CU2CL: An Automated CUDA-to-OpenCL Source-to-Source Translator

Wu FENG

Dept. of Computer Science and Dept. of Electrical & Computer Engineering
NSF Center for High-Performance Reconfigurable Computing (CHREC)
Center for High-End Computing Systems (CHECS)

VirginiaTech
Invent the Future

wfeng@vt.edu, 540.231.1192

SyNeRG
synergy.cs.vt.edu

Paying For Performance

- "The free lunch is over..." †
  - Programmers can no longer expect substantial increases in single-threaded performance.
  - The burden falls on developers to exploit parallel hardware for performance gains.

† H. Sutter, "The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software," *Dr. Dobb's Journal*, 30(3), March 2005. (Updated August 2009.)

- How do we lower the cost of concurrency?

wfeng@vt.edu, 540.231.1192

The Berkeley View †

- Traditional Approach
  - Applications that target existing hardware and programming models
- Berkeley Approach
  - Hardware design that keeps future applications in mind
  - Basis for future applications? 13 computational dwarfs

A computational dwarf is a pattern of communication & computation that is common across a set of applications.

Sparse Linear Algebra

Dense Linear Algebra

Spectral Methods

Structured Grids

N-Body Methods

Unstructured Grids

Monte Carlo → MapReduce

and

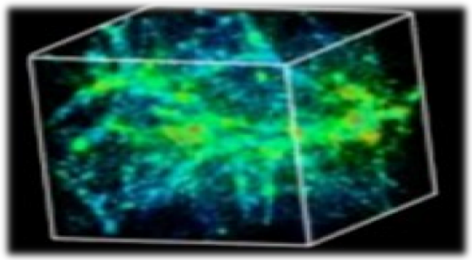
Combinational Logic
Graph Traversal
Dynamic Programming
Backtrack & Branch+Bound
Graphical Models
Finite State Machine

† Asanovic, K., et al. *The Landscape of Parallel Computing Research: A View from Berkeley*. Tech. Rep. UCB/EECS-2006-183, University of California, Berkeley, Dec. 2006.

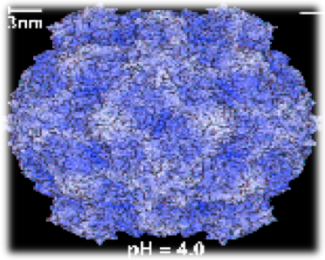
wfeng@vt.edu, 540.231.1192

Example of a Computational Dwarf: N-Body

- N-Body problems are studied in
  - Cosmology, particle physics, biology, and engineering
- All have similar structures
- An N-Body benchmark can provide meaningful insight to people in all these fields
- Optimizations may be generally applicable as well



RoadRunner Universe: Astrophysics



GEM: Molecular Modeling

wfeng@vt.edu, 540.231.1192

OpenDwarfs (a.k.a. OpenCL and the 13 Dwarfs)

https://github.com/opendwarfs/OpenDwarfs

- Provide common algorithmic methods, i.e., dwarfs, in a language that is "write once, run anywhere" (CPU, GPU, or even FPGA), i.e., OpenCL

- Part of a larger umbrella project (2008-2012) funded by the NSF Center for High-Performance Reconfigurable Computing

CHREC
NSF Center for High-Performance Reconfigurable Computing

AMD
HARRIS

Los Alamos NATIONAL LABORATORY
XILINX

wfeng@vt.edu, 540.231.1192

Status of OpenCL & the 13 Dwarfs

2009 – 2011

Dwarf	Done
Dense linear algebra	LU Decomposition
Sparse linear algebra	Matrix Multiplication
Spectral methods	FFT
N-Body methods	GEM
Structured grids	SRAD
Unstructured grids	CFD solver
MapReduce	
Combinational logic	CRC
Graph traversal	Breadth-First Search (BFS)
Dynamic programming	Needleman-Wunsch
Backtrack and branch-and-bound	
Graphical models	Hidden Markov Model
Finite state machines	Temporal Data Mining

88x → 371x

wfeng@vt.edu, 540.231.1192

Our Solutions

- Functional Portability (2 years → real time)
  - ***CU2CL*** (pronounced as "cuticle")
    An Automated CUDA-to-OpenCL Source-to-Source Translator

    OpenMP → OpenCL

    OpenCL → (AutoESL+ GCC) → FPGA
- Performance Portability (88x → 371x)
  - M. Daga, T. Scogland, and W. Feng, "Architecture-Aware Mapping and Optimizations on a 1600-Core GPU," *17th IEEE Int'l Conf. on Parallel and Distributed Systems*, December 2011.

wfeng@vt.edu, 540.231.1192

Our Solutions

- Functional Portability (2 years → real time)
  - ***CU2CL*** (pronounced as "cuticle")
    An Automated CUDA-to-OpenCL Source-to-Source Translator

    OpenMP → OpenCL

    OpenCL → (AutoESL+ GCC) → FPGA
- Performance Portability (88x → 371x)
  - M. Daga, T. Scogland, and W. Feng, "Architecture-Aware Mapping and Optimizations on a 1600-Core GPU," *17th IEEE Int'l Conf. on Parallel and Distributed Systems*, December 2011.

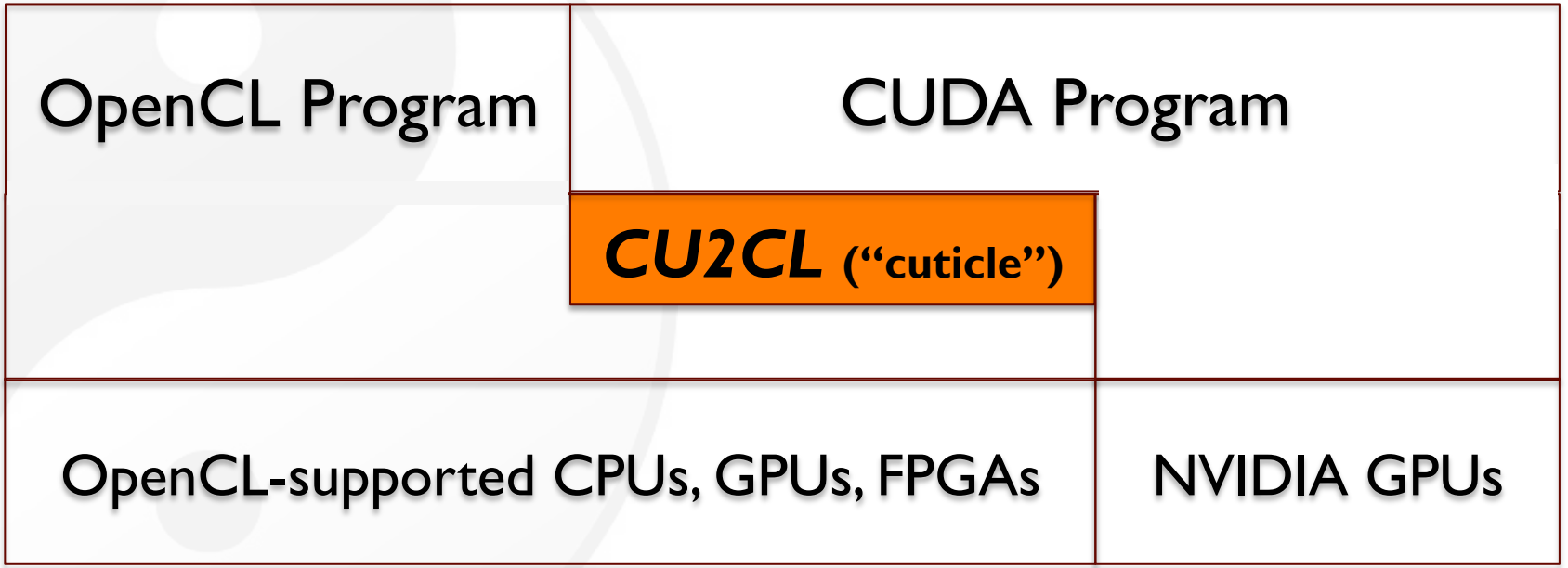
wfeng@vt.edu, 540.231.1192

Forecast

- Motivation & Background
- CU2CL: A CUDA-to-OpenCL Source-to-Source Translator
  - Goals & Background
  - Architecture
  - Evaluation
    - Coverage, Translation Time, and Performance
- Future Work
- Summary

wfeng@vt.edu, 540.231.1192

Overarching Goal: "Write Once, Run Anywhere"



wfeng@vt.edu, 540.231.1192

Goals of CU2CL ("cuticle")

- Automatically create a treasure trove of

  … *maintainable OpenCL code for future development*

wfeng@vt.edu, 540.231.1192

Examples of Available CUDA Source Code

Source: http://gpgpu.org/

- odeint: ODE solver
- OpenCurrent: PDE solver
- R+GPU: accelerate R
- Alenka: "SQL for CUDA"
- GPIUTMD: multi-particle dyn
- rCUDA: remote inv
- HOOMD-blue: pa
- Exact String Match
- GMAC: asymmetric
- TRNG: random num
- OpenNL: numeric libr
- VMD: visual molecular amics
- CUDA memtest

A treasure trove of free GPU applications for OpenCL

- GPU-accelera d Ising model
- Image s on via Livewire
- erated CFD
- ng
- de
- ve propagation
- cosmological lattice
- C: volumetric iso-surface extraction
- OpenMM: molecular dynamics
- MUMmerGPU: DNA alignment
- SpMV4GPU: sparse-matrix multiplication toolkit

and many more…

wfeng@vt.edu, 540.231.1192

Goals of CU2CL ("cuticle")

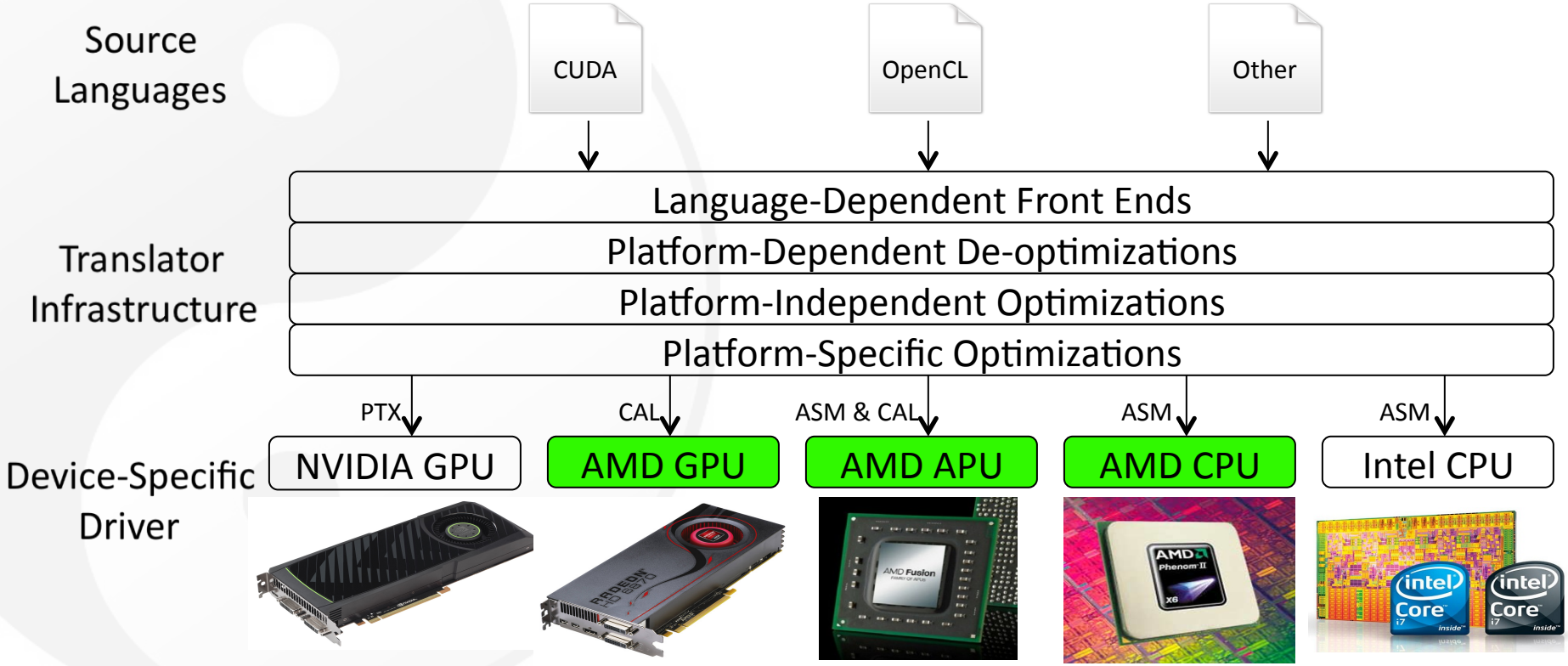
- Automatically create a treasure trove of
  … ***maintainable OpenCL code for future development***

- Promote the increasing adoption of OpenCL
  … from AMD, ARM, & Intel to Altera, Xilinx, & Qualcomm

Already receiving ***nearly daily requests*** for the CU2CL tool
… from end users wanting to translate their codes

wfeng@vt.edu, 540.231.1192

Ecosystem for Source-to-Source Translation



wfeng@vt.edu, 540.231.1192

Forecast

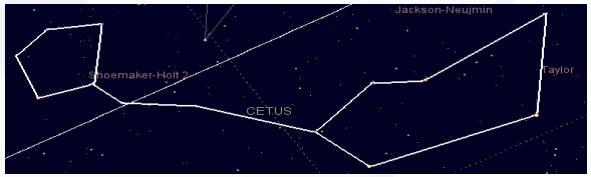
- Motivation & Background
- CU2CL: A CUDA-to-OpenCL Source-to-Source Translator
  - Goals & Background
  - Architecture
  - Evaluation
    - Coverage, Translation Time, and Performance
- Future Work
- Summary

wfeng@vt.edu, 540.231.1192

Translator Base to Build Upon

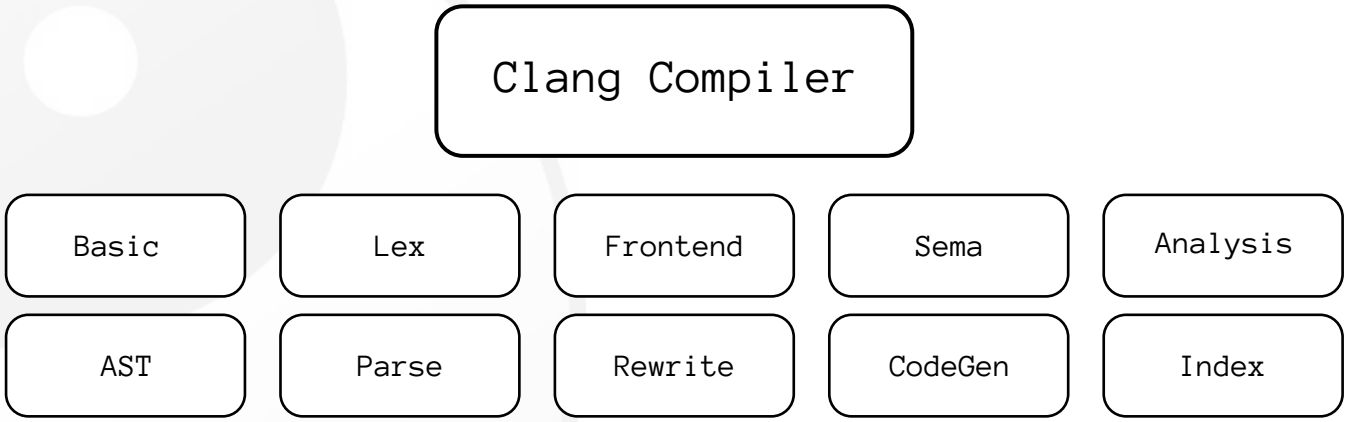
- Production-quality compiler
- Ease of extensibility

Cetus



wfeng@vt.edu, 540.231.1192

The Clang Compiler Framework

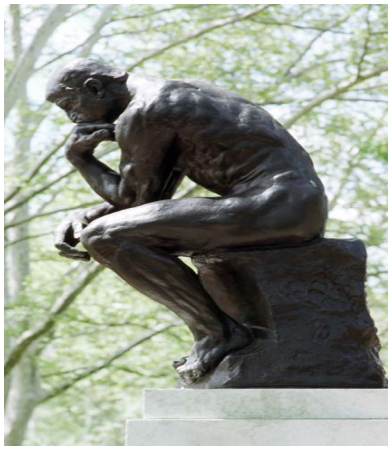


- Useful libraries for C/C++ source-level tools
- Powerful AST representation
- Clang compiler built on top

wfeng@vt.edu, 540.231.1192

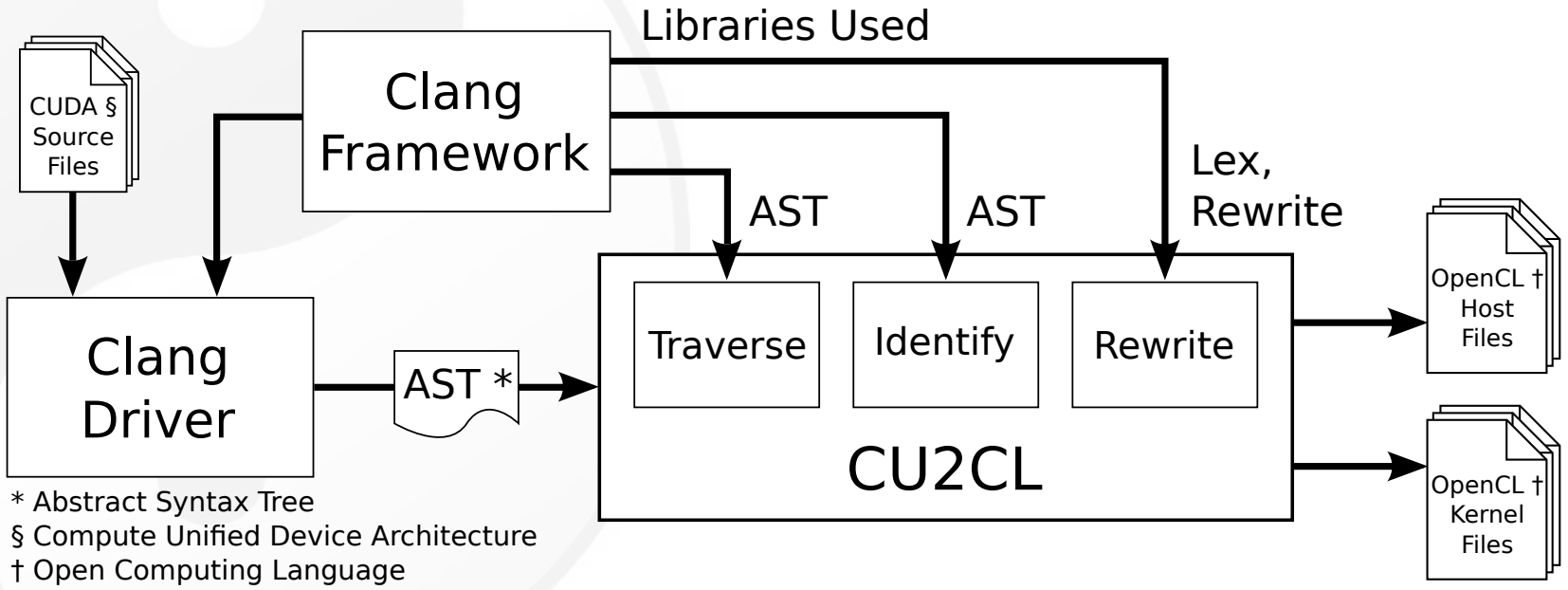
AST-Driven, String-Based Rewriting

- Characteristics
  - Does not modify the AST
  - Instead, edit text in source ranges
- Benefits
  - Useful for transformations with limited scope
  - Preserves formatting and comments



wfeng@vt.edu, 540.231.1192

Architecture of CU2CL



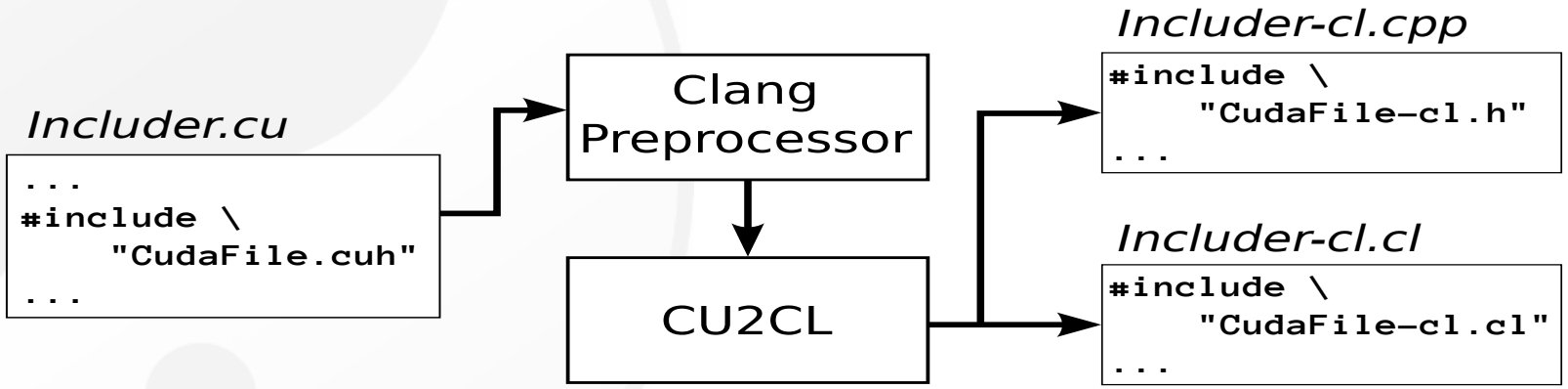
wfeng@vt.edu, 540.231.1192

Translation Procedure of CU2CL

- Traverse the AST
  - Clang's AST library, walking nodes and children
- Identify structures of interest
  - Common patterns arise
- Rewrite original source range as necessary
  - Variable declarations: rewrite type
  - Expressions: recursively rewrite full expression
  - Host code: remove from kernel files
  - Device code: remove from host files
  - #includes: rewrite to point to new files

wfeng@vt.edu, 540.231.1192

Rewriting #includes



wfeng@vt.edu, 540.231.1192

Forecast

- Motivation & Background
- CU2CL: A CUDA-to-OpenCL Source-to-Source Translator
  - Goals & Background
  - Architecture
  - Evaluation
    - Coverage, Translation Time, and Performance
- Future Work
- Summary

wfeng@vt.edu, 540.231.1192

Experimental Set-Up

- CPU
  - 2 x 2.0-GHz Intel Xeon E5405 quad-core
  - 4 GB of Ram
- GPU
  - NVIDIA GTX 280
  - 1 GB of graphics memory
- Applications
  - CUDA SDK
    - asyncAPI, bandwidthTest, BlackScholes, matrixMul, scalarProd, vectorAdd
  - Rodinia
    - Back Propagation, Breadth-First Search, Hotspot, Needleman-Wunsch, SRAD

wfeng@vt.edu, 540.231.1192

Coverage: CUDA SDK and Rodinia

Source	Application	CUDA Lines	Changed	Percentage
CUDA SDK	asyncAPI	136	4	97.06
	bandwidthTest	891	9	98.99
	BlackScholes	347	4	98.85
	matrixMul	351	2	99.43
	scalarProd	171	4	97.66
	vectorAdd	147	0	100.00
Rodinia	Back Propagation	313	5	98.40
	Breadth-First Search	306	8	97.39
	Hotspot	328	7	97.87
	Needleman-Wunsch	418	0	100.00
	SRAD	541	0	100.00

wfeng@vt.edu, 540.231.1192

Coverage: Molecular Modeling Application

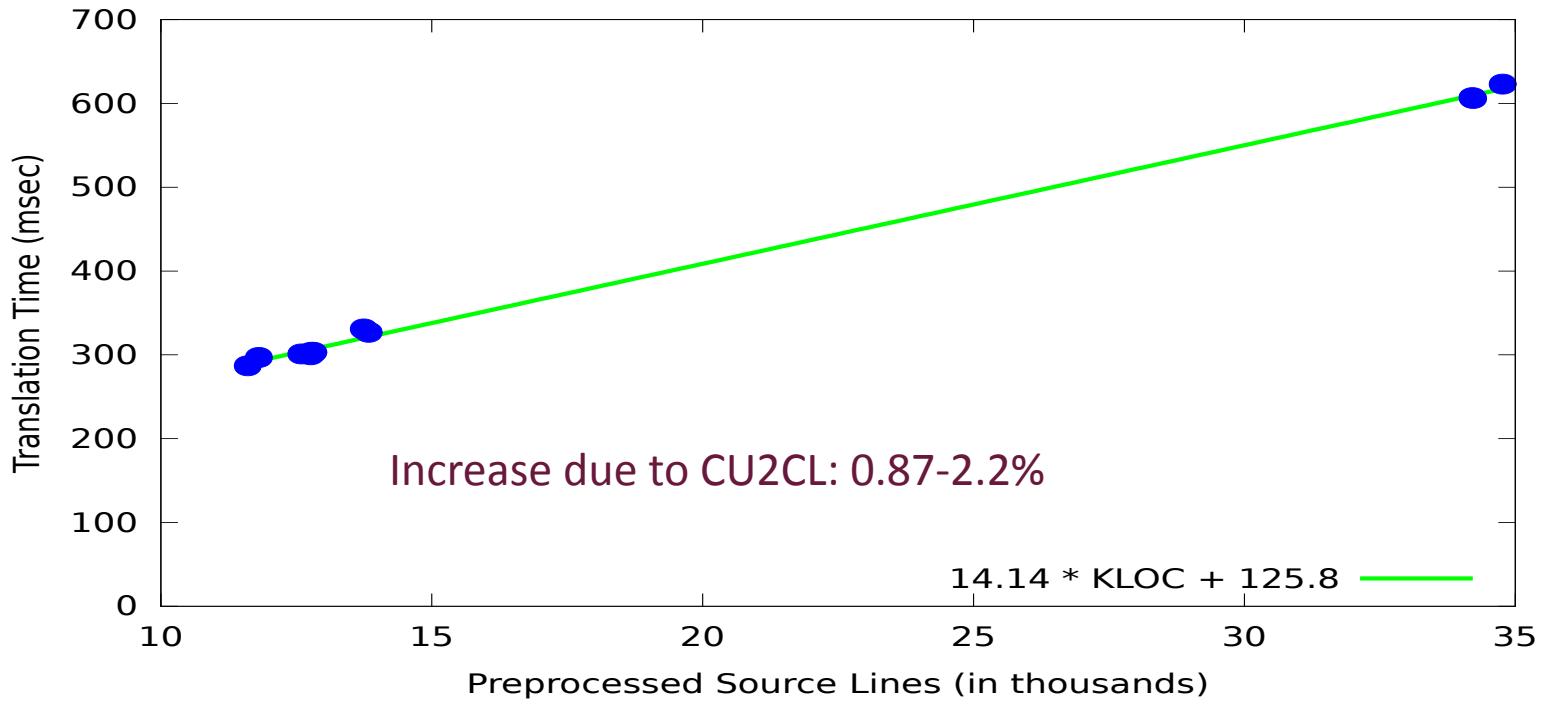
Source	Application	CUDA Lines	Changed	Percentage
Virginia Tech	GEM	2,511	5	99.8

2,511 CUDA lines out of 6,727 total SLOC in GEM application

- *Fundamental Application in Computational Biology*
  - Simulate interactions between atoms & molecules for a period of time by approximations of known physics
- *Example Usage*
  - Understand mechanism behind the function of molecules
    - Catalytic activity, ligand binding, complex formation, charge transport

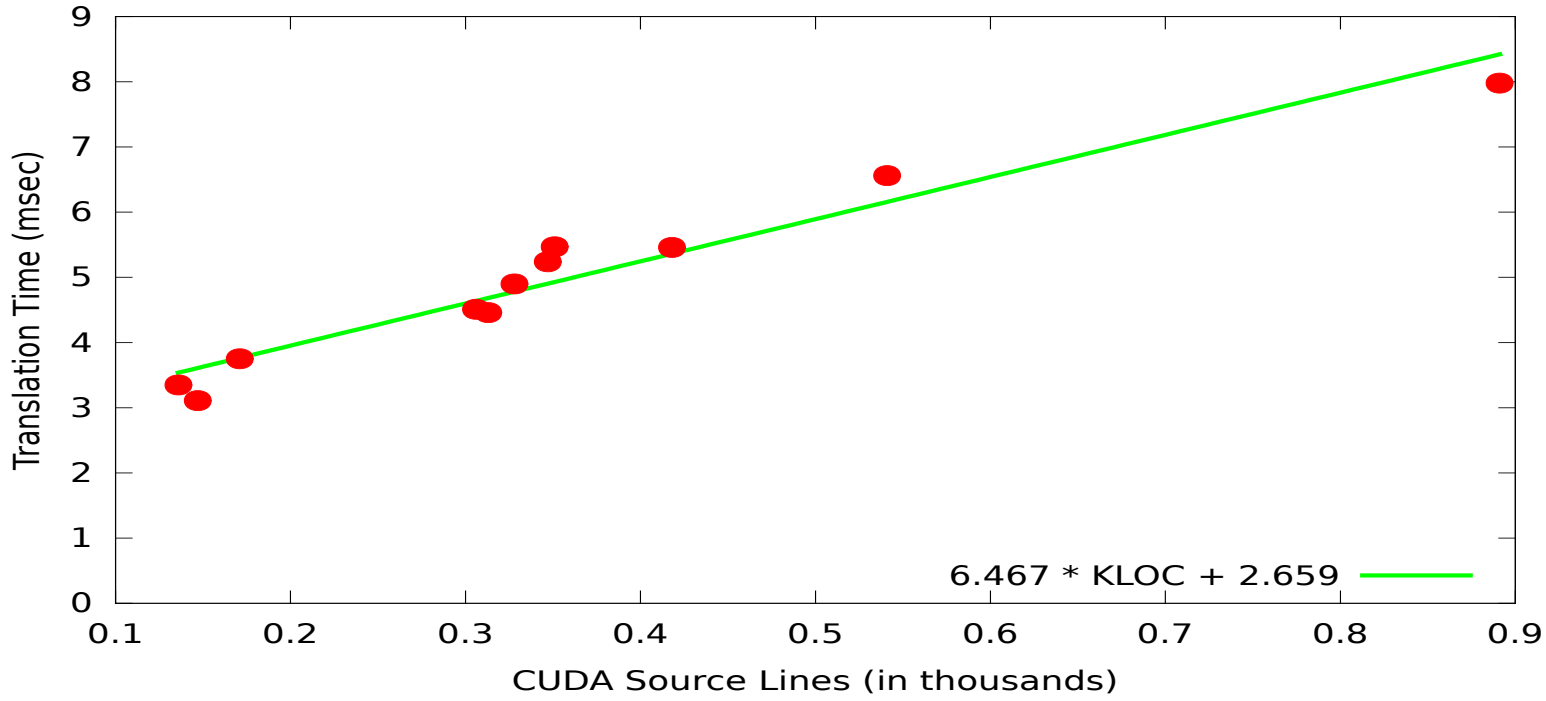
wfeng@vt.edu, 540.231.1192

Model for Total Translation Time



wfeng@vt.edu, 540.231.1192

Model for CU2CL-Only Translation Time



wfeng@vt.edu, 540.231.1192

Status of OpenCL & the 13 Dwarfs

2009 – 2011
vs. CU2CL

Dwarf	Done
Dense linear algebra	LU Decomposition
Sparse linear algebra	Matrix Multiplication
Spectral methods	FFT
N-Body methods	GEM
Structured grids	SRAD
Unstructured grids	CFD solver
MapReduce	
Combinational logic	CRC
Graph traversal	BFS, Bitonic sort
Dynamic programming	Needleman-Wunsch
Backtrack and Branch-and-Bound	
Graphical models	Hidden Markov Model
Finite state machines	Temporal Data Mining

wfeng@vt.edu, 540.231.1192

Translated Application Performance (sec)

Application	CUDA	Automatic OpenCL	Manual OpenCL
vectorAdd	0.0499	0.0516	0.0521
Hotspot	0.0177	0.0565	0.0561
Needleman-Wunsch	6.65	8.77	8.77
SRAD	1.25	1.55	1.54

- Automatically translated OpenCL codes yield similar execution times to manually translated OpenCL codes
- OpenCL performance lags CUDA (at least for OpenCL 1.0)
  - Similar for OpenCL 1.1

wfeng@vt.edu, 540.231.1192

CU2CL with OpenCL and the 13 Dwarfs

Dwarf	Implemented	AMD GPU Unoptimized	NVIDIA GPU Unoptimized	AMD CPU Unoptimized
Dense Linear Algebra	LU Decomposition			
Sparse Linear Algebra	Matrix Multiplication			
Spectral Methods	FFT			
N-Body Methods	GEM	GEM	GEM	GEM
Structured Grids	SRAD			
Unstructured Grids	CFD Solver			
MapReduce	StreamMR	StreamMR		
Combinational Logic	CRC			
Graph Traversal	BFS, Bitonic Sort			
Dynamic Programming	Needleman-Wunsch		Smith-Waterman	
Backtrack and Branch-and-Bound				
Graphical Models	Hidden Markov Model			
Finite State Machines	Temporal Data Mining		TDM	

CU2CL

wfeng@vt.edu, 540.231.1192

Status of OpenCL & the 13 Dwarfs

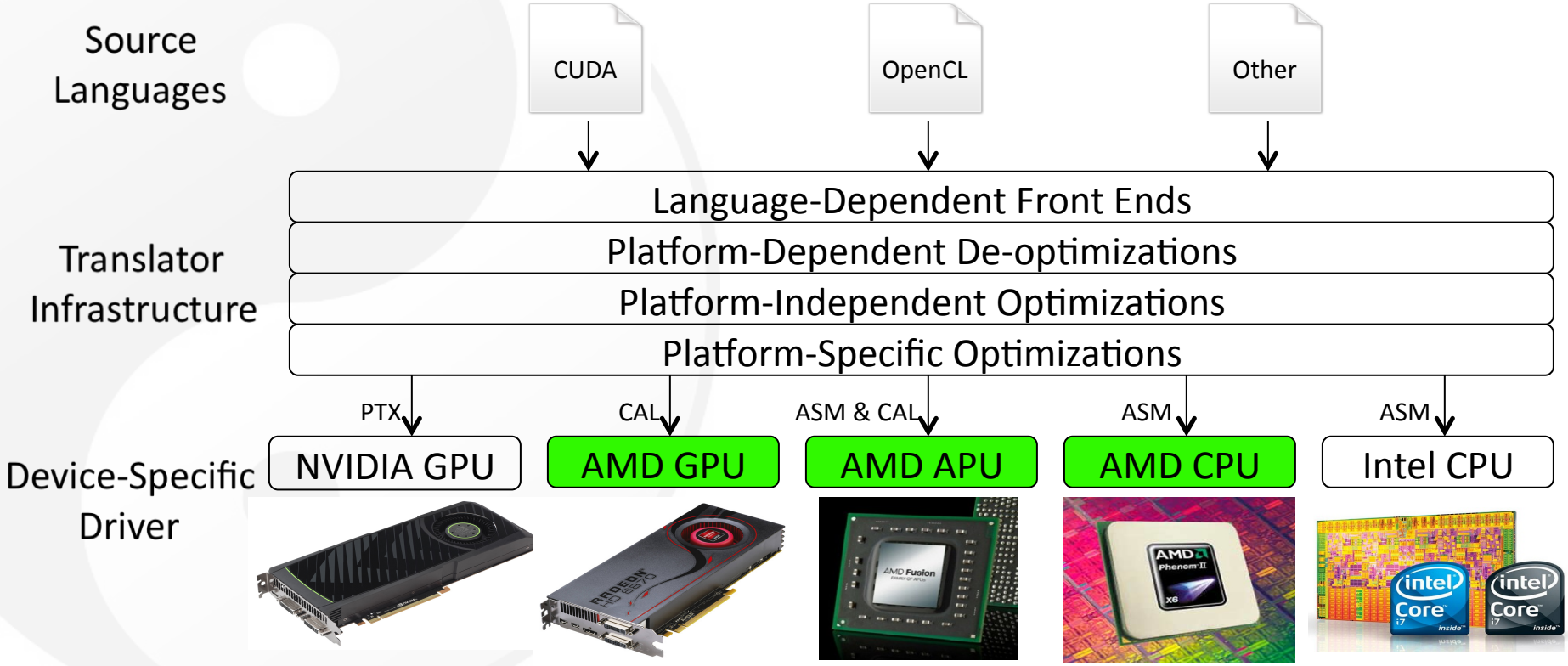
2009 – 2011

Dwarf	Done
Dense linear algebra	LU Decomposition
Sparse linear algebra	Matrix Multiplication
Spectral methods	FFT
N-Body methods	GEM
Structured grids	SRAD
Unstructured grids	CFD solver
MapReduce	
Combinational logic	CRC
Graph traversal	Breadth-First Search (BFS)
Dynamic programming	Needleman-Wunsch
Backtrack and branch-and-bound	
Graphical models	Hidden Markov Model
Finite state machines	Temporal Data Mining

88x → 371x

wfeng@vt.edu, 540.231.1192

Ecosystem for Source-to-Source Translation



wfeng@vt.edu, 540.231.1192

Our Solutions

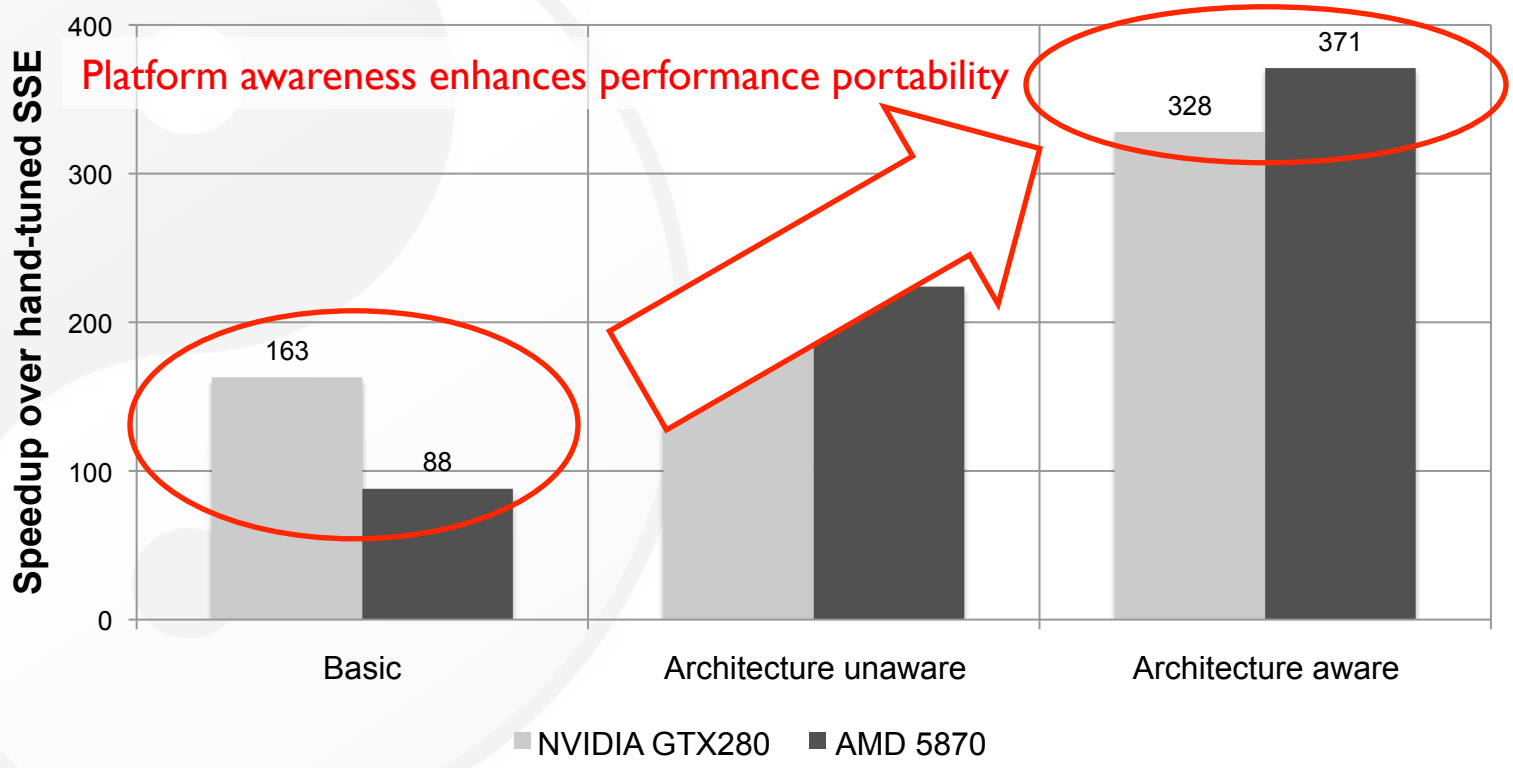
- Functional Portability (2 years → real time)
  - ***CU2CL*** (pronounced as "cuticle")
    An Automated CUDA-to-OpenCL Source-to-Source Translator

    OpenMP → OpenCL

    OpenCL → (AutoESL+ GCC) → FPGA
- Performance Portability (88x → 371x)
  - M. Daga, T. Scogland, and W. Feng, "Architecture-Aware Mapping and Optimizations on a 1600-Core GPU," *17th IEEE Int'l Conf. on Parallel and Distributed Systems*, December 2011.

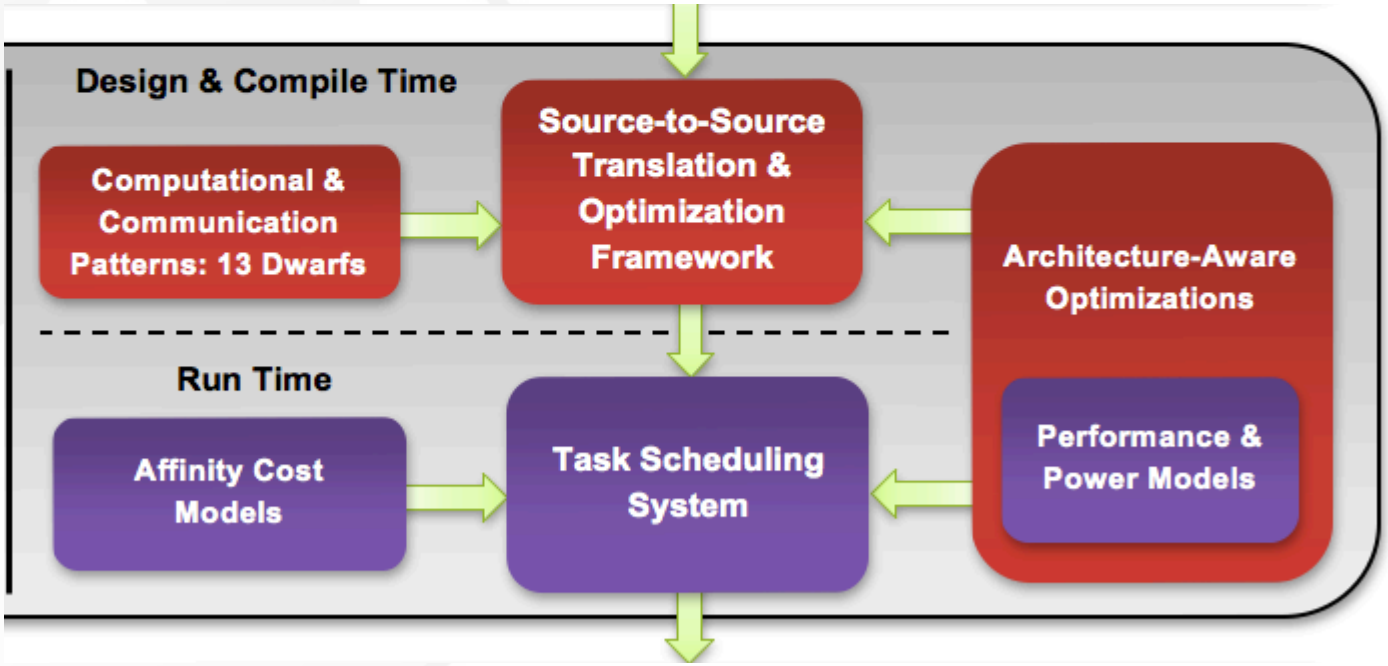
wfeng@vt.edu, 540.231.1192

Potential Due to Optimization



wfeng@vt.edu, 540.231.1192

The Bigger Picture



wfeng@vt.edu, 540.231.1192

CU2CL: Acknowledgments

- Collaborators
  - Gabriel Martinez, M.S.
  - Mark Gardner, Ph.D.
- Infrastructure
  - Clang compiler and LLVM framework

NSF CHREC NSF Center for High-Performance Reconfigurable Computing
AMD HARRIS
Los Alamos NATIONAL LABORATORY XILINX

wfeng@vt.edu, 540.231.1192

Conclusion:

General Approach for Translating CUDA to OpenCL

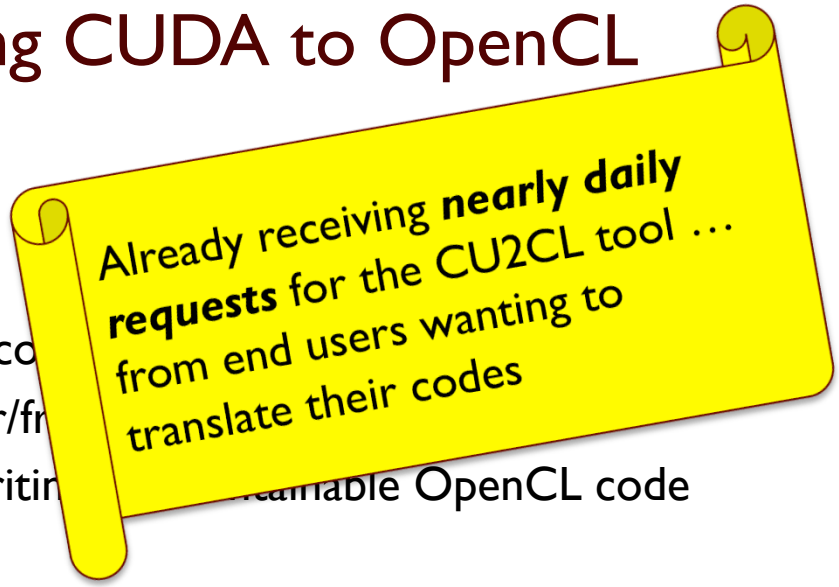
- First Instantiation: CU2CL
  - Profile
    - Approximately 2000 source lines of code
    - Extends open-source Clang compiler/framework
    - AST-driven, string-based source rewriting → maintainable OpenCL code
  - Utility
    - Eliminates the hand translation of virtually all CUDA constructs
    - Translated OpenCL performance = hand-translated

wfeng@vt.edu, 540.231.1192

Conclusion:

General Approach for Translating CUDA to OpenCL

- First Instantiation: CU2CL
  - Profile
    - Approximately 2000 source lines of co
    - Extends open-source Clang compiler/fr
    - AST-driven, string-based source rewritin ainable OpenCL code
  - Utility
    - Eliminates the hand translation of virtually all CUDA constructs
    - Translated OpenCL performance = hand-translated



wfeng@vt.edu, 540.231.1192

Our Solutions

- ***Functional Portability (2 years → real time)***
  - ***CU2CL*** (pronounced as "cuticle")
    An Automated CUDA-to-OpenCL Source-to-Source Translator

    OpenMP → OpenCL

    OpenCL → (AutoESL+ GCC) → FPGA
- ***Performance Portability (88x → 371x)***
  - M. Daga, T. Scogland, and W. Feng, "Architecture-Aware Mapping and Optimizations on a 1600-Core GPU," *17th IEEE Int'l Conf. on Parallel and Distributed Systems*, December 2011.

wfeng@vt.edu, 540.231.1192

Conclusion

General Approach for Translating CUDA to OpenCL

- First Instantiation: CU2CL
  - Profile
    - Approximately 2000 source lines of code
    - Extends open-source Clang compiler/framework
    - AST-driven, string-based source rewriting → maintainable OpenCL code
  - Utility
    - Eliminates the hand translation of virtually all CUDA constructs
    - Translated OpenCL performance = hand-translated
  - ***Future Work***
    - ***A translation ecosystem that also delivers performance portability***

wfeng@vt.edu, 540.231.1192