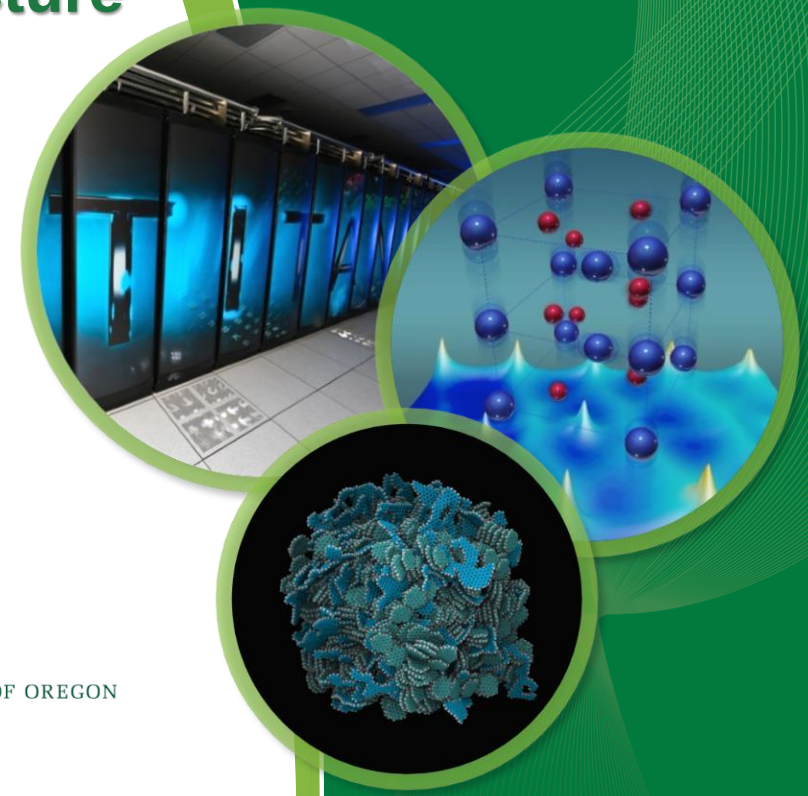
Vancouver: Designing a Next-Generation Software Infrastructure for Productive Heterogeneous Exascale Computing



Jeffrey S. Vetter (ORNL), PI
Allen Malony (Oregon), Co-PI
Wen-Mei Hwu (UIUC), Co-PI
Rich Vuduc (GT), Co-PI
Seyong Lee (ORNL)
Jungwon Kim (ORNL)
Joel Denny (ORNL)
Kittisak Sajjapongse (ORNL)
Sameer Shende (Oregon)
Nicholas Chaimov (Oregon)
Robert Lim (Oregon)
Kevin Huck (Oregon)
John Larson (UIUC)
Carl Pearson (UIUC)
Liwen Chang (UIUC)

UNIVERSITY OF OREGON

ILLINOIS

Georgia Tech College of Computing
Computational Science and Engineering

http://ft.ornl.gov/trac/vancouver

ORNL is managed by UT-Battelle
for the US Department of Energy

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

OAK RIDGE National Laboratory

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®

Recent SHC Announcements

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 - M

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

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

123 Comments

+ Add Comment

"SEATTLE" 64-BIT ARM SERVER PROCESSOR

FIRST 28NM AR

Industry prove

- The
- Serv

CPU

- 2-4x impr
- 8 cor
- Base
- Exte CPU
- Serv
- Lega
- Stora

SAMPLI

Nvidia CEO Jen-H

Nvidia

Nvidia and IBM

units, letting GP

The fatter pipe

compared to 16

Nvidia Jetson TK1 mini supercomputer is up for pre-order

Will ship on 15 May

By Lee Bell

Fri May 02 2014, 11

computers attempt self-driving cars.

Speaking at the GP Hsun Huang descri can run, but at a sl

With a total perfor Raspberry Pi board in the US - a numb launched at CES in

"The Jetson TK1 als comes with a whole it could be classifyi

Parameters are loa recognises objects,

Around 15 mon 2014. Less than A1100: a 64-bit

The Opteron A1 talking about ha away entirely, b bets going on. E process at Glob

A-SERIES REDEFINES COMPUTE

Kaveri

4 "Steamroller" CPU Cores

8 GCN GPU Cores

Multi

AMD Tr techn

UV

VC

Conne

PCIe

PCIe

Displ

DisplayPort 1.2

HDMI 1.4a

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

f 8 13 8+

Aa

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

NEW ORLEANS, Nov. 17, 2014 /PRNewswire/

Corporation ALTR, +0.00% and IBM IBM, +0

FPGA-based acceleration platform that cohe

CPU leveraging IBM's Coherent Accelerator

reconfigurable hardware accelerator feature

FPGA and processor which significantly imp

flexibility in high-performance computing (H

and IBM are presenting several POWER8 sys

using FPGAs at SuperComputing 2014.

Working together through the OpenPOWER

Intel to acquire Altera for $54 a share

Monday, 1 Jun 2015 | 8:33 AM ET

REUTERS

Intel Mates FPGA With Future Xeon Server Chip

June 18, 2014 by Timothy Prickett Morgan

inside

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.

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

BROADCOM

18

has started open-sourcing the Linux driver for Broadwell's GPU. Broadwell is the 14nm die shrink of Intel's

nd while the CPU

xpected to

lwell's GPU looks

es (Iris) GPU

modified

GPU, and the

OAK RIDGE National Laboratory

Vancouver: Designing a Next-Generation Software Infrastructure for Productive Heterogeneous Exascale Computing

- **Funded in X-Stack call in 2010: LAB 10-257**
  - Renewed in 2013
- **Focus on Scalable Heterogeneous Computing (SHC)**
  - In 2010, no SHC productions systems existed in DOE ASCR
  - SHC systems offered high performance, energy efficiency, and density
  - However, also had
  - challenges of low productivity,
  - poor portability,
  - lack of tools and libraries,
  - performance sensitivities
- **Performance tools and benchmarks (SHOC, TAU, Parboil)**
  - Benchmarking, Code analysis, inspection, transformation
- **Create next-generation of tools to develop and understand SHC applications**
  - Design high-level abstractions (OpenARC, MxPA, Tanagram)
- **Directive-based SHC programming models w/ multi-level parallelism**
- **Implement libraries, tools, and runtime systems (autotuning SuperLU)**
- **Focus on DOE SHC applications and systems (many)**
- **Tutorials and Hackathons at conferences, various sites**

Recent Highlights

- Languages
  - OpenARC performance portability studies across NVIDIA, AMD, Xeon Phi
    - Altera FPGA
  - Intelligent compiler selection of optimizations
- Benchmarks
  - Improvements to SHOC suite: OpenACC, more kernels
- Performance Tools
  - Performance measurement and analysis support for latest manycore accelerators (NVIDIA GPUs) and coprocessors (Intel KNC (KNL)) fully integrated in TAU Performance System
  - GPU performance prediction techniques based on static code analysis (instruction mix, control flow graph, occupancy) and dynamic analysis (instruction counts, branch frequency, memory reuse) and techniques with autotuning frameworks.
- Algorithms and autotuning
  - first hybrid, distributed memory CPU+co-processor (GPU + Phi) sparse direct solver, with on-going integration into SuperLU_DIST jointly with Sherry Li @ LBNL
- Code Synthesis
  - High-performance, energy efficient, and portable code synthesis with Tangram
  - OpenCL performance portability for RSBench between CPUs and GPUs with MxPA

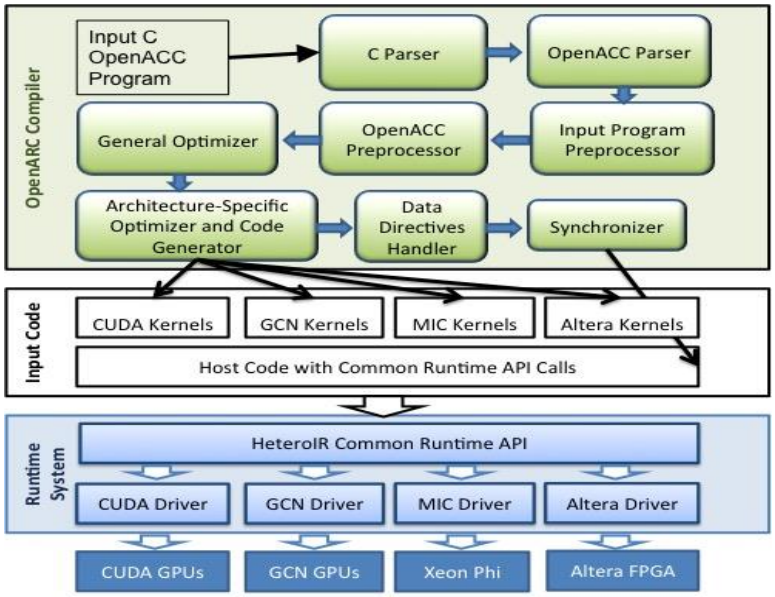
Programming Models for Heterogeneous Systems

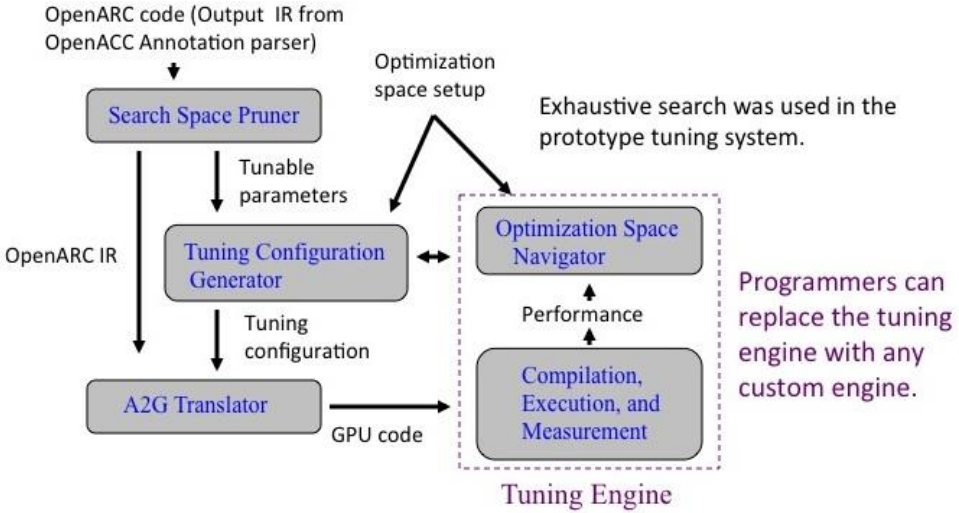
OAK RIDGE National Laboratory

http://ft.ornl.gov/research/openarc

OpenARC: Open Accelerator Research Compiler

- Problem
  - Directive-based accelerator programming models provide abstraction over architectural details and low-level programming complexities. However, too much abstraction puts significant burdens on performance tuning, debugging, and scaling.
- Solution
  - OpenARC is *an open-sourced, very High-level Intermediate Representation (HIR)-based, extensible compiler framework*, where various performance optimizations, traceability mechanisms, fault tolerance techniques, etc., can be built for better debuggability/performance/resilience on the complex accelerator computing.
- Tech Transfer
  - Deployed to Argonne National Laboratory, Los Alamos National Laboratory, IBM France, IBM US, Barcelona Supercomputing Center, University of La Laguna, Tokyo Institute of Technology, Auburn University, Duke University, Purdue University, etc.
  - Participate in OpenACC technical committee





S. Lee and J.S. Vetter, "OpenARC: Open Accelerator Research Compiler for Directive-Based, Efficient Heterogeneous Computing," in *ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC). Vancouver: ACM, 2014*

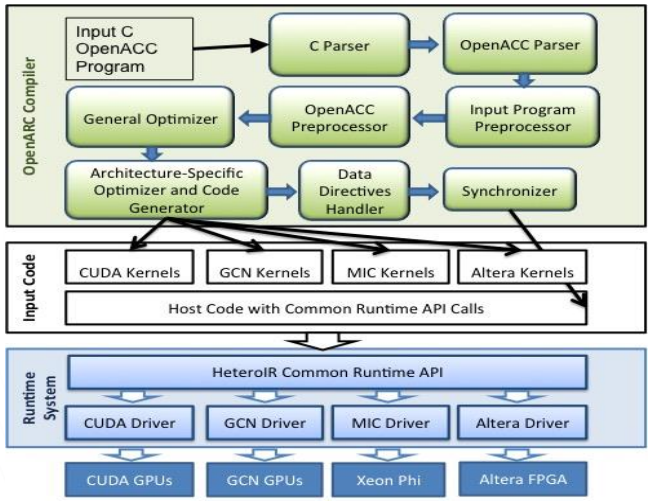
Understanding Performance Portability of High-level Programming Models for Heterogeneous Systems

- Problem

  Directive-based, high-level accelerator programming models such as OpenACC provide code portability. But how does it fare on performance portability? And what architectural features/compiler optimizations affect the performance portability? And how much?

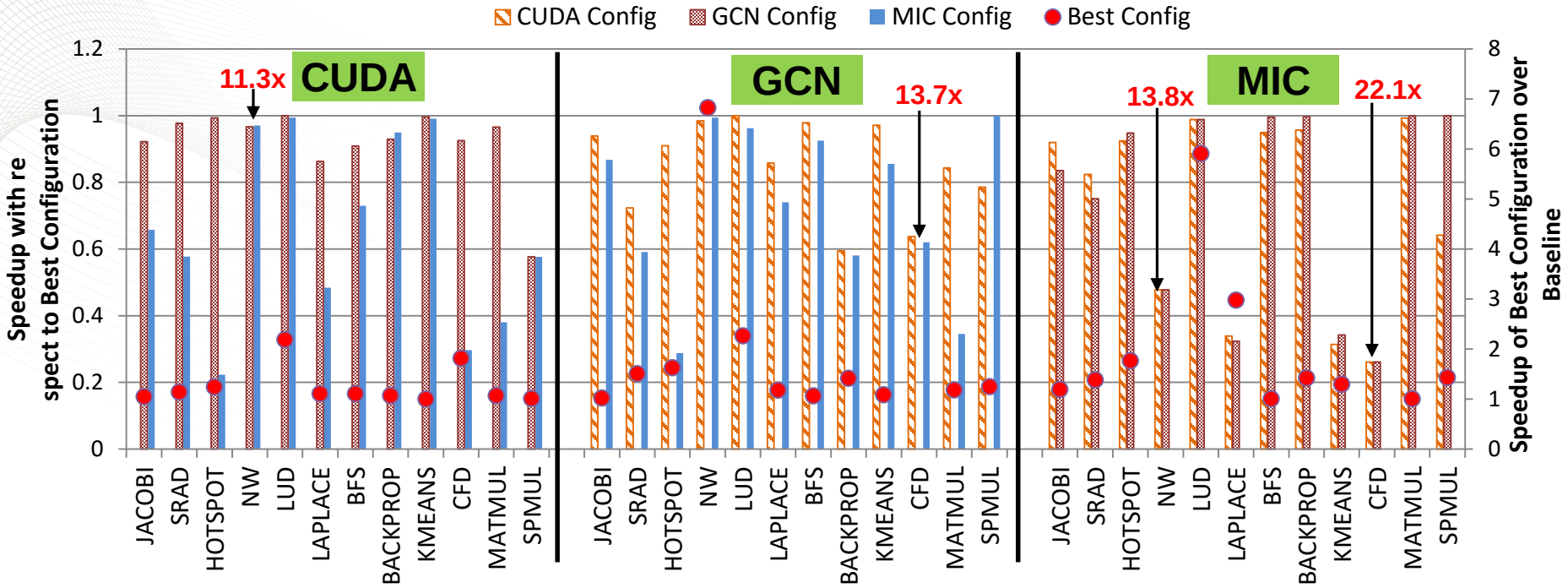
- Solution
  - Proposed a high-level, architecture-independent intermediate language (HeteroIR) to map high-level programming models (e.g., OpenACC) to diverse heterogeneous devices while maintaining portability.
  - Using HeteroIR, port and measure the performance portability of various OpenACC applications 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.



Amit Sabne, Putt Sakdhnagool, Seyong Lee, and Jeffrey S. Vetter. Understanding Portability of a High-level Programming Model on Contemporary Heterogeneous Architectures, IEEE Micro Volume 35, Issue 4 (DOI: 10.1109/MM.2015.73), 2015.

Overall Performance Portability



- Better perf. portability among GPUs
- Lesser across GPUs and MIC
- Main reasons
  - Parallelism arrangement
  - Compiler optimizations : e.g. device-specific memories, unrolling etc.

Performance Portability Matrix

Best Program version of	Executed on: CUDA	Executed on: GCN	Executed on: MIC
CUDA	100	84	65
GCN	91	100	67
MIC	58	68	100

Amit Sabne, Putt Sakdhnagool, Seyong Lee, and Jeffrey S. Vetter. Understanding Portability of a High-level Programming Model on Contemporary Heterogeneous Architectures, IEEE Micro Volume 35, Issue 4 (DOI: 10.1109/MM.2015.73), 2015.

Intelligent selection of optimizations based on target architecture

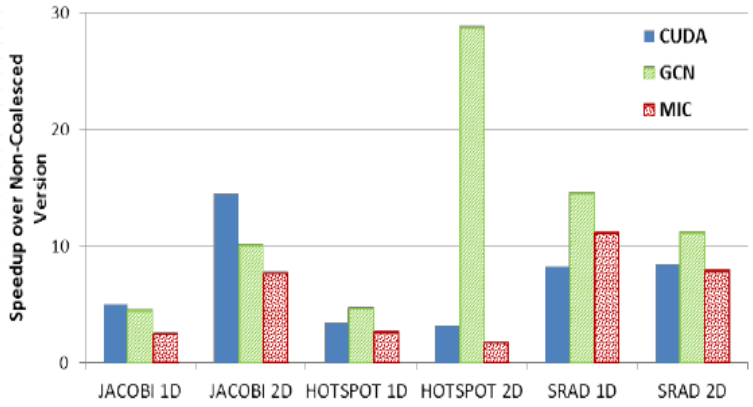


Figure 5: Memory Coalescing Benefits on Different Architectures : MIC is impacted the least by the non-coalesced accesses

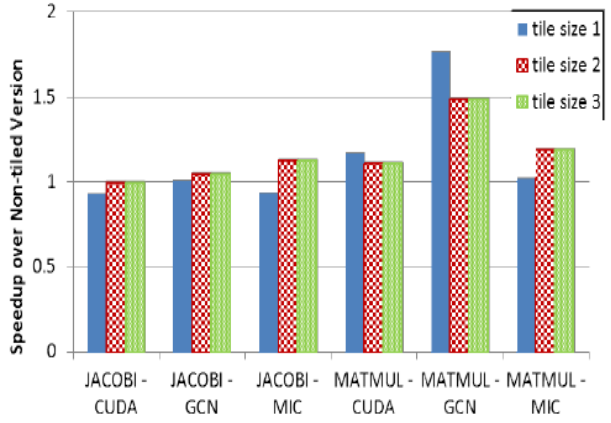


Figure 7: Impact of Tiling Transformation : *MATMUL* shows higher benefits than *JACOBI* owing to more contiguous accesses

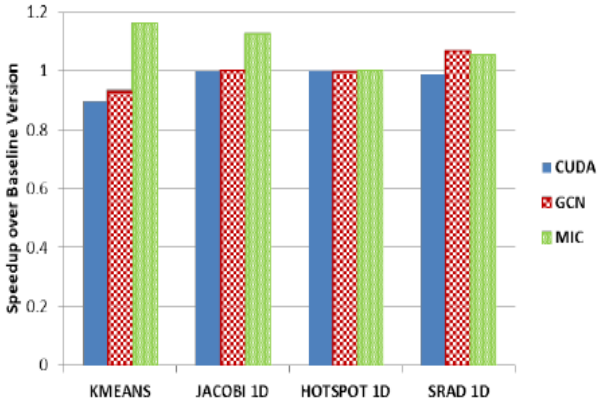


Figure 9: Effects of Loop Unrolling - MIC shows benefits on unrolling

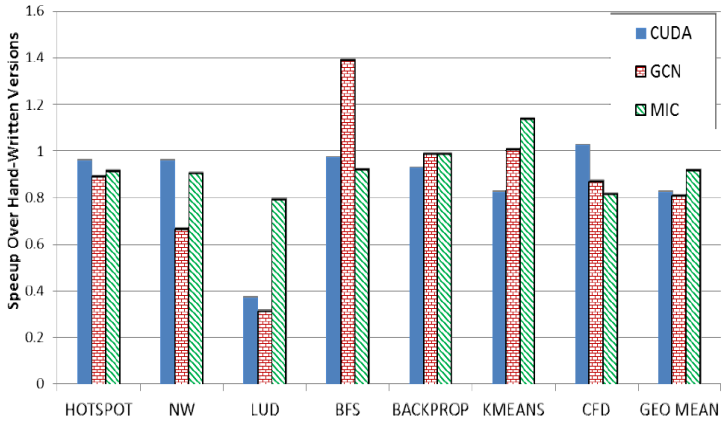
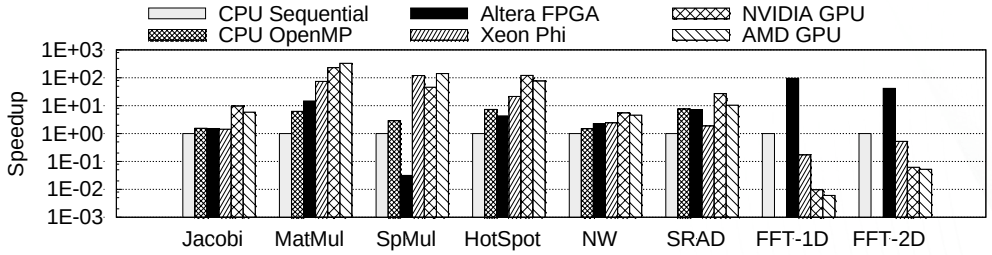


Fig. 11: Comparison of hand-written CUDA/OpenCL programs against auto-tuned OpenARC code versions : Tuned OpenACC programs perform reasonably well against hand-written codes

OpenACC to FPGA: A Framework for Directive-Based High-Performance Reconfigurable Computing

- Problem
  - Reconfigurable computers, such as FPGAs, offer more performance and energy efficiency for specific workloads than other heterogeneous systems, but their programming complexities and low portability have limited their deployment in large scale HPC systems.
- Solution
  - Proposed an OpenACC-to-FPGA translation framework, which performs source-to-source translation of the input OpenACC program into an output OpenCL code, which is further compiled to an FPGA program by the underlying backend Altera OpenCL compiler.
- Recent Results
  - Proposed several FPGA-specific OpenACC compiler optimizations and pragma extensions to achieve higher throughput.
  - Evaluated the framework using eight OpenACC benchmarks, and measured performance variations on diverse architectures (Altera FPGA, NVIDIA/AMD GPUs, and Intel Xeon Phi).



- Impact
  - Proposed translation framework is the first work to use a standard and portable, directive-based, high-level programming system for FPGAs.
  - Preliminary evaluation of eight OpenACC benchmarks on an FPGA and comparison study on other accelerators identified that the unique capabilities of an FPGA offer new performance tuning opportunities different from other accelerators.

Seyong Lee, Jungwon Kim, and Jeffrey S. Vetter. OpenACC to FPGA: A Framework for Directive-based High-Performance Reconfigurable Computing, IPDPS, 2016. http://ft.ornl.gov/research/openarc

Optimizations for Reconfigurable Computing

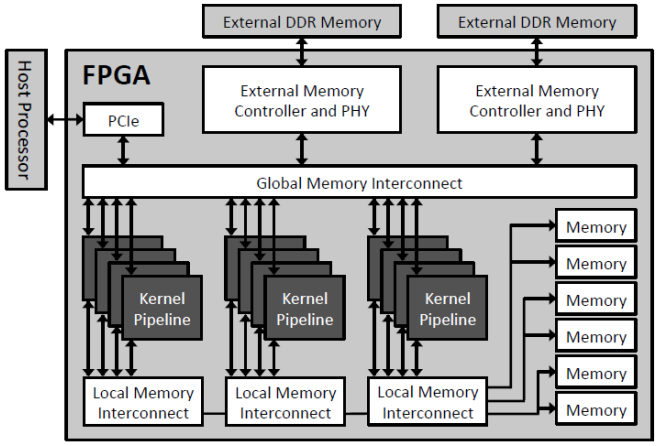
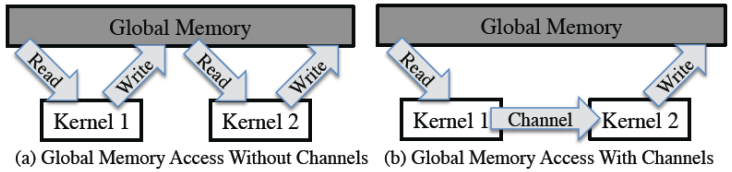


Figure 2: FPGA OpenCL Architecture



(a) Global Memory Access Without Channels (b) Global Memory Access With Channels

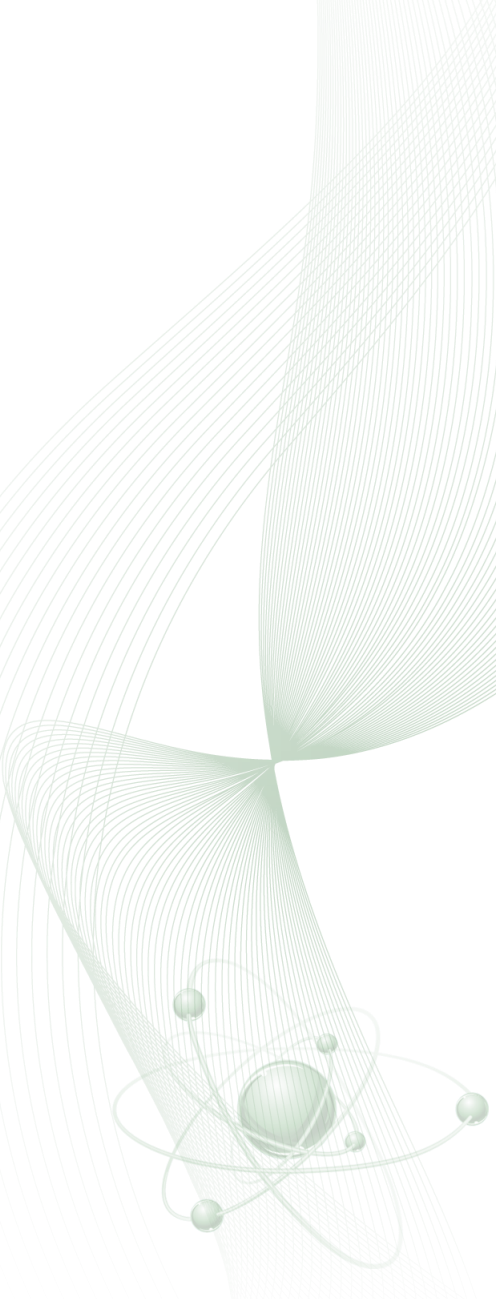
Figure 3: Difference in Global Memory Access Pattern as a Result of Channels Implementation

Listing 4: Altera OpenCL (AOCL) Channel Example

```
1  #pragma acc data copyout(a[0:N]) create(b[0:N]) \\
2  copyin(c[0:N])
3    {
4  #pragma acc kernels loop gang worker present(b, c)
5      for (i=0; i<N; i++) b[i] = c[i]*c[i];
6  #pragma acc kernels loop gang worker present(a, b)
7      for (i=0; i<N; i++) a[i] = b[i];
8    }
9          (a) Input OpenACC code
10
11 #pragma acc data copyout(a[0:N]) pipe(b[0:N]) \\
12 copyin(c[0:N])
13   {
14 #pragma acc kernels loop gang worker pipeout(b) present(c)
15     for (i=0; i<N; i++) b[i] = c[i]*c[i];
16 #pragma acc kernels loop gang worker pipein(b) present(a)
17     for (i=0; i<N; i++) a[i] = b[i];
18   }
19         (b) Modified OpenACC code for kernel-pipelining
20
21 #pragma OPENCL EXTENSION cl_altera_channels : enable
22 channel float pipe__b;
23 __kernel void kernel0(__global float * c)
24 {
25   int i = get_global_id(0);
26   write_channel_altera(pipe__b, (c[i]*c[i]));
27 }
28 __kernel void kernel1(__global float * a)
29 {
30   int i = get_global_id(0);
31   a[i] = read_channel_altera(pipe__b);
32 }
33         (c) Output OpenCL code with channels
```


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.

Benchmarks for Understanding Architectures and Programming Models



OAK RIDGE National Laboratory

The Scalable HeterOgeneous Computing (SHOC) Benchmark Suite

PI: Jeffrey S. Vetter, ORNL
Future Technologies Group

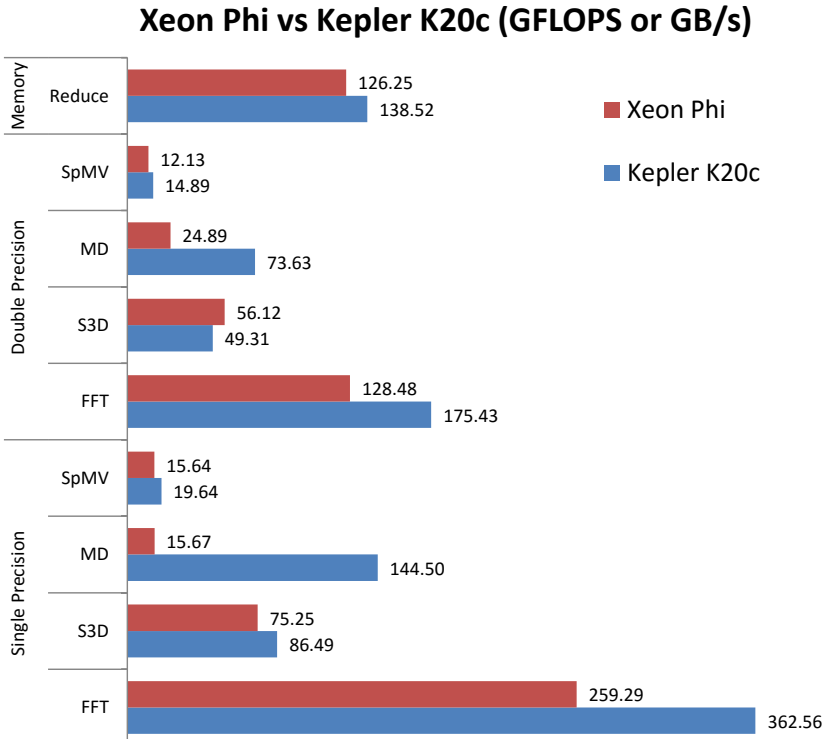
https://github.com/vetter/shoc/wiki

Objectives

- *Design and implement a set of performance and stability tests for HPC systems with heterogeneous architectures*
- *Implement each test in OpenCL, CUDA, OpenACC, OpenMP, and with MPI, to:*
  - *Evaluate the differences in these emerging programming models*
  - *Across diverse set of architectures (NVIDIA, AMD, ARM, Xeon Phi)*
- *Increase understanding of how important applications will map to emerging architectures*
- *Open Source for easy use, porting, contributions*

Accomplishments

- *Consistent open source software releases*
- *Overview published at 3rd Workshop General-Purpose Computation on Graphics Processing Units (GPGPU '10)*
- *Updated 2.0 version adds OpenACC and Intel Xeon Phi support via OpenMP and offlload directives*
- *New architecture and programming model analysis to be published at 6th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS15) at SC15*



Impact and Champions

- *Over 10000 downloads internationally since 2010*
- *Approximately 275 citations*
- *Used by vendors, researchers, and for procurements*
- *Provide a standardized test suite for architecture evaluations, procurements, and acceptance tests*
- *Researchers and engineers from several computing device vendors are engaged and providing contributions*

M.G. Lopez, J. Young, J.S. Meredith, P.C. Roth, M.Horton and J.S. Vetter, "Examining Recent Many-core Architectures and Programming Models Using SHOC", in 6th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS15), Austin, 2015.

A. Danalis, G. Marin, C. McCurdy, J. Meredith, P.C. Roth, K. Spafford, V. Tipparaju, and J.S. Vetter, "The Scalable HeterOgeneous Computing (SHOC) Benchmark Suite," in Third Workshop on General-Purpose Computation on Graphics Processors (GPGPU 2010)`. Pittsburgh, 2010.

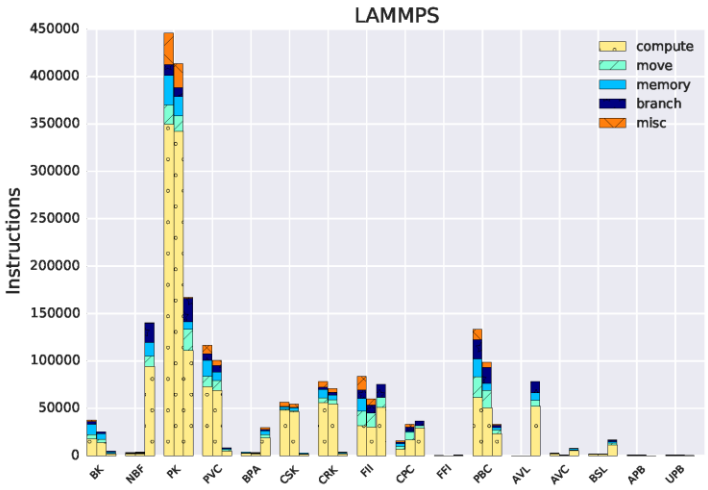
Performance Tools

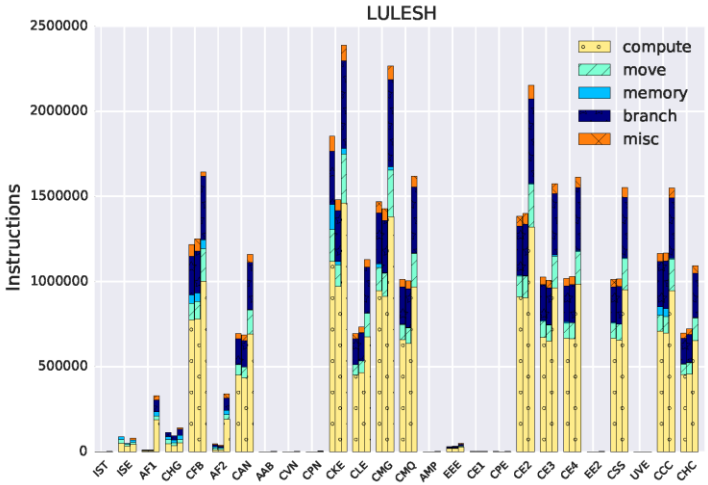
Performance Tools: TAU

- New GPU sampling capabilities in TAU were applied to LAMMPS and LULESH to characterize their behavior during execution across multple GPU architectures.
- New control flow graph (CFG) analysis of GPU kernels was integrated in TAU and allowed more detailed characterization and autotuning of Gaussian across GPU architectures.
- Optimization of the OpenMC framework on Xeon Phi clusters (TACC Stampede, LBNL Babbage) with TAU resulted in the highest known single-node calculation rate for a Monte Carlo neutron transport code using a standard benchmark (17,000 particles/second, 95% distributed efficiency with 512 concurrent MIC devices).

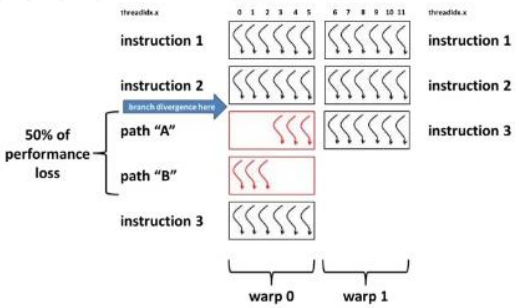
TAU Sampling Enables Insights in GPU Performance

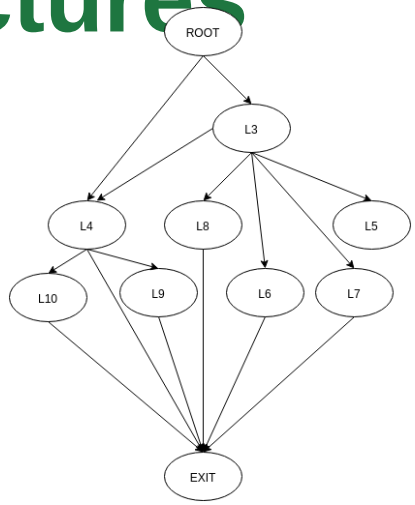
- Problem
  - Hard to correlate performance spikes with source code
  - Event queue method: inject event at beginning and end of execution (no idea what happens in between)
- Solution
  - Sample GPU program counter during runtime
  - Map PC samples with disassembled instructions
  - Calculate metrics intensity at kernel level
    - Paper: Identifying Optimization Opportunities within Kernel Execution in GPU Codes (HeteroPar 2015)
- Recent results
  - Three GPUs: M2090 (Fermi), K80 (Tesla), M6000 (Maxwell)
  - LAMMPS: PK kernel mostly compute operations
  - LULESH: CKE, CMG, CE2 more compute-intensive, also branch- and move-intensive
- Impact
  - Understand kernel behavior in real time
  - Identifying where in code to spend tuning efforts

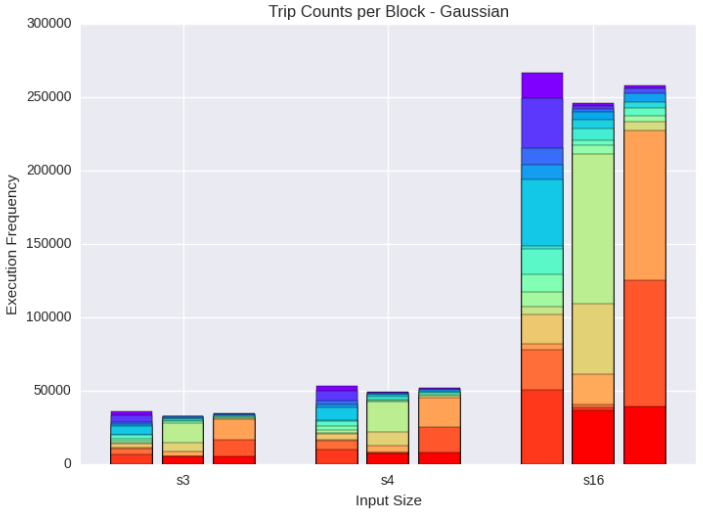




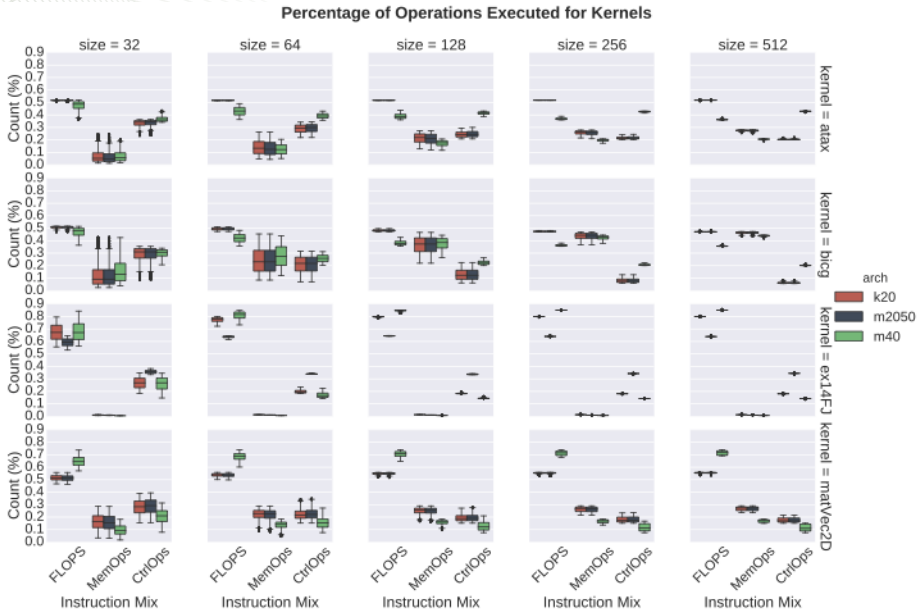
Modeling Control Flow Behavior in GPU Architectures







- Problem
  - GPU architectures execute SIMD in lock step (mask threads that do not satisfy branch conditions)
  - Lanes allow branching threads to execute and non-branching threads to wait and synchronize (performance drawbacks)
- Solution
  - Control flow graphs (CFG) used to represent divergent branches
  - Derive execution frequencies, determine how application of input size N will perform without compiling or running application
- Recent results
  - Three GPUs: M2090 (Fermi), K80 (Tesla), M6000 (Maxwell)
    - Each GPU creates its own CFG
    - Higher trip counts seen for Gaussian on Fermi
  - Autotuned four kernel computations using Orio on three GPU architectures
- Impact
  - Predict optimal performance parameters for applications on a given architecture (e.g. thread counts, block sizes), based on PC Sampling data and basic block counts



Percentage of overall instruction operations executed for ATAX, BiCG, ex14FJ and MatVec2D kernels for five input sizes, comparing various architecture generations.

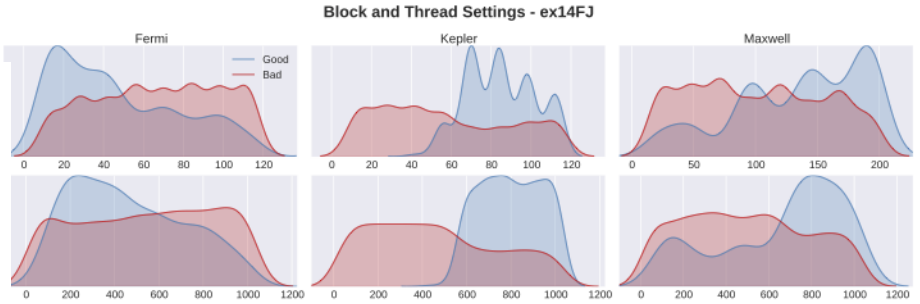
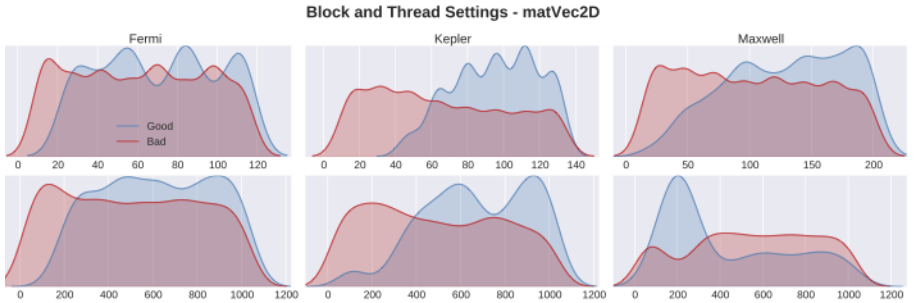
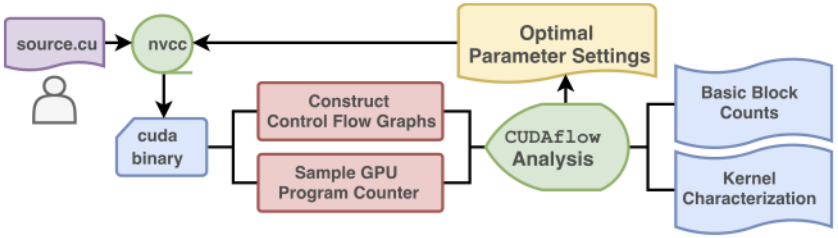


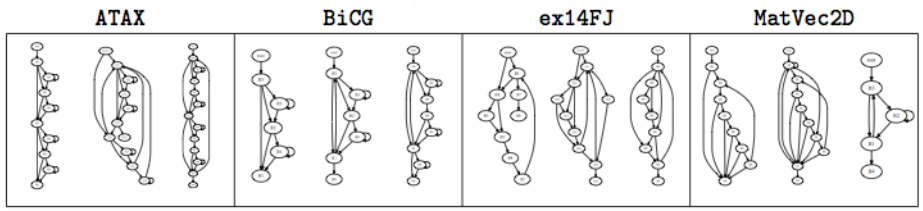
Fig. 6. Block and thread sizes for ex14FJ kernel, comparing various architectures.



Block and thread sizes for matVec2D kernel, comparing various architectures.



Overview of `CUDAflow` methodology in discovering application settings.



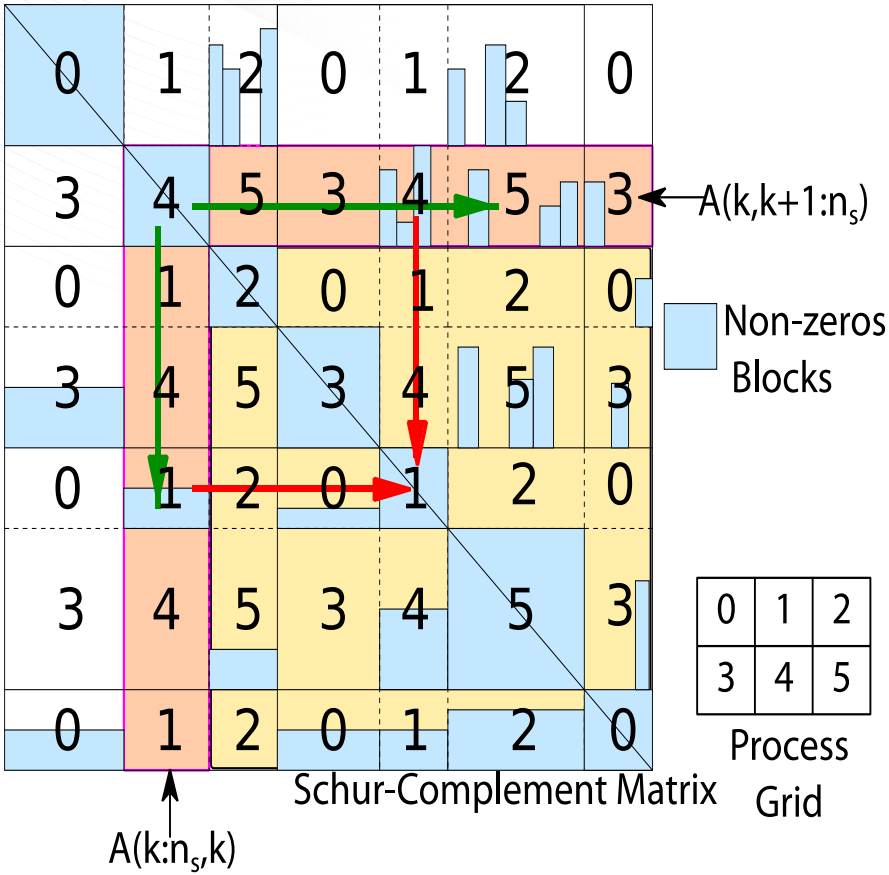
Control flow graphs generated for each CUDA kernel, comparing architecture families (Fermi, Kepler, Maxwell).

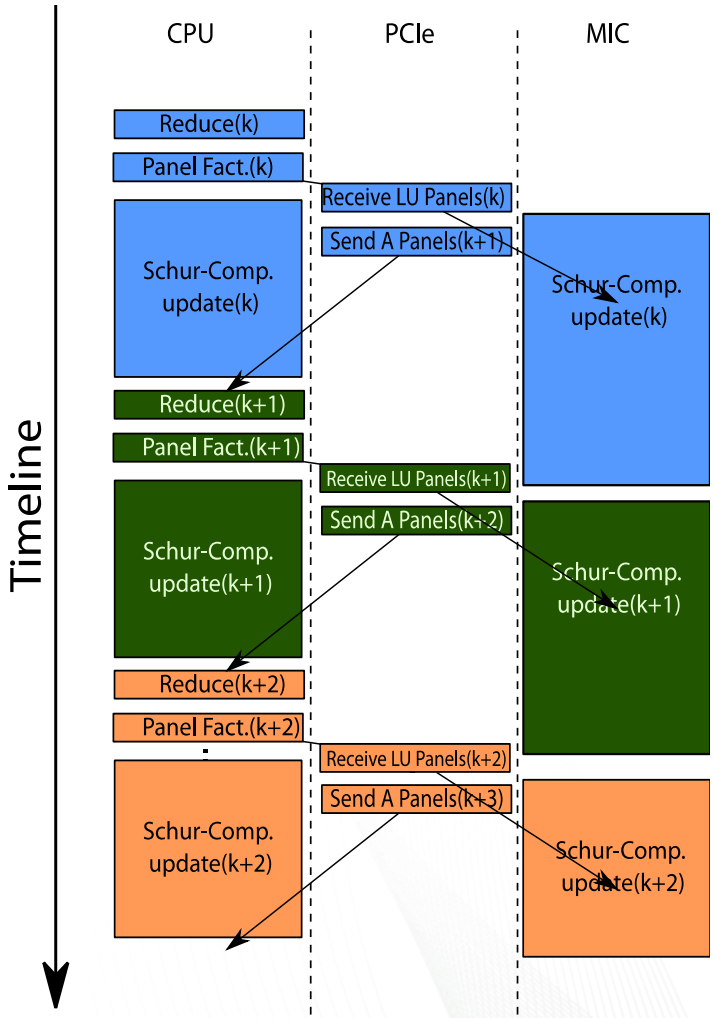
Model Driven Autotuning Libraries: SuperLU Dist

Model Driven Autotuning Libraries

- Focus on enabling technologies for libraries, especially for sparse and irregular computations on heterogeneous platforms
  - Computations we have accelerated under Vancouver 1 & 2:
    Fast transforms, N-body tree codes, and sparse matrix computations
  - Technologies we have developed:
    New data structures, model-driven autotuning
  - Platforms: *All* work targets distributed memory heterogeneous systems
- Highlight (from Vancouver 2): The *first* hybrid, distributed memory CPU+GPU sparse direct solver (sparse Gaussian elimination)
  - Collaboration with Sherry Li at LBNL (SciDAC-funded)
  - GPU work is integrated into the open-source library, SuperLU_DIST
  - A unified framework targeting GPU *and* Xeon Phi is in progress

SuperLU_DIST (sparse Ax=b solver)

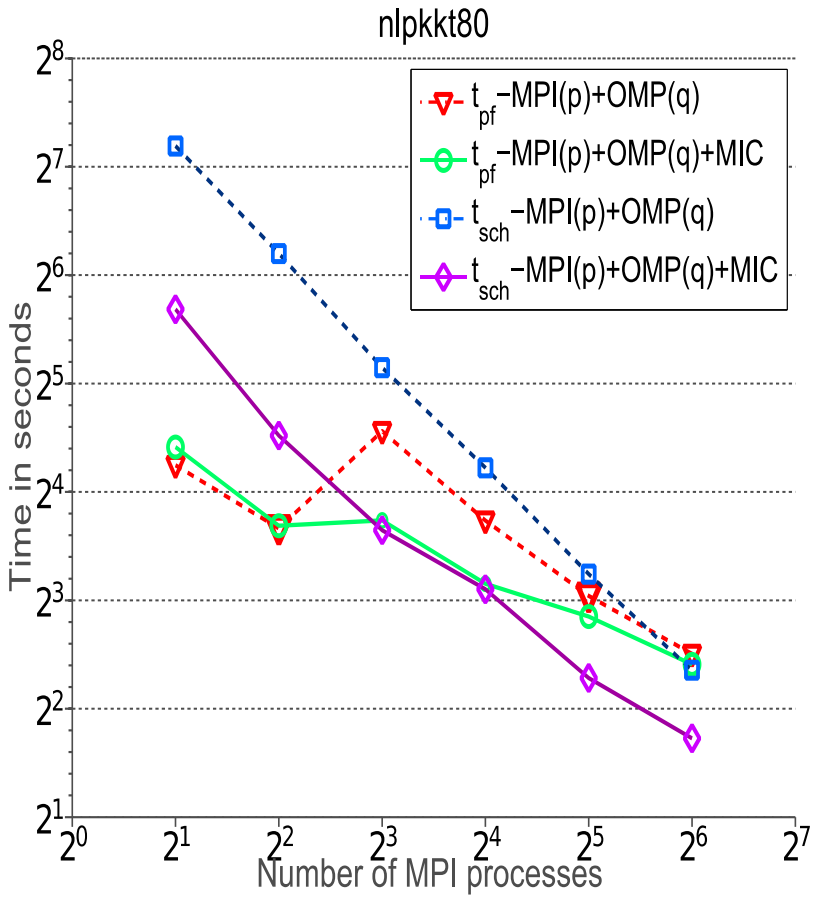


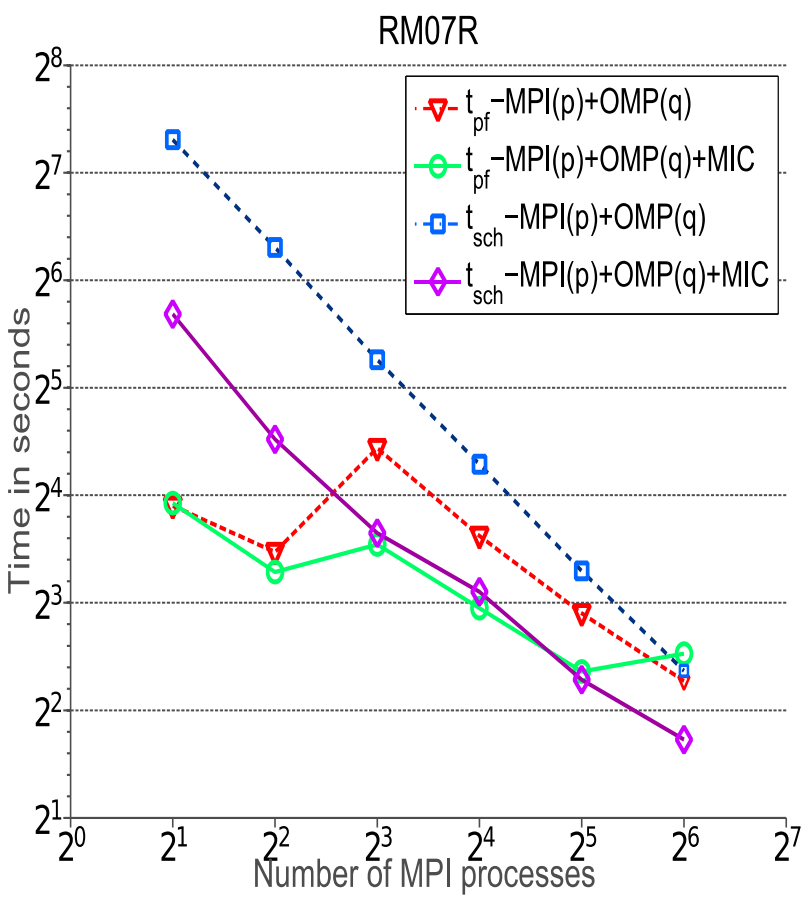


We analyzed complex dependences and data structures in prior version of SuperLU

We designed a model-driven autotuning framework for scheduling and adapting SuperLU to use co-processors (GPU + Xeon Phi)

SuperLU_DIST (sparse Ax=b solver)



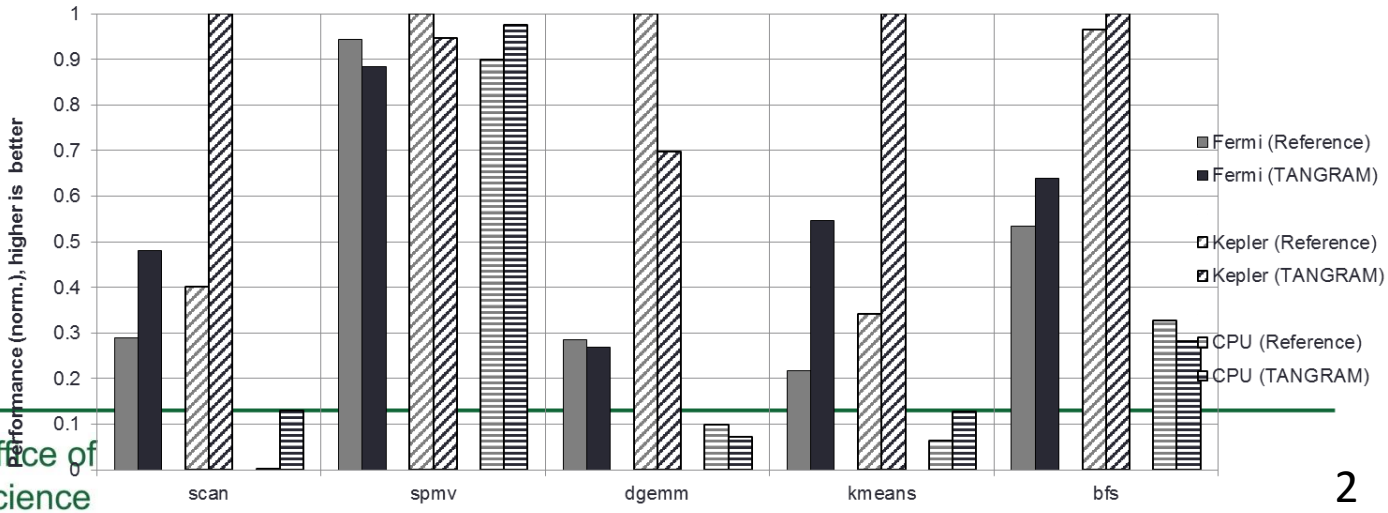


Our results indicate good strong scaling (above), weak scaling (not shown) and absolute performance (fraction of peak) when running on accelerated systems.

Code Synthesis Tools

Tangram – Code Synthesis for Performance Portability

- Problem
  - High-performance, energy-efficient code is too costly to develop and maintain
- Solution
  - Idea: large, complex high-performance code are often built from small building block and a small set of decomposition/composition schemes
  - Complexity comes from multi-level composition/decomposition and optimizations
  - Users provide building blocks and rules (easy, reusable) and Tangram automate the multi-level composition/decomposition/optimization (complex, specific to each hardware type).
- Recent results
  - Synthesized code from simple portable sources matches or beats custom, hand-tuned C-level code across GPUs and CPUs
- Impact
  - Drastically reduced development and porting cost for high-performance, energy-efficient code at the exascale



Vancouver II: RSbench results

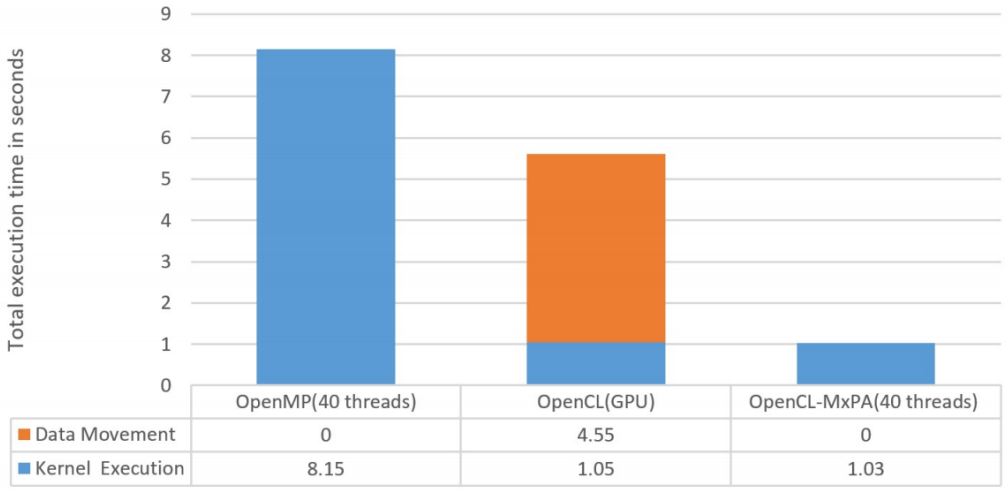


Figure 1: OpenCL/MxPA performance versus original OpenMP code for RSbench. Hardware:[CPU:4 Intel Xeon E5-2680V2, GPU:1 Nvidia K40m]. OpenCL implementation was only 45% faster than the original OpenMP implementation; this was due to the overhead of transferring data to and from the GPU. The OpenCL re-targeted code by MXPA gets about 8x speedup over the original. The OpenCL explicitly exposes more of the available parallelism than the OpenMP code, MxPA takes advantage of this when scheduling work to the CPU cores.

- **Available OpenCL GPU code retargeted for CPU execution through MxPA**
- **MxPA successfully provides portability from a single source language**
- **Outperforms the original OpenMP version when targeting both the GPU and CPU.**

Wrapup

Tech Transfer and Some Futures

- **Tech transfer approaches**
  - Open source
  - Deployment
  - Working directly with users
  - Vendor deployment
  - Impact specifications like OpenACC
- **Futures (same areas)**
  - Memory hierarchies
    - Unified memory
    - Paging
    - Software managed cache
  - New heterogeneous architectures and levels of integration
  - Performance prediction of architectural alternatives at runtime
  - Preemption
  - New performance analysis features

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

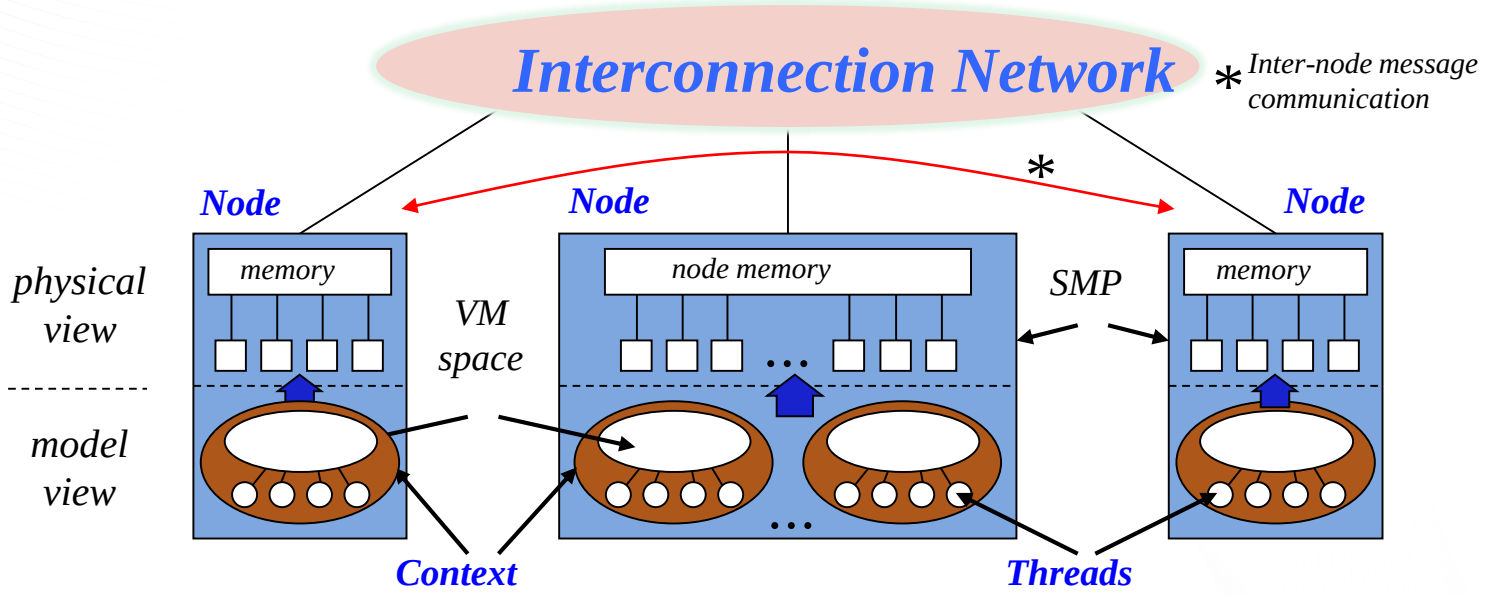
Bonus

Software and tech transfer strategies

- Many ways our project distributes software and technology
  - Publicly available open source (OpenARC, SHOC, etc)
  - Deployed directly on DOE systems (Tau)
  - Deployed through vendor
    - NVIDIA CUSPARSE Tridiagonal Solver by UIUC
    - UIUC MxPA distributed by MultiCoreware
  - Influence broad community software (LLVM)
- Additional strategy
  - Develop research prototype reference implementation
  - Pursue interesting, relevant challenges
  - Use results and experiences to influence standards
    - OpenACC, OpenMP, SPEC, etc
  - Use pull of procurements to get new standards implemented

TAU Performance System®

- Performance problem solving framework for HPC
  - Integrated, scalable, flexible, portable
  - Target all parallel programming / execution paradigms



- Integrated performance toolkit (open source)
  - Multi-level performance instrumentation
  - Flexible and configurable performance measurement
  - Widely-ported performance profiling / tracing system
  - Performance data management and data mining

TAU Deployment Status

- Initial port of TAU to IBM Power 8 Linux with GPUs
- OpenACC support with PGI 15.x compilers
- Support for Intel ® Xeon Phi™ systems:
  - BFD (binutils for address translation),
  - libunwind for callstack unwinding
  - OpenMP Tools interface based on LLVM runtime
  - Support for tracking memory footprint
  - Support for hardware performance counters to measure vectorization
  - Support for tracking NUMA effects (remote to total DRAM accesses)
- Support for ARM64 systems added, port to ARM64 with GPUs is underway
- TAU v2.25 released with BSD style license in November 2015
- http://tau.uoregon.edu