An Ecosystem for the New HPC: Heterogeneous Parallel Computing

Wu FENG
Dept. of Computer Science and Dept. of Electrical & Computer Engineering
Virginia Bioinformatics Institute

VirginiaTech
Invent the Future

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

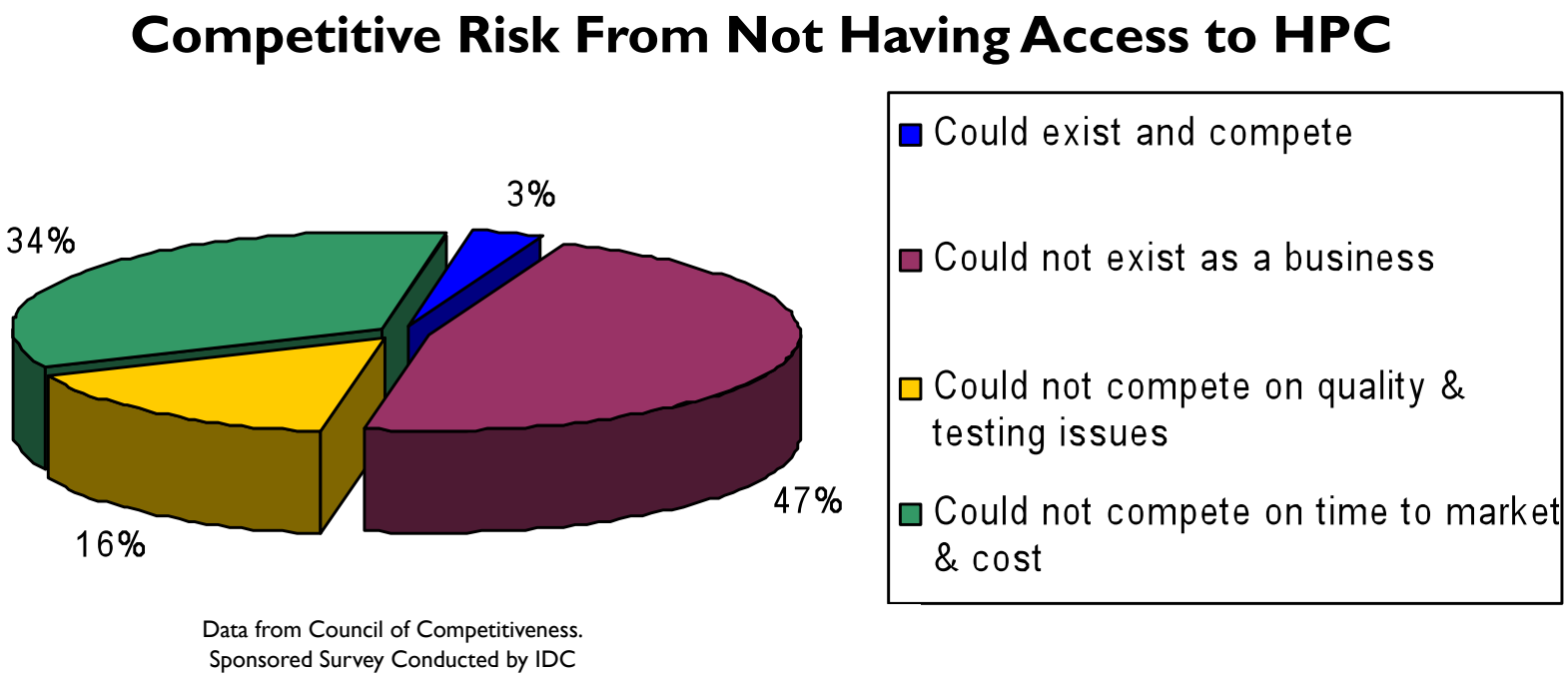
SyNeRG
synergy.cs.vt.edu

Japanese 'Computnik' Earth Simulator Shatters U.S. Supercomputer Hegemony

Tokyo 20 April 2002 *The Japanese Earth Simulator is on-line and producing results that alarm the USA, that considered itself as being leading in supercomputing technology. With over 35 Tflop/s, it five times outperforms the Asci White supercomputer that is leading the current TOP500 list. No doubt that position is for the Earth Simulator, not only for the next list, but probably even for*

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Importance of High-Performance Computing (HPC)



© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Computnik 2.0?

HOME PAGE | TODAY'S PAPER | VIDEO | MOST POPULAR | TIMES TOPICS

The New York Times — Business Day Technology

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH

Search Technology — Go

Inside Technology — Internet | Start-Ups | Business Computing

China Wrests Supercomputer Title From U.S.

By ASHLEE VANCE
Published: October 28, 2010

A Chinese scientific research center has built the fastest supercomputer ever made, replacing the United States as maker of the swiftest machine,

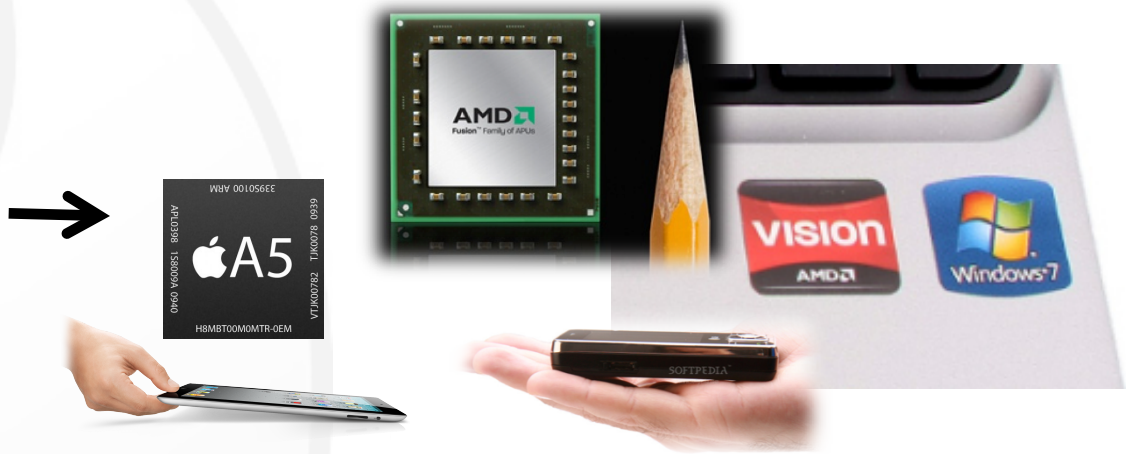
RECOMMEND
TWITTER
LINKEDIN

- The Second Coming of Computnik? Computnik 2.0?
  - No … "only" 43% faster than the previous #1 supercomputer, *but*
    - → $20M cheaper than the previous #1 supercomputer
    - → 42% less power consumption
- The Second Coming of the "Beowulf Cluster" for HPC
  - The further commoditization of HPC

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Vision

- **Commoditizing *personal supercomputing* for the masses**



2.5 x 10^{15} floating-pt ops per sec
= 2.5 petaflops (Linpack benchmark)

iPad2: 1.5 x 10^{9} flops = 1.5 gigaflops
1.5 gigaflops/iPad2 * 9.3M iPad2 (Q2 2011)
= 14 petaflops (Linpack benchmark extrapolated)

- **A software ecosystem**
  - **… for supporting heterogeneous parallel computing**
  - **… by exploiting intra-node parallelism**
  - **… to commoditize personal supercomputing for the masses**

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

CPU Core Counts …

- Doubling every 18-24 months
  - 2006: 2 cores
    - Examples: AMD Athlon 64 X2, Intel Core Duo
  - 2010: 8-12 cores
    - Examples: AMD Magny Cours, Intel Nehalem EX
- Penetrating all markets …
  - Desktops
  - Laptops: Most in this room are multicore
  - Tablets: Apple iPad 2, HP TX1000, Sony S2
  - Cell Phones: LG Optimus 2X, Motorola Droid X2

A world of ubiquitous parallelism …

… how to extract performance … and then scale out

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

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.

- How do we lower the cost of concurrency?

† 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.)

© W. Feng, May 2012
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.

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.

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Project Goal

An Ecosystem for Heterogeneous Parallel Computing

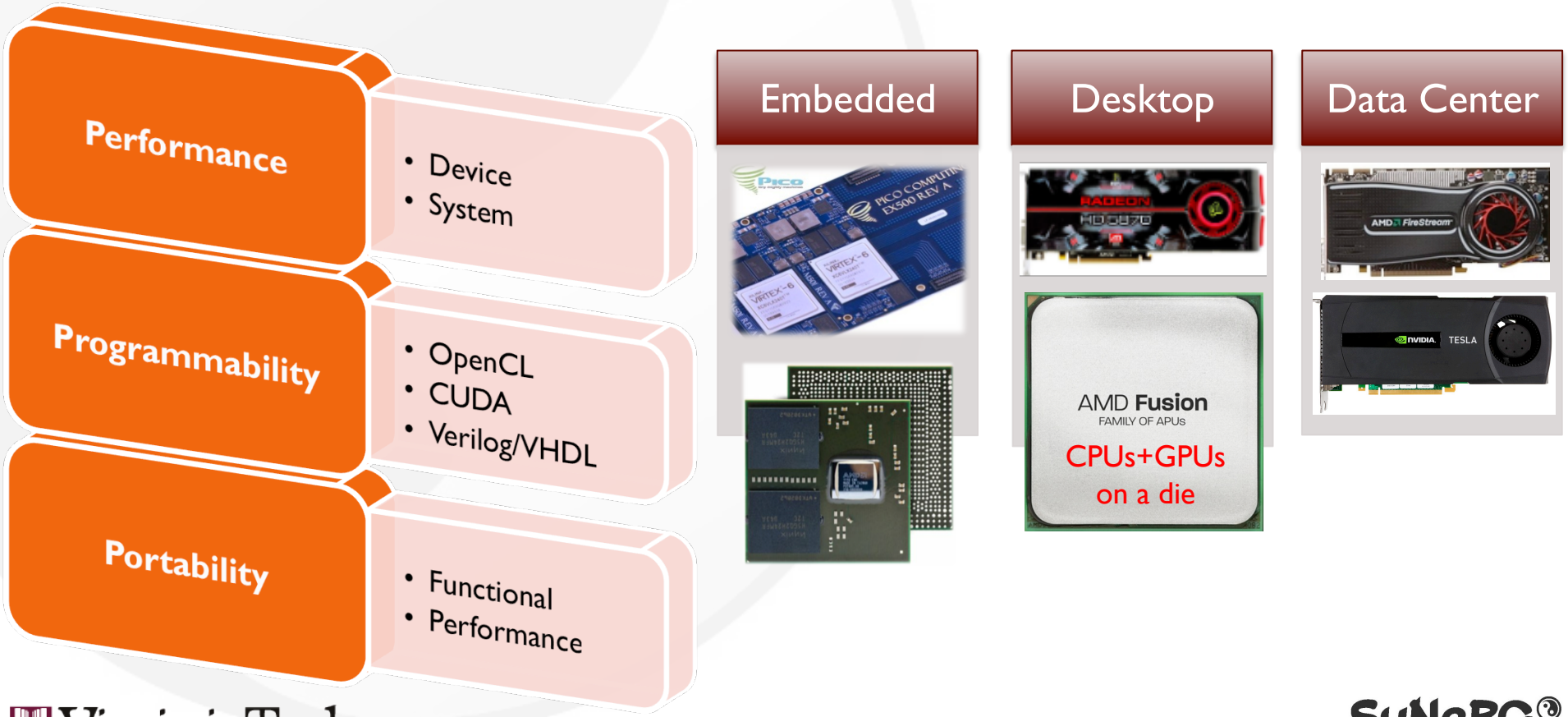
- Deliver personalized supercomputing to the masses
  - Heterogeneity of hardware devices for a "cluster on a chip" plus …
  - Enabling software that tunes the parameters of the hardware devices with respect to performance, power, and programmability

  via a benchmark suite of computational dwarfs and apps

©W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Multi-Dimensional Understanding of 3 P's

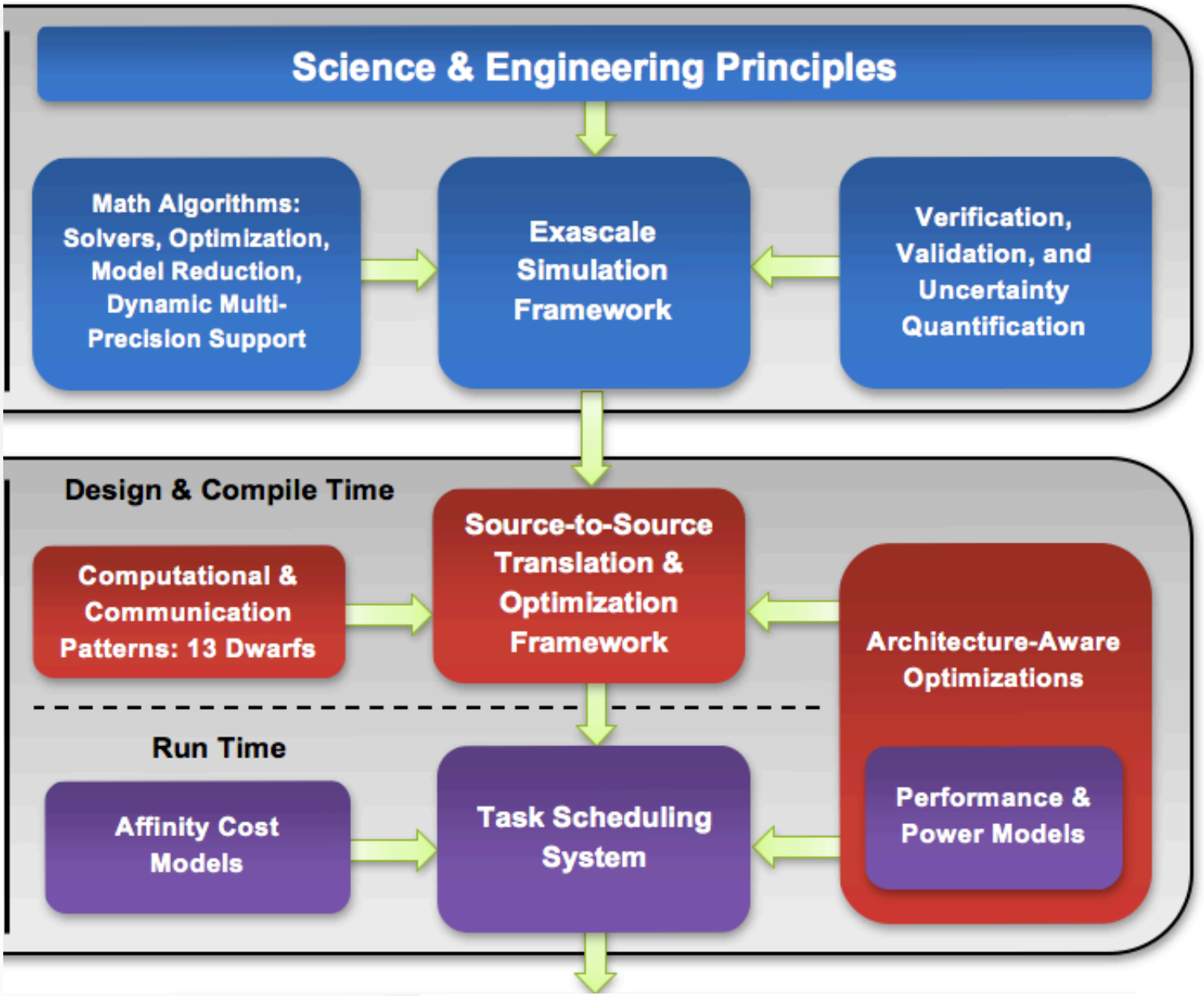
- A multi-dimensional understanding of how to optimize ***performance***, ***programmability***, ***portability*** or some combination thereof



©W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Design of Composite Structures

Software Ecosystem

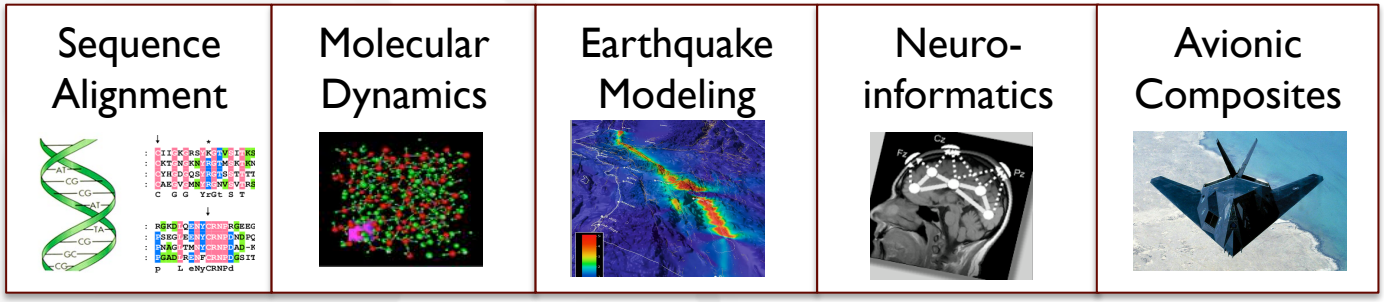


Heterogeneous Parallel Computing (HPC) Platform

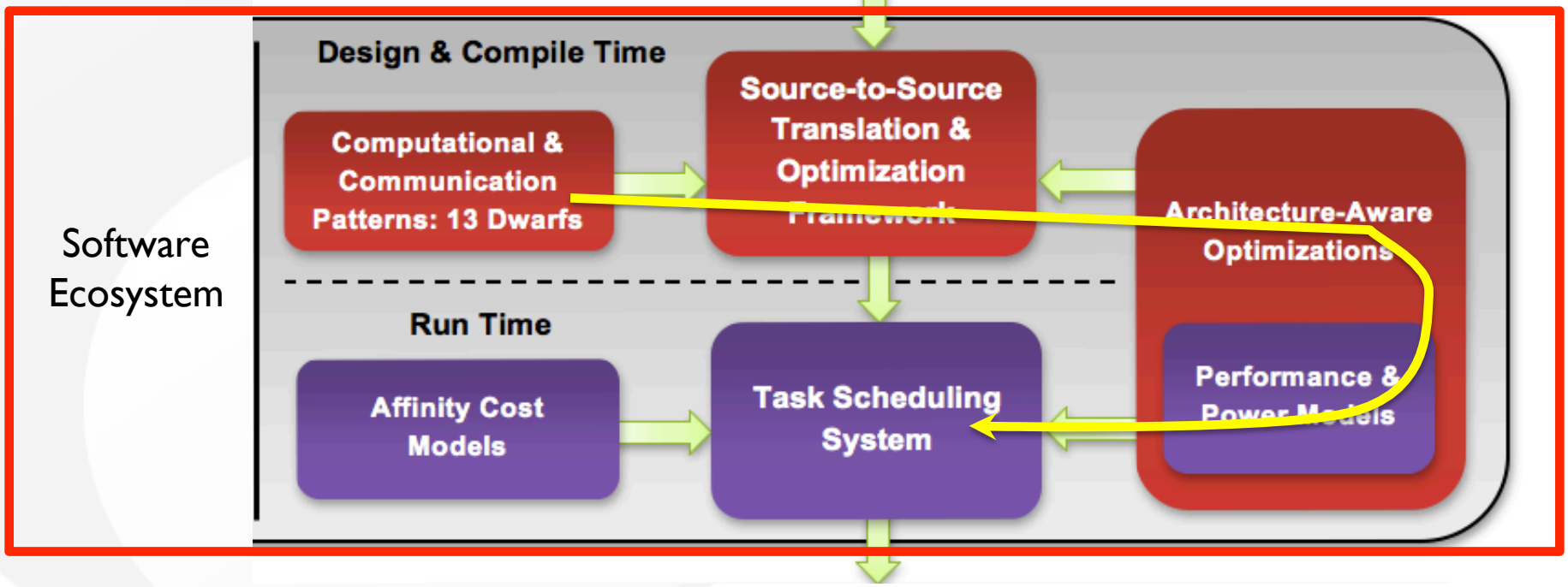
© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

An Ecosystem for Heterogeneous Parallel Computing

Applications



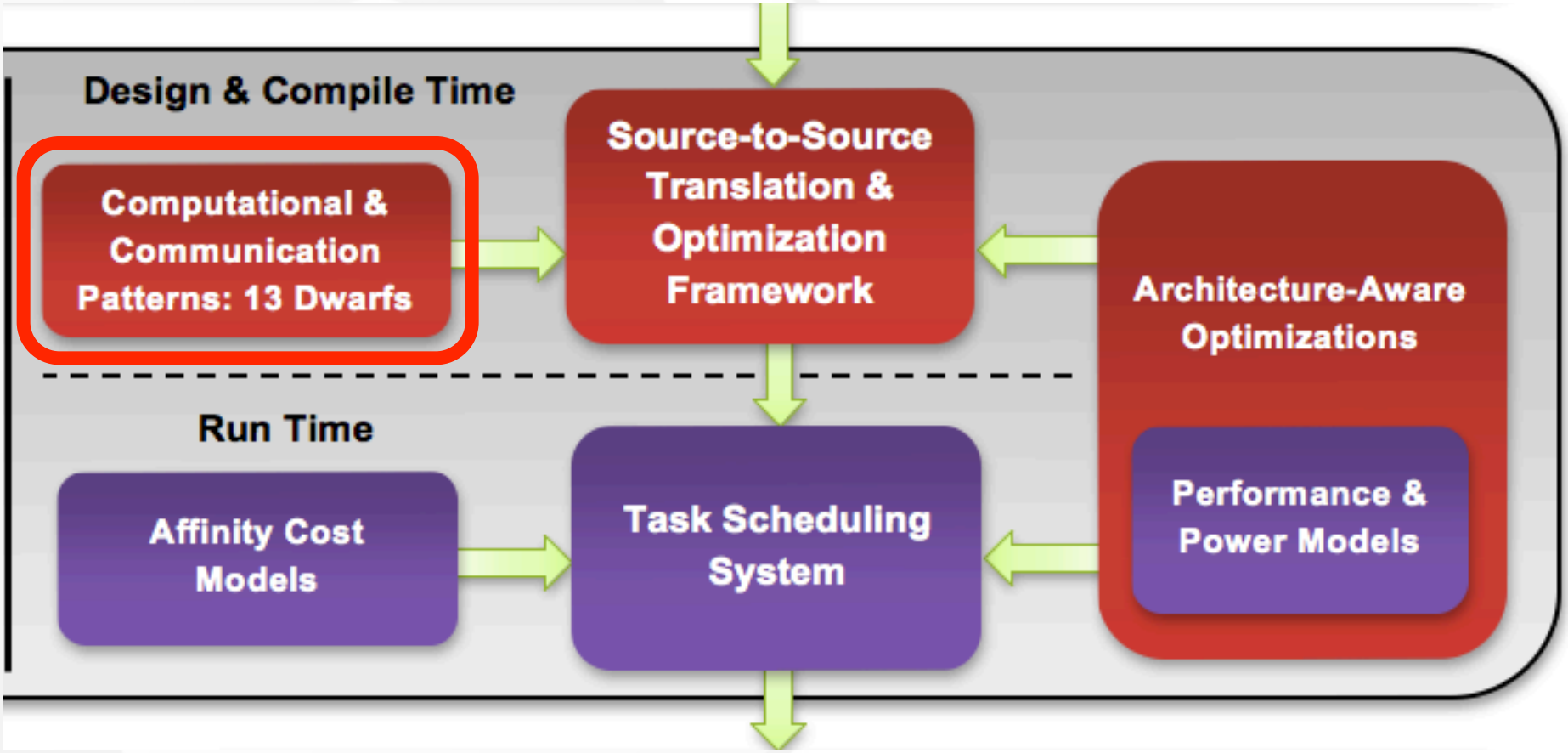
Software Ecosystem



Heterogeneous Parallel Computing (HPC) Platform

©W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

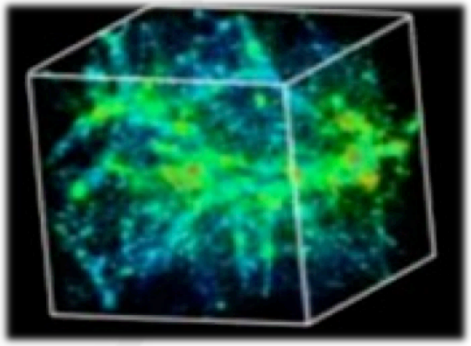
Roadmap



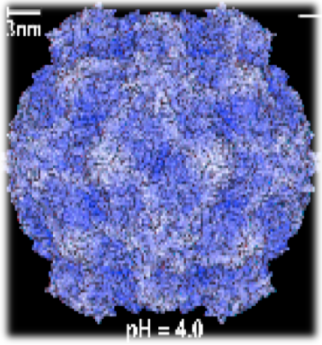
© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Example of a Computational Dwarf: N-Body

- Computational Dwarf: *Pattern of computation & communication … that is common across a set of applications*
- 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

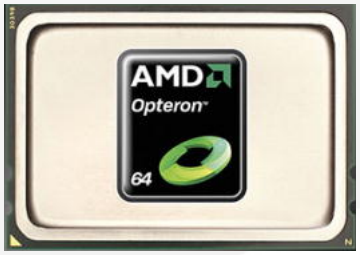


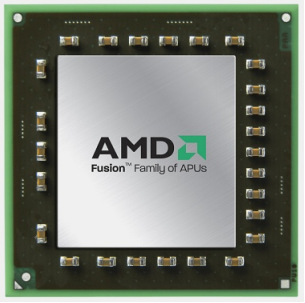
GEM: Molecular Modeling

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

First Instantiation: OpenCL & the 13 Dwarfs

- Goal
  - 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)

CHREC
NSF Center for High-Performance Reconfigurable Computing

AMD HARRIS XILINX

Air Force Research Laboratory · Department of Defense · National Security Agency · Los Alamos National Laboratory

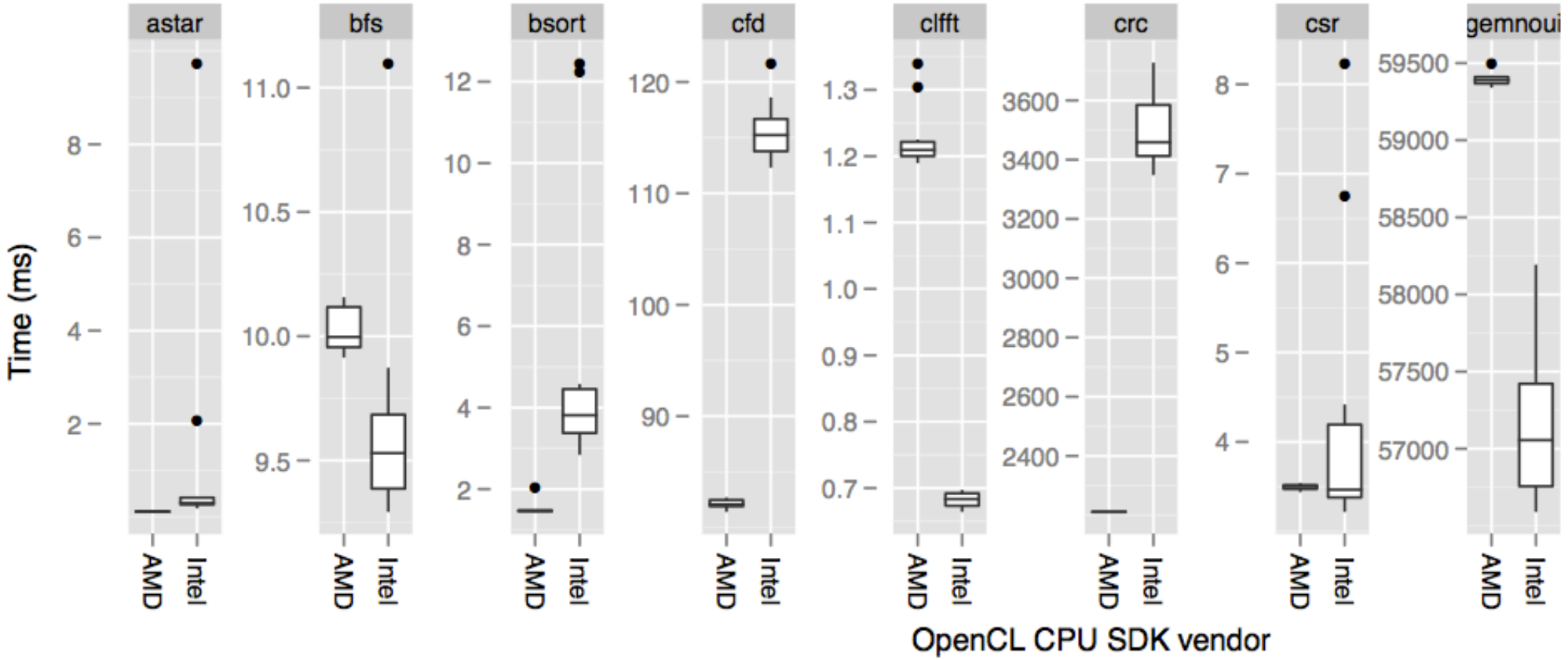
© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Status of OpenCL and the 13 Dwarfs

Dwarf	Done	In progress
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		StreamMR
Combinational logic	CRC	
Graph traversal	Breadth-First Search	
Dynamic programming	Needleman-Wunsch	
Backtrack and Branch-and-Bound		N-Queens, Traveling Salesman
Graphical models	Hidden Markov Model	
Finite state machines	Temporal Data Mining	

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Execution Time (ms) of Dwarfs with SDKs



© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Experimental Set-Up: Machines and Workload

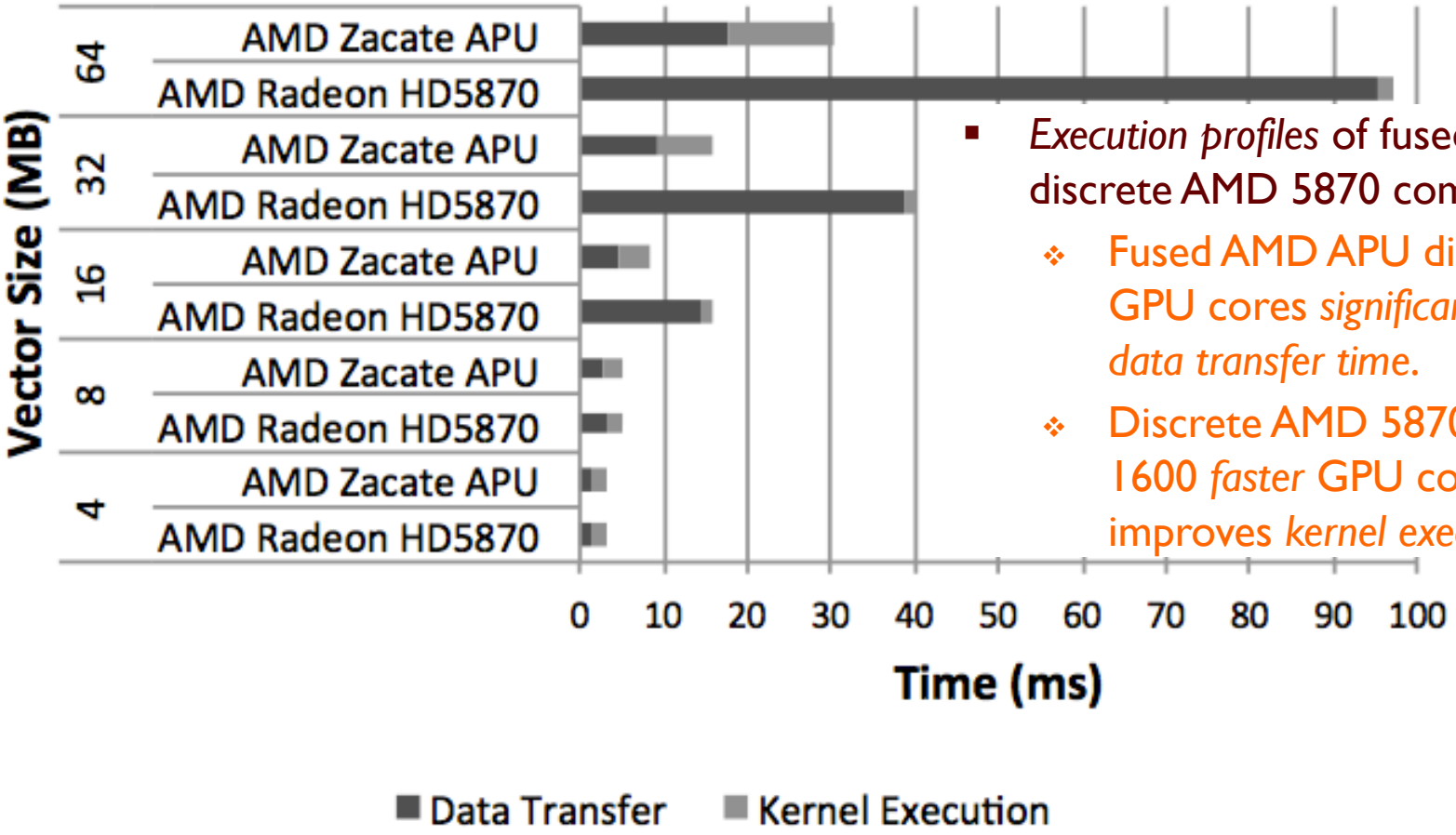
Platform	AMD Zacate APU	AMD Radeon HD 5870	AMD Radeon HD 5450
Stream Processors	80	1600	80
Compute Units	2	20	2
Memory Bus Type	NA	GDDR5	DDR3
Device Memory	192 MB	1024 MB	512 MB
Local Memory	32 KB	32 KB	32 KB
Max. Workgroup Size	256 Threads	256 Threads	128 Threads
Core Clock Frequency	492 MHz	850 MHz	675 MHz
Peak FLOPS	80 GFlops/s	2720 GFlops/s	104 GFlops/s
Host:			
Processor	AMD Engg. Sample @1.6 GHz	Intel Xeon E5405 @2.0 GHz	Intel Celeron 430 @1.8 GHz
System Memory	2 GB (NA)	2 GB DDR2	2 GB DDR2
Cache	L1: 32K, L2: 512K	L1: 32K, L2: 6M	L1: 32K, L2: 512K
Kernel	Ubuntu 2.6.35.22	Ubuntu 2.6.28.19	Ubuntu 2.6.32.24

- OpenCL and the 13 Dwarfs
  - Sparse Linear Algebra: SpMV
  - N-body: Molecular Modeling
  - Spectral: FFT
  - Dense Linear Algebra: Scan and Reduce (SHOC @ ORNL)

M. Daga, A. Aji, W. Feng, "On the Efficacy of a Fused CPU+GPU Processor for Parallel Computing," *Symposium on Application Accelerators in High Performance Computing*, Jul. 2011.

©W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Performance: Reduction (Dense Linear Algebra)



- *Execution profiles* of fused AMD APU & discrete AMD 5870 complementary.
  - Fused AMD APU die with only 80 GPU cores *significantly* improves *data transfer time.*
  - Discrete AMD 5870 die with 1600 *faster* GPU cores *significantly* improves *kernel execution time.*

© W. Feng, May 2012

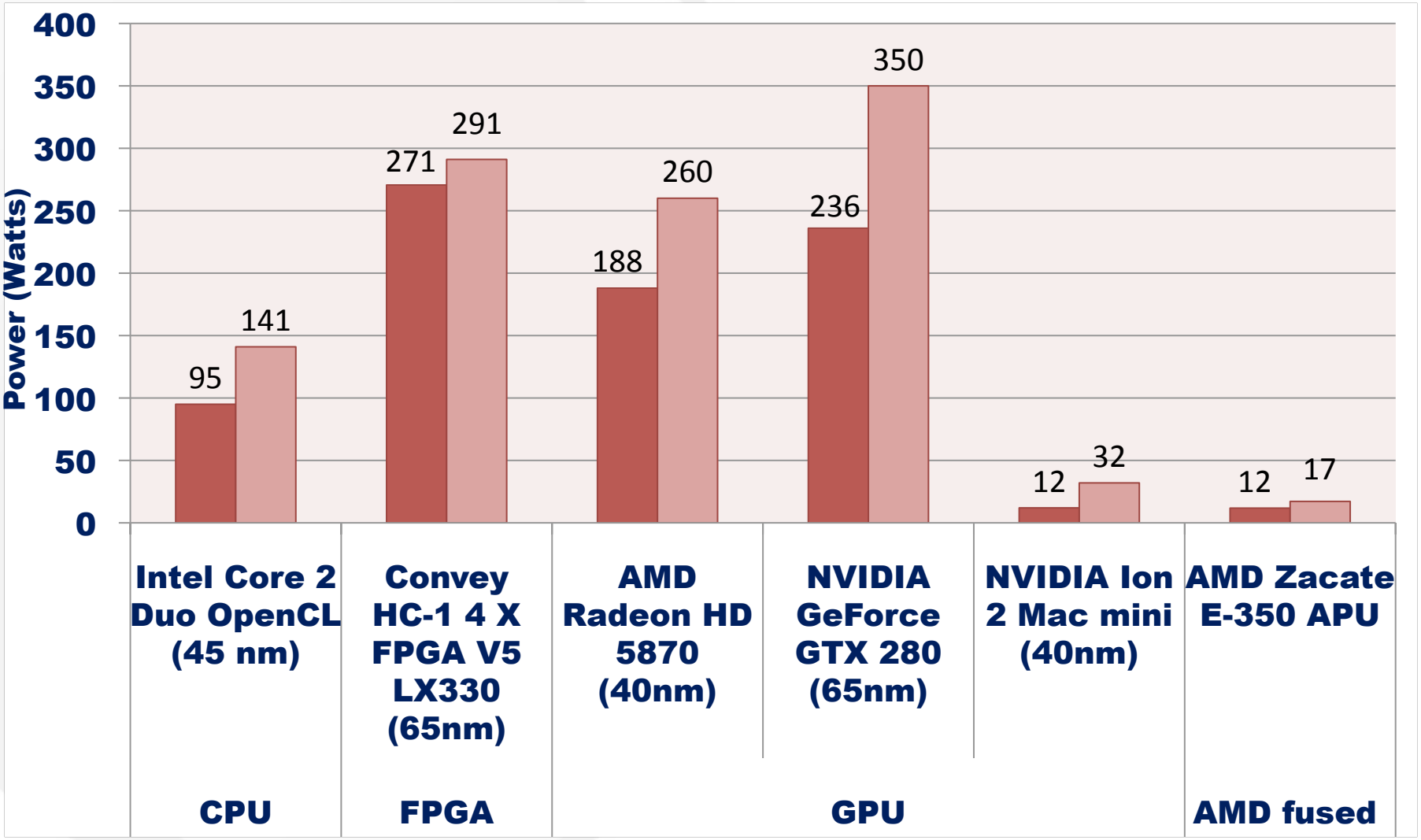
wfeng@vt.edu, 540.231.1192

System Power

- AMD Fusion APU
  - At idle: 12 watts
  - At load: 17 watts
    (Spectral Method: FFT)
  - At load: 20 watts
    (N-body: Molecular Modeling)
- AMD Radeon HD 5870 Machine w/ 2-GHz Intel Xeon E5405
  - At idle: 188 watts
  - At load: 260 watts

©W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Total System Power: Idle vs. At Load (w/ FFT)



© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Status of OpenCL & the 13 Dwarfs

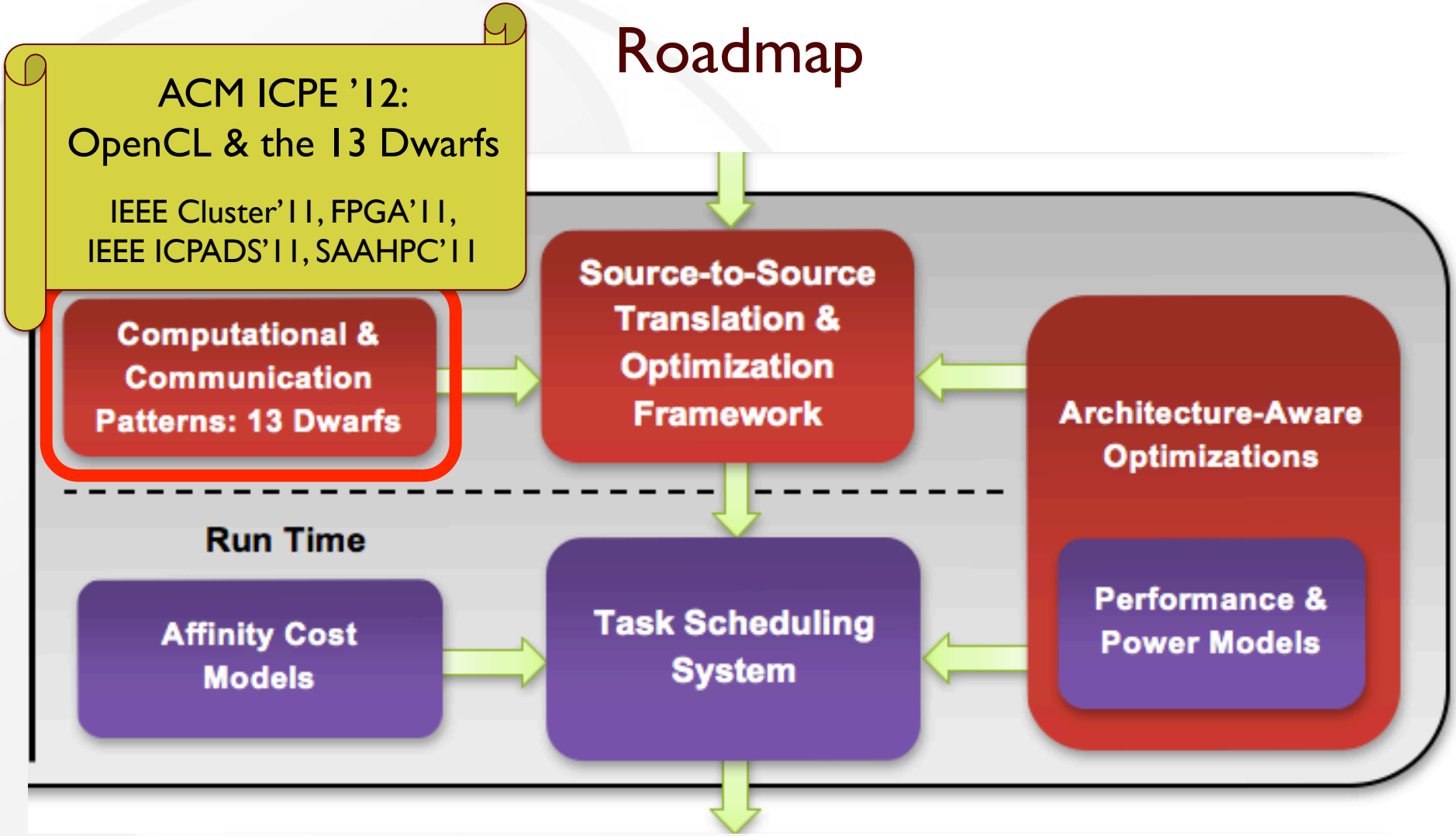
2009 – 2011

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

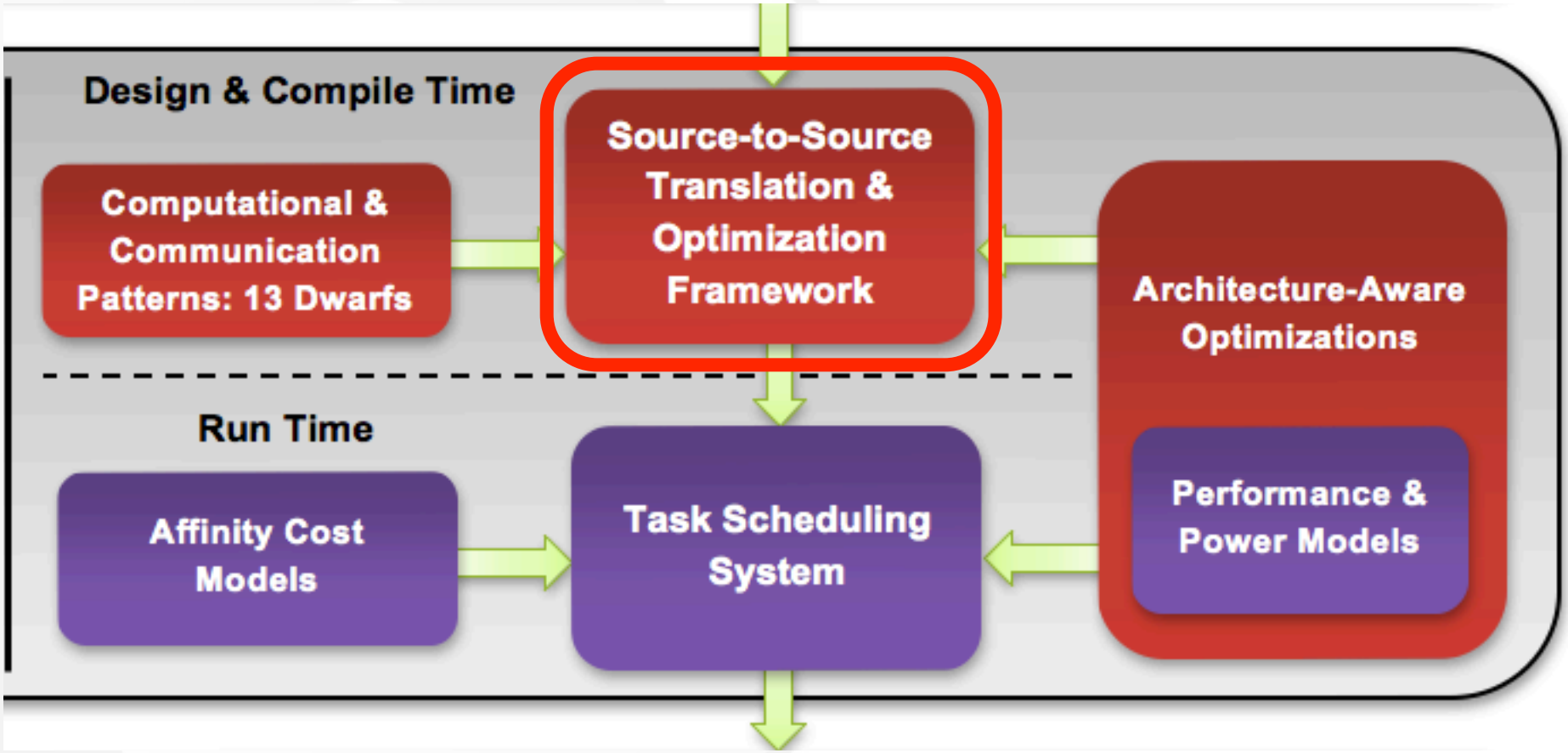
© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Roadmap



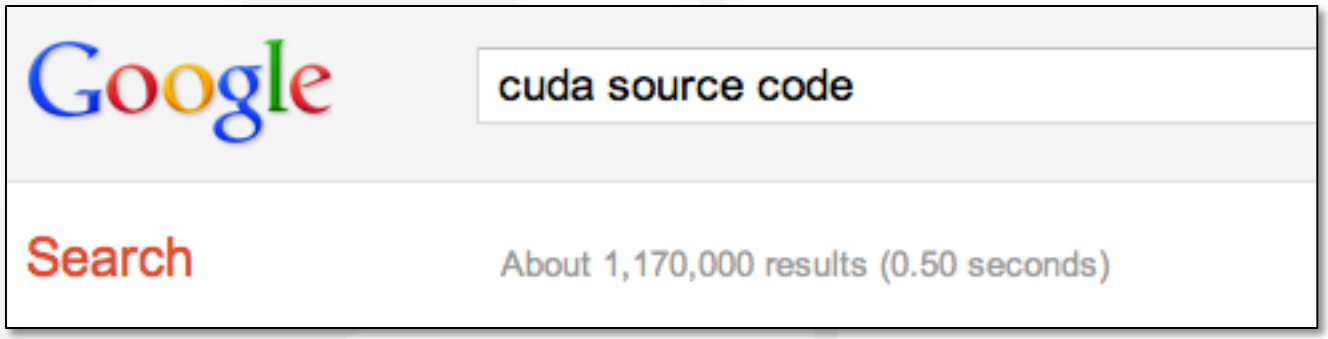
© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Roadmap

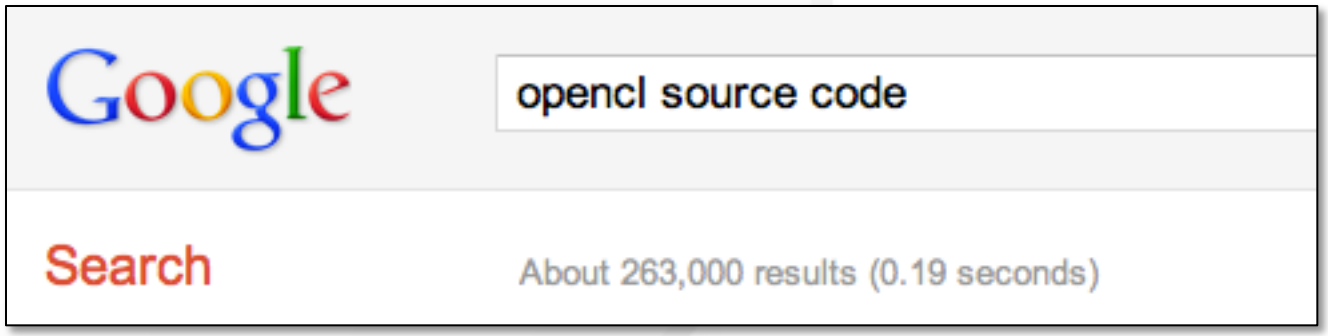


© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Prevalence



First public release February 2007



First public release December 2008

Search Date: 2011-09-14

©W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

CUDA-Accelerated Applications

GOVERNMENT & DEFENSE
Ikena: Imagery Analysis and Video Forensics
Signal Processing Library: GPU VSIPL
IDL and MATLAB® Acceleration: GPULib
GIS: Manifold

MOLECULAR DYNAMICS, COMPUTATIONAL CHEMISTRY
OpenMM library for molecular dynamics on GPUs
GROMACS using OpenMM
NAMD molecular dynamics
VMD visualization of molecular dynamics
HOOMD molecular dynamics
Acellera: ACEMD bio-molecular dynamics package
BigDFT: DFT (Density functional theory) electronic structure
MDGPU
GPUGrid.net

LIFE SCIENCES, BIO-INFORMATICS
GPU HMMER
DNA Sequence alignment: MUMmerGPU
LISSOM: model of human neocortex using CUDA
Silicon Informatics: AutoDock

ELECTRODYNAMICS AND ELECTROMAGNETIC
Acceleware: FDTD Solver
Acceleware: EM Solutions
Remcom XStream FDTD
SPEAG Semcad X
CST Microwave Studio
Quantum electrodynamics library
GPMAD : Particle beam dynamics simulator

MEDICAL IMAGING, CT, MRI
RealityServer
GPULib:IDL acceleration
Acceleware: Imaging Solutions
Digisens: SnapCT tomographic reconstruction software
Techniscan: Whole Breast Ultrasound Imaging System

OIL & GAS
Acceleware: Kirchoff and Reverse Time Migration
SeismicCity: 3D seismic imaging for prestack depth migration
OpenGeoSolutions: Spectral decomposition and inversion
Mercury Computer systems: 3D data visualization
ffA: 3D Seismic processing software
Headwave: Prestack data processing

FINANCIAL COMPUTING AND OPTIONS PRICING
SciComp: derivatives pricing
Hanweck: options pricing
Exegy: Risk Analysis
Aqumin: 3D Visualization of market data
Level 3 Finance
OnEye (Australia): Accelerated Trading Solutions
Arbitragis Trading

MATLAB, LABVIEW, MATHEMATICA, R
CUDA Acceleration for MATLAB
Accelereyes: Jacket™ engine for MATLAB
GPULib: mathematical functions for IDL and MATLAB
Integrating Simulink with CUDA using S-functions
Enabling GPU Computing in the R Statistical Environment
Mathematica plug-in for CUDA
National Instruments LabView for NVIDIA GPUs

ELECTRONIC DESIGN AUTOMATION
Agilent EESof: ADS SPICE simulator
Synopsys: Sentaraus TCAD
Gauda: Optical proximity correction (OPC)

WEATHER AND OCEAN MODELING
CUDA-accelerated WRF code

VIDEO, IMAGING, AND VISION APPLICATIONS
Axxon Intellect Enterprise Video Surveillance Software
Pflow CUDA Plugin for Autodesk 3ds Max
RUINS Shatter CUDA Plug-in for Maya
Bullet 3D Multi-Physics Library with CUDA Support
CUDA Voxel Rendering Engine
Furryball: Direct3D GPU Rendering Plugin for Maya

See: http://www.nvidia.com/object/cuda_app_tesla.html

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

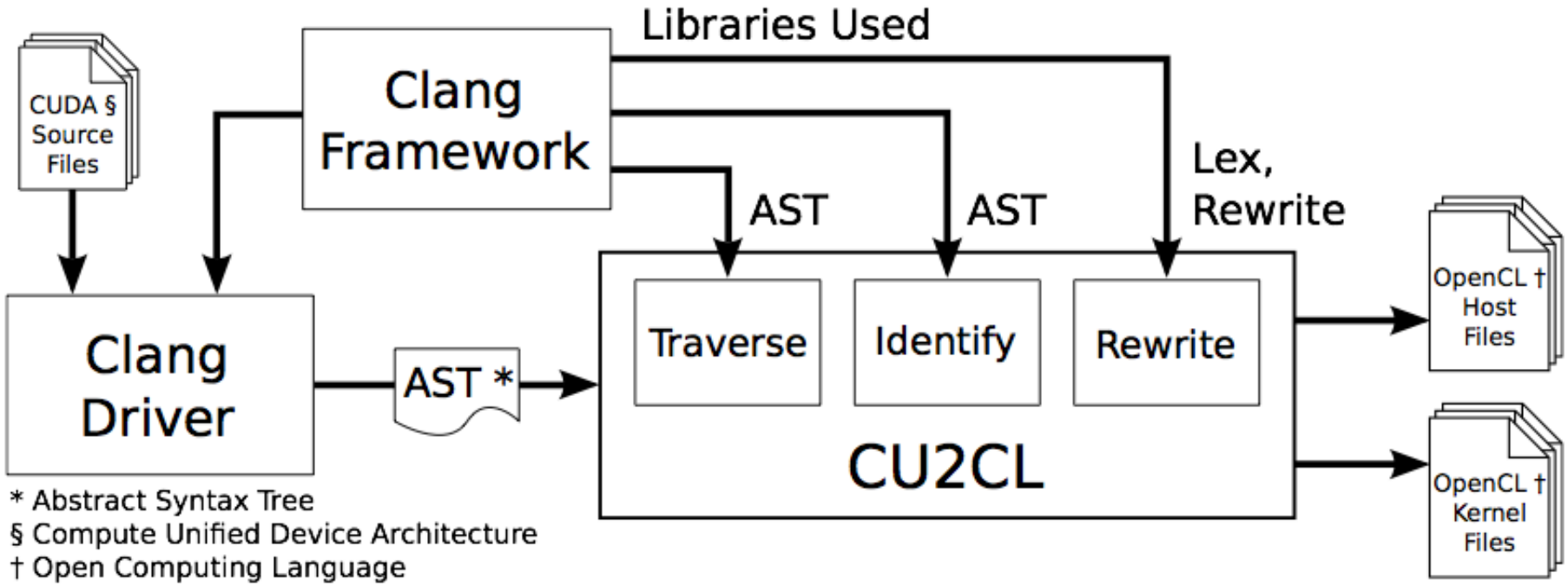
CU2CL: CUDA-to-OpenCL Source-to-Source Translator†

- Implemented as a Clang plug-in to leverage its production-quality compiler framework and target LLVM bytecode.
- Covers primary CUDA constructs found in CUDA C and CUDA run-time API.
- Performs as well as codes manually ported from CUDA to OpenCL.
- *Others: OpenCL-to-CUDA and OpenMP-to-OpenCL*

† "CU2CL: A CUDA-to-OpenCL Translator for Multi- and Many-core Architectures," *17th IEEE Int'l Conf. on Parallel & Distributed Systems (ICPADS)*, Dec. 2011.

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

CU2CL Translation and Performance



- Automatically translated OpenCL codes (via CU2CL) yield similar execution times to manually translated OpenCL codes

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

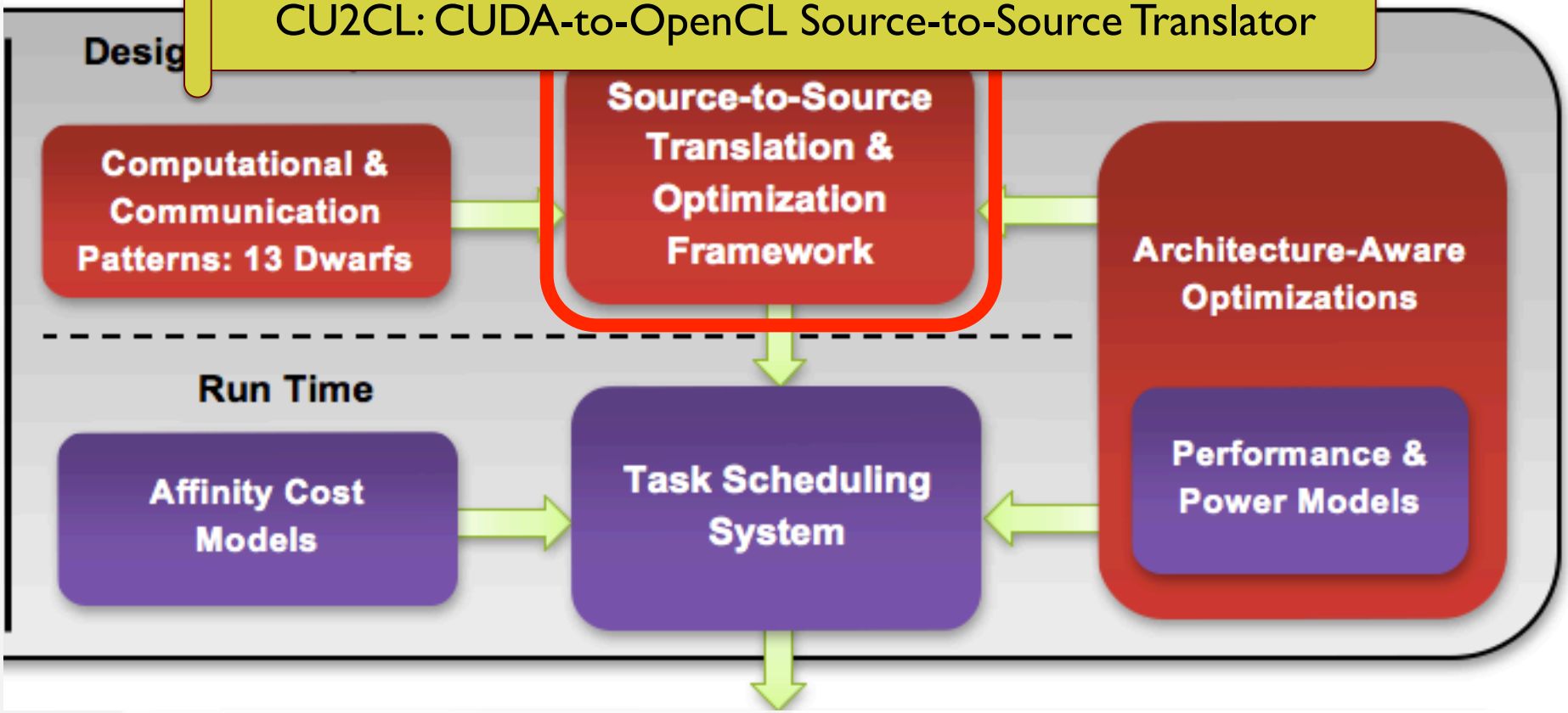
CU2CL Coverage

Source	Application	Lines	Changed	Percentage
CUDA SDK	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
	Hotspot	328	7	97.87
	Needleman-Wunsch	418	0	100.00
	SRAD	541	0	100.00
Virginia Tech	GEM: n-body molecular	2,511	5	99.80

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

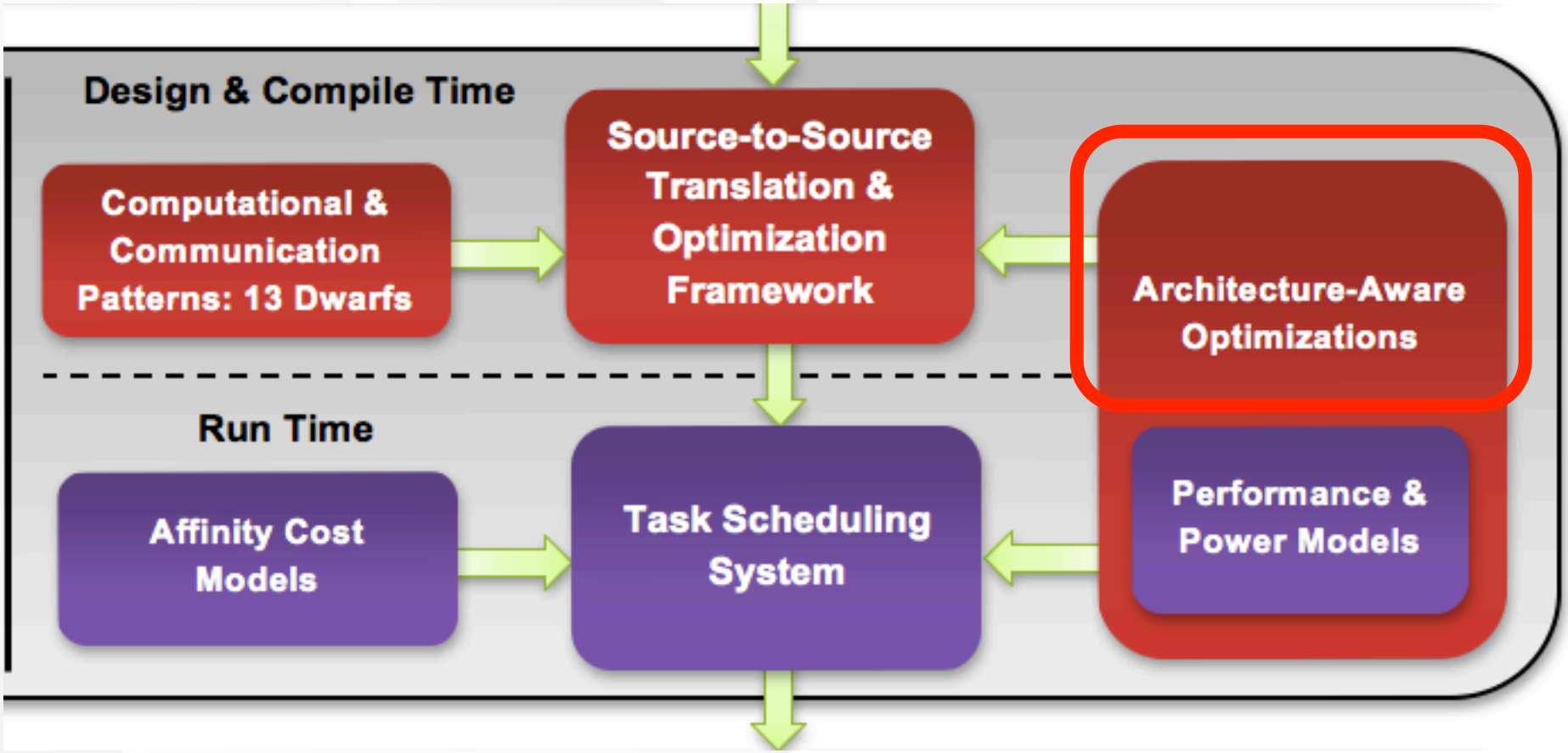
Roadmap

IEEE ICPADS '11:
CU2CL: CUDA-to-OpenCL Source-to-Source Translator



© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

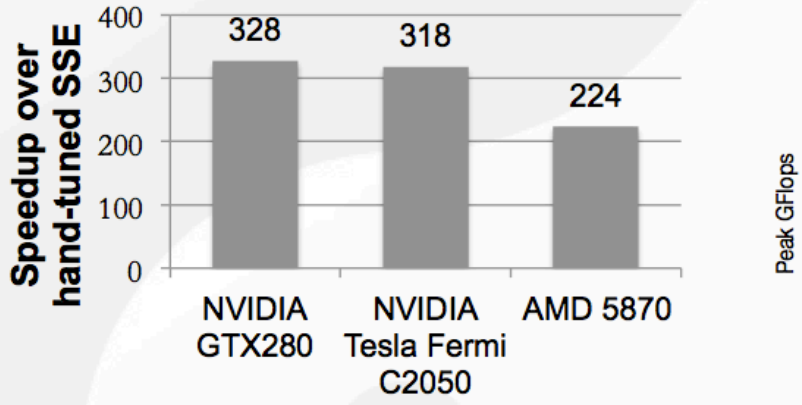
Roadmap



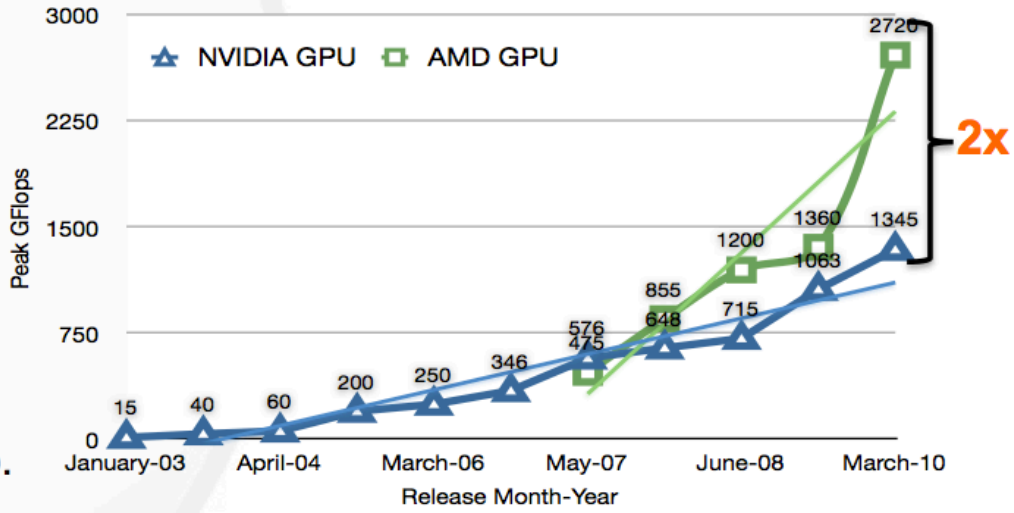
© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Computational Units *Not* Created Equal

- "AMD CPU ≠ Intel CPU" and "AMD GPU ≠ NVIDIA GPU"
- Initial performance of a *CUDA-optimized* N-body dwarf

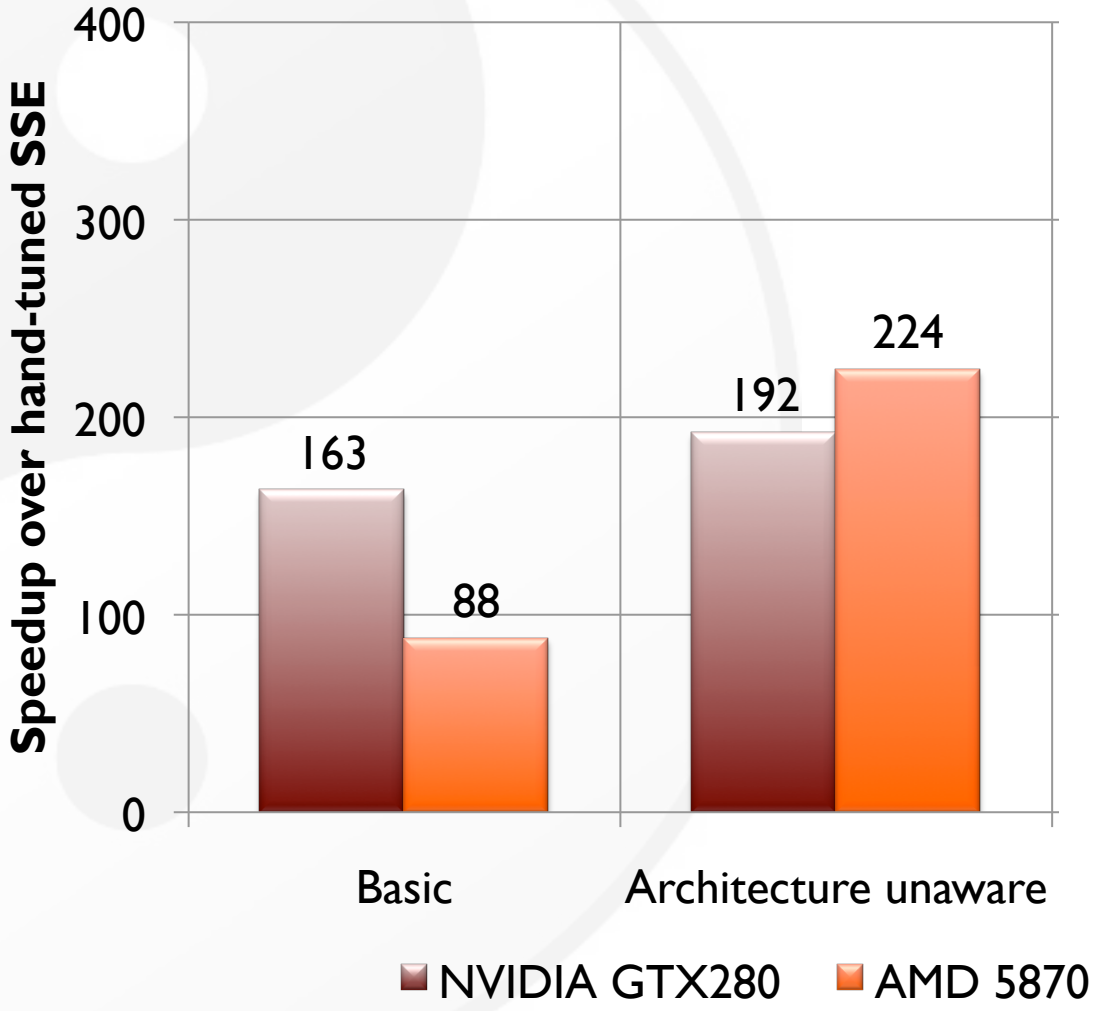


Performance of a Molecular Modeling App.



© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

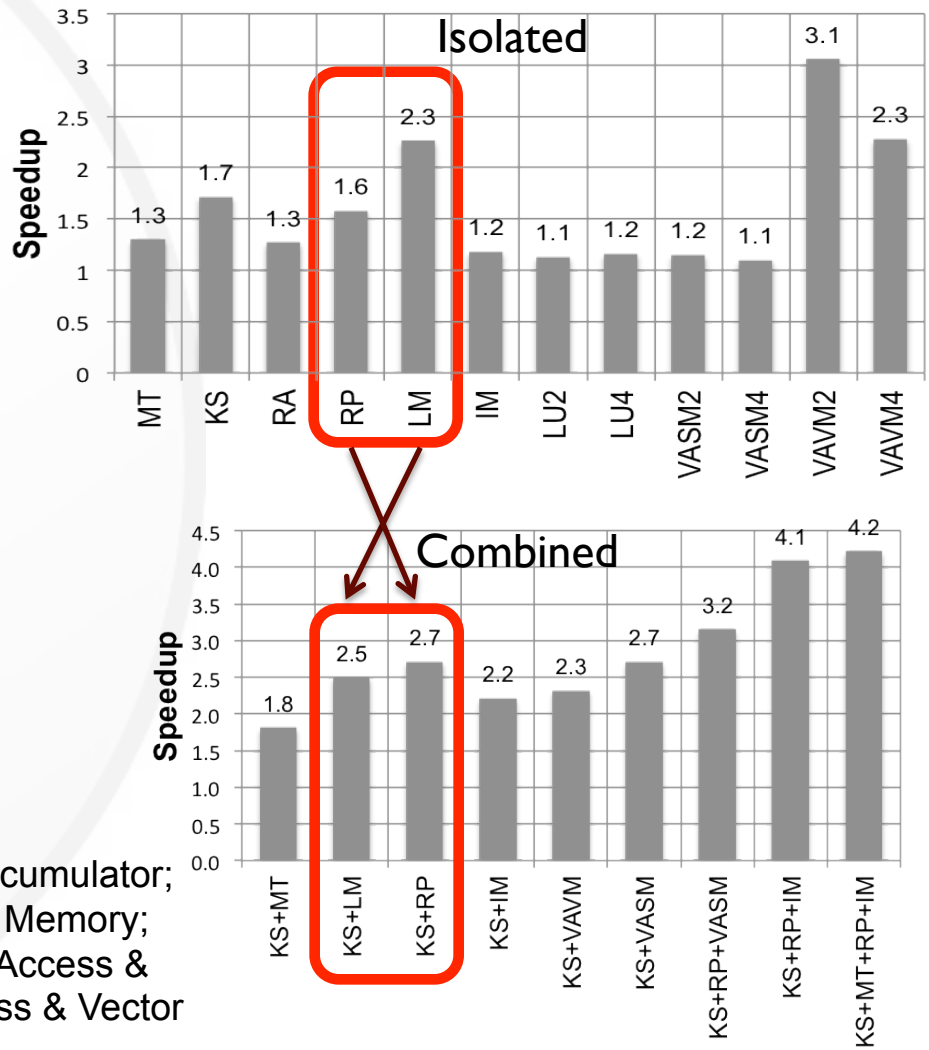
Basic & Architecture-Unaware Execution



© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Architecture-Aware Optimization
(N-body Code for Molecular Modeling)

- Optimization techniques on AMD GPUs
  - Removing conditions → kernel splitting
  - Local staging
  - Using vector types
  - Using image memory
- Speedup over basic OpenCL GPU implementation
  - Isolated optimizations
  - Combined optimizations

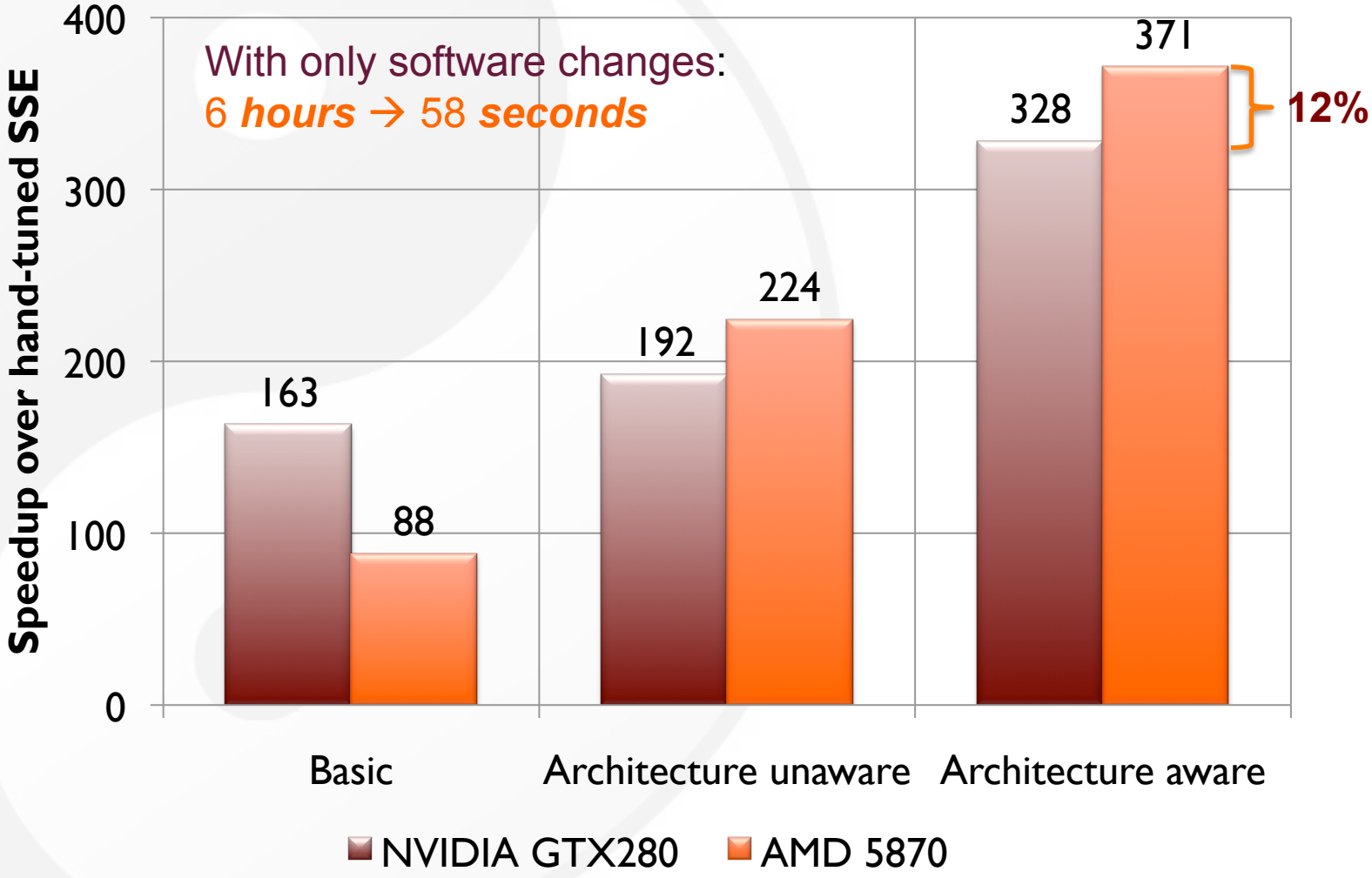


MT: Max Threads; **KS**: Kernel Splitting; **RA**: Register Accumulator; **RP**: Register Preloading; **LM**: Local Memory; **IM**: Image Memory; **LU{2,4}**: Loop Unrolling{2x,4x}; **VASM{2,4}**: Vectorized Access & Scalar Math{float2, float4}; **VAVM{2,4}**: Vectorized Access & Vector Math{float2, float4}

© W. Feng, May 2012

wfeng@vt.edu, 540.231.1192

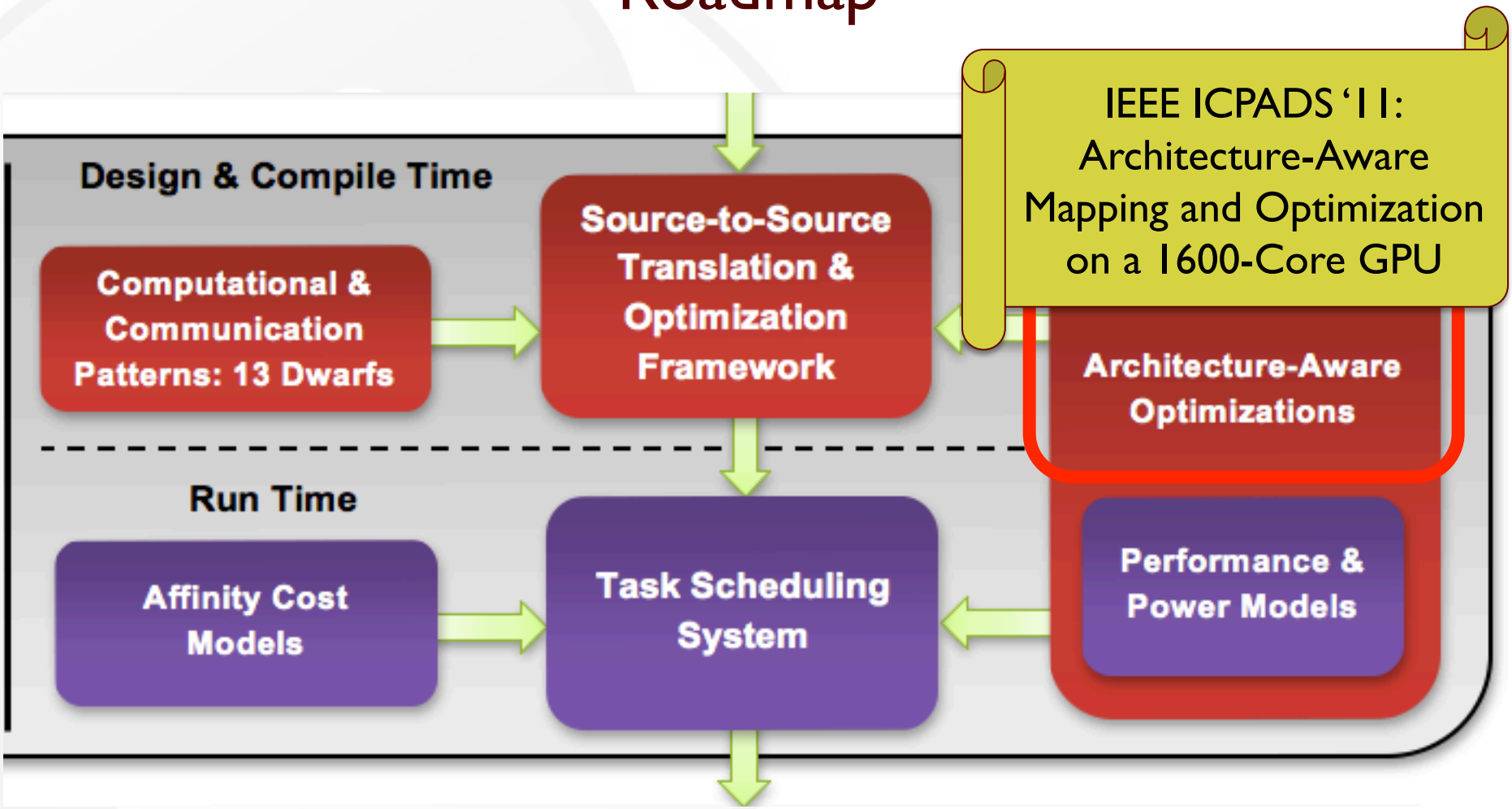
Summary: Architecture-Aware Optimization



© W. Feng, May 2012

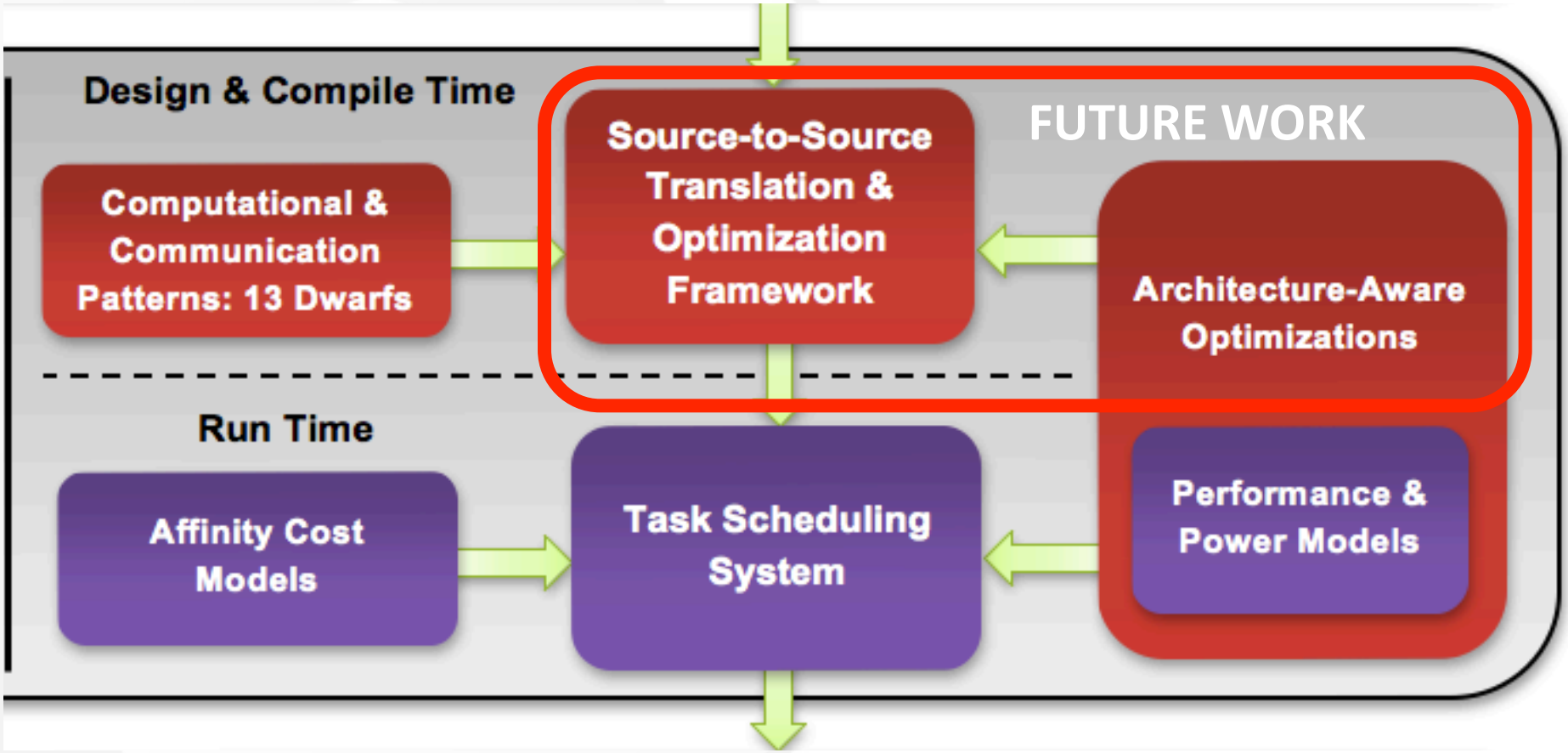
wfeng@vt.edu, 540.231.1192

Roadmap



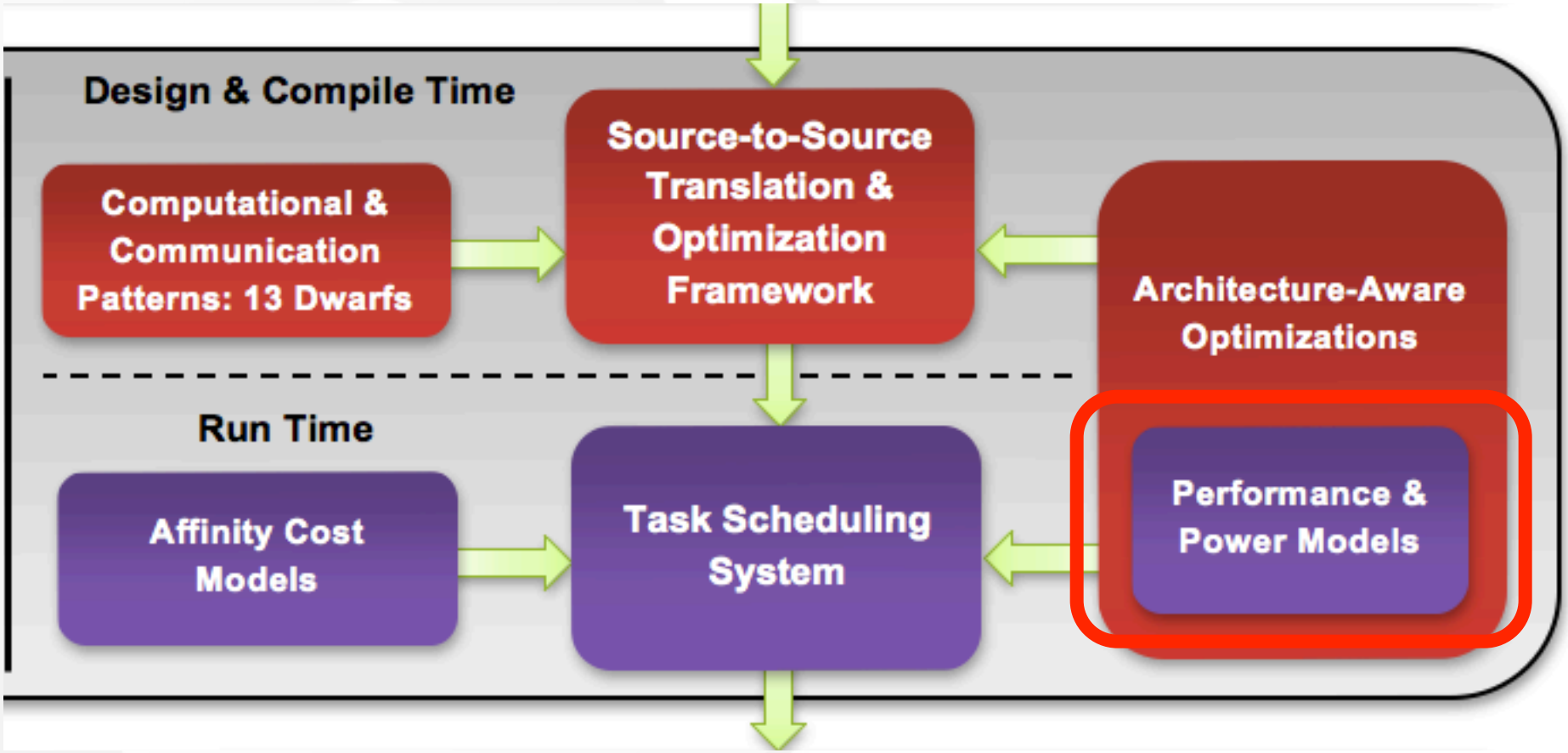
© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Roadmap



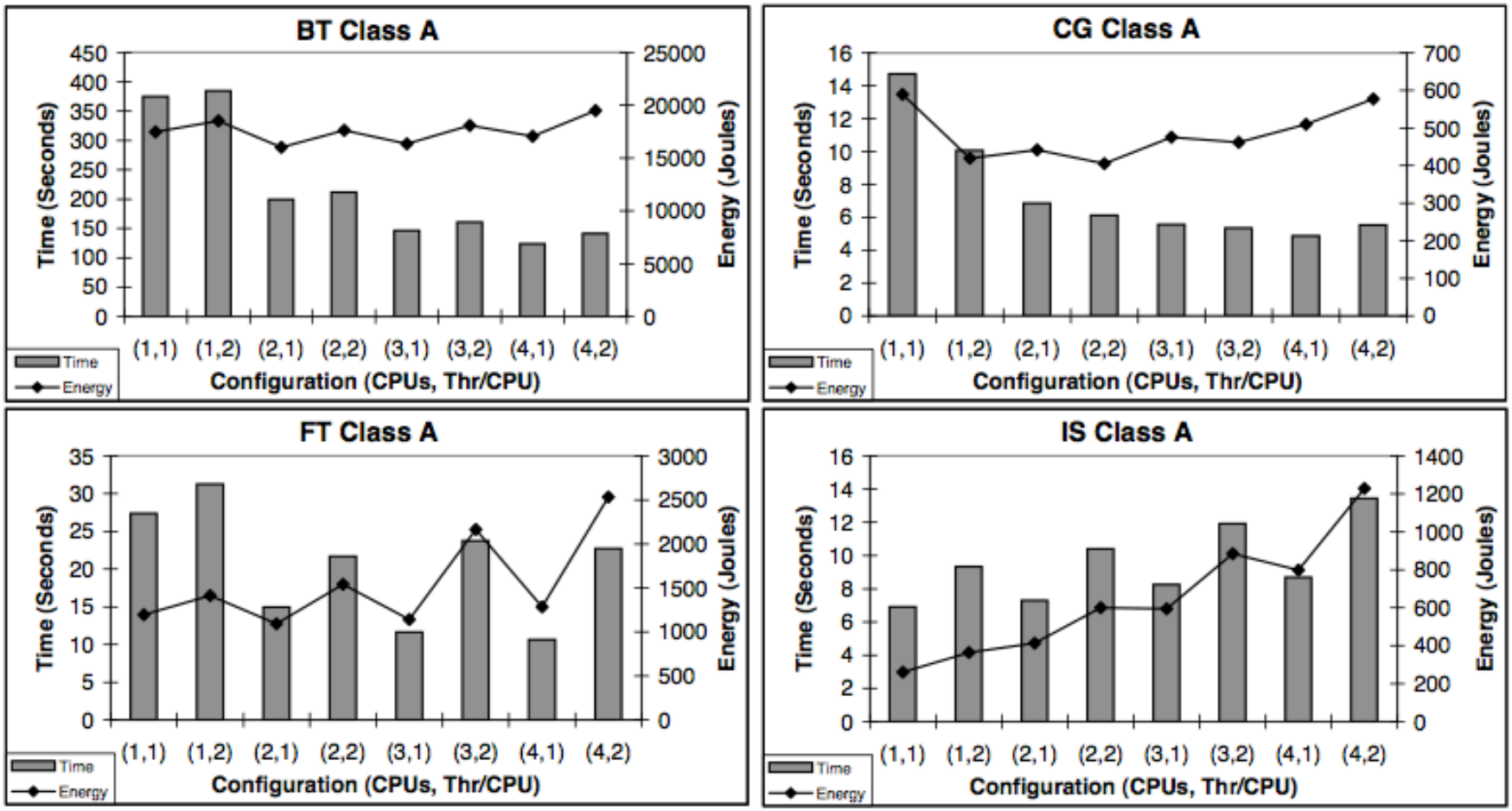
©W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Roadmap



© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Need for Performance & Power Modeling



Source: Virginia Tech

© W. Feng, May 2012

wfeng@vt.edu, 540.231.1192

Performance & Power Modeling

- Goals
  - Robust framework
  - Very high accuracy (Target: < 5% prediction error)
  - Identification of portable predictors for performance and power
  - Multi-dimensional characterization
    - Performance → sequential, intra-node parallel, inter-node parallel
    - Power → component level, node level, cluster level

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Problem Formulation:

LP-Based Energy-Optimal DVFS Schedule

- Definitions
  - A DVFS system exports n $\{ (f_i, P_i) \}$ settings.
  - T_i : total execution time of a program running at setting i
- Given a program with deadline D, find a DVS schedule (t_1*, …, t_n*) such that
  - If the program is executed for t_i seconds at setting i, the total energy usage E is minimized, the deadline D is met, and the required work is completed.

$$\min E = \sum_i P_i \cdot t_i$$

subject to

$$\sum_i t_i \leq D$$
$$\sum_i t_i / T_i = 1$$
$$t_i \geq 0$$

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Single-Coefficient β Performance Model

- Our Formulation
  - Define the relative performance slowdown δ as

    $T(f) / T(f_{MAX}) - 1$
  - Re-formulate two-coefficient model as a single-coefficient model:

$$\frac{T(f)}{T(f_{max})} = \beta \cdot \frac{f_{max}}{f} + (1 - \beta)$$

with

$$\beta = \frac{W_{cpu}}{W_{cpu} + T_{mem} \cdot f_{max}}$$

  - The coefficient β is computed at run-time using a regression method on the past MIPS rates reported from the built-in PMU.

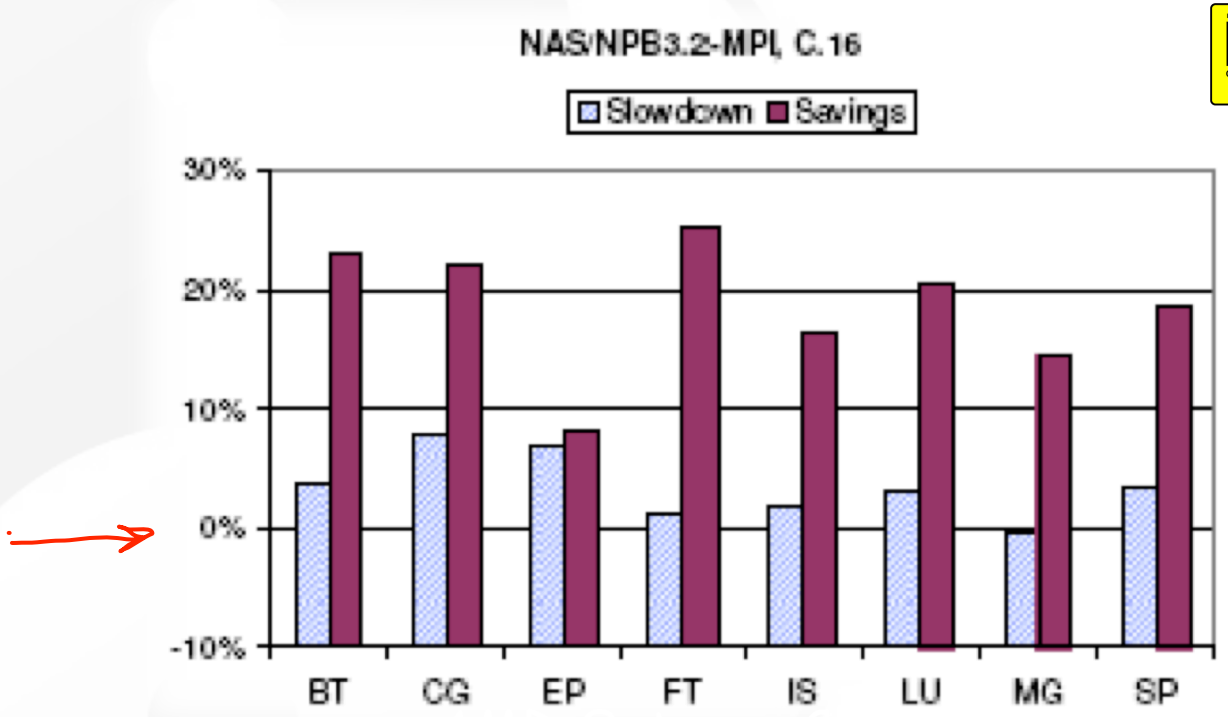
$$\beta = \frac{\sum_i (\frac{f_{max}}{f_i} - 1)(\frac{\mathtt{mips}(f_{max})}{\mathtt{mips}(f_i)} - 1)}{\sum_i (\frac{f_{max}}{f_i} - 1)^2}$$

C. Hsu and W. Feng. "A Power-Aware Run-Time System for High-Performance Computing," *SC|05,* Nov. 2005.

© W. Feng, May 2012

wfeng@vt.edu, 540.231.1192

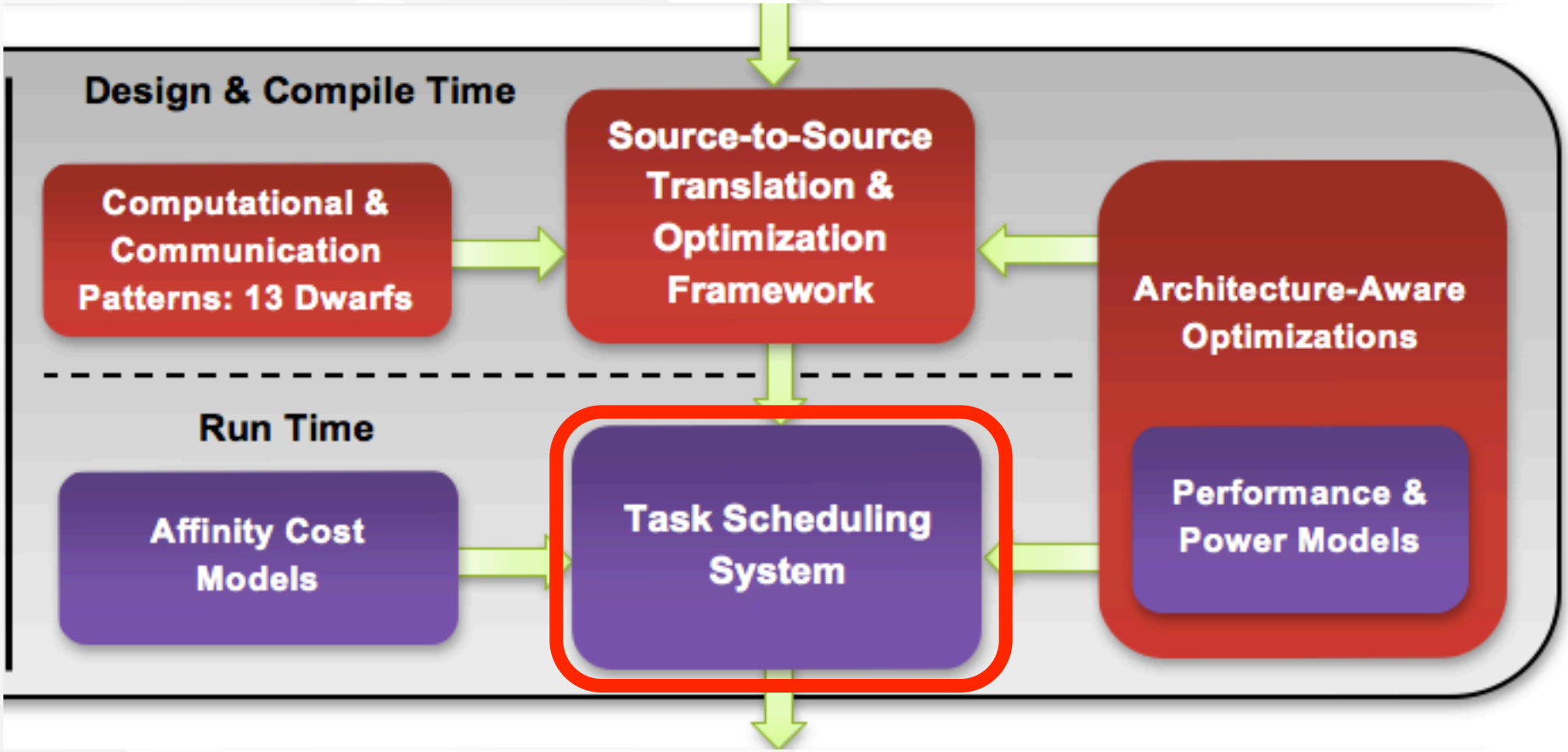
NAS Parallel on an AMD Opteron Cluster



"A Power-Aware Run-Time System for High-Performance Computing," *SC|05,* Nov. 2005

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Roadmap



©W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

What is Heterogeneous Task Scheduling?

- Automatically spreading tasks across heterogeneous compute resources
  - CPUs
  - GPUs
  - APUs
- Specify tasks at a higher level (currently OpenMP extensions)
- Run them across available resources automatically

Goal

- A run-time system that intelligently uses what is available resource-wise and optimize for performance portability
  - Each user should *not* have to implement this for themselves!

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

How to Heterogeneous Task Schedule (HTS)

- Accelerated OpenMP offer heterogeneous task scheduling with
  - Programmability
  - Functional portability, given underlying compilers
  - Performance portability
- How?
  - A simple extension to Accelerated OpenMP syntax for **programmability**
  - Automatically dividing parallel tasks across arbitrary heterogeneous compute resources for **functional portability**
    - CPUs
    - GPUs
    - APUs
  - Intelligent runtime task scheduling for **performance portability**

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Programmability: Why Accelerated OpenMP?

```
#pragma omp parallel for        \
        shared(in1,in2,out,pow)
for (i=0; i<end; i++){
    out[i] = in1[i]*in2[i];
    pow[i] = pow[i]*pow[i];
}
```

Traditional OpenMP

```
#pragma acc_region_loop                    \
        acc_copyin(in1[0:end],in2[0:end])\
        acc_copyout(out[0:end])            \
        acc_copy(pow[0:end])
for (i=0; i<end; i++){
    out[i] = in1[i] * in2[i];
    pow[i] = pow[i]*pow[i];
}
```

OpenMP Accelerator Directives

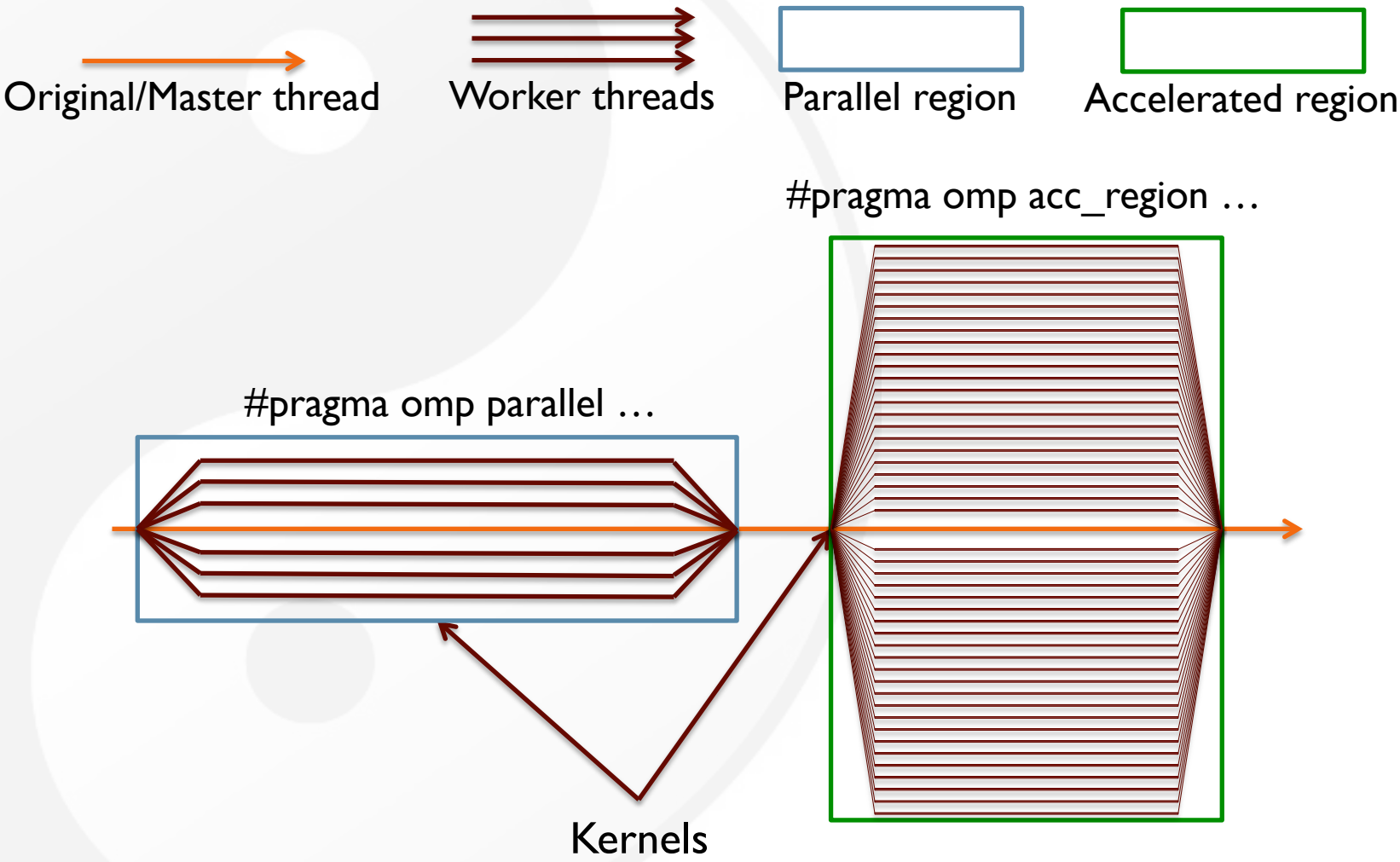
```
#pragma acc_region_loop                                  \
        acc_copyin(in1[0:end],in2[0:end])   \
        acc_copyout(out[0:end])                  \
        acc_copy(pow[0:end])                       \
        hetero(<cond>[,<scheduler>[,<ratio>\
                [,<div>]]])
for (i=0; i<end; i++){
    out[i] = in1[i] * in2[i];
    pow[i] = pow[i]*pow[i];
}
```

Source: T. Scogland et al., IPDPS 2012

Our Proposed Extension

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

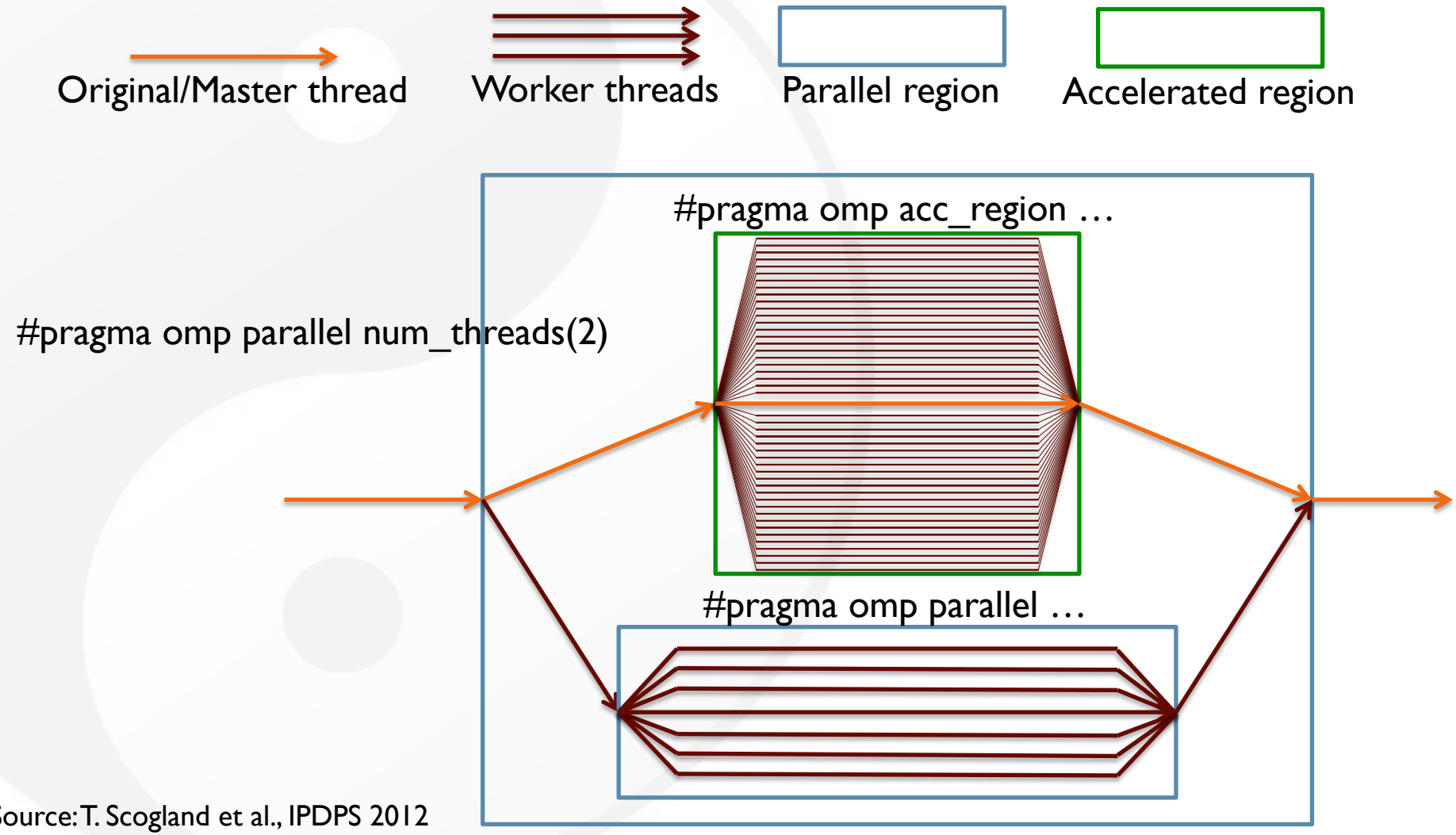
OpenMP Accelerator Behavior



Source: T. Scogland et al., IPDPS 2012

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

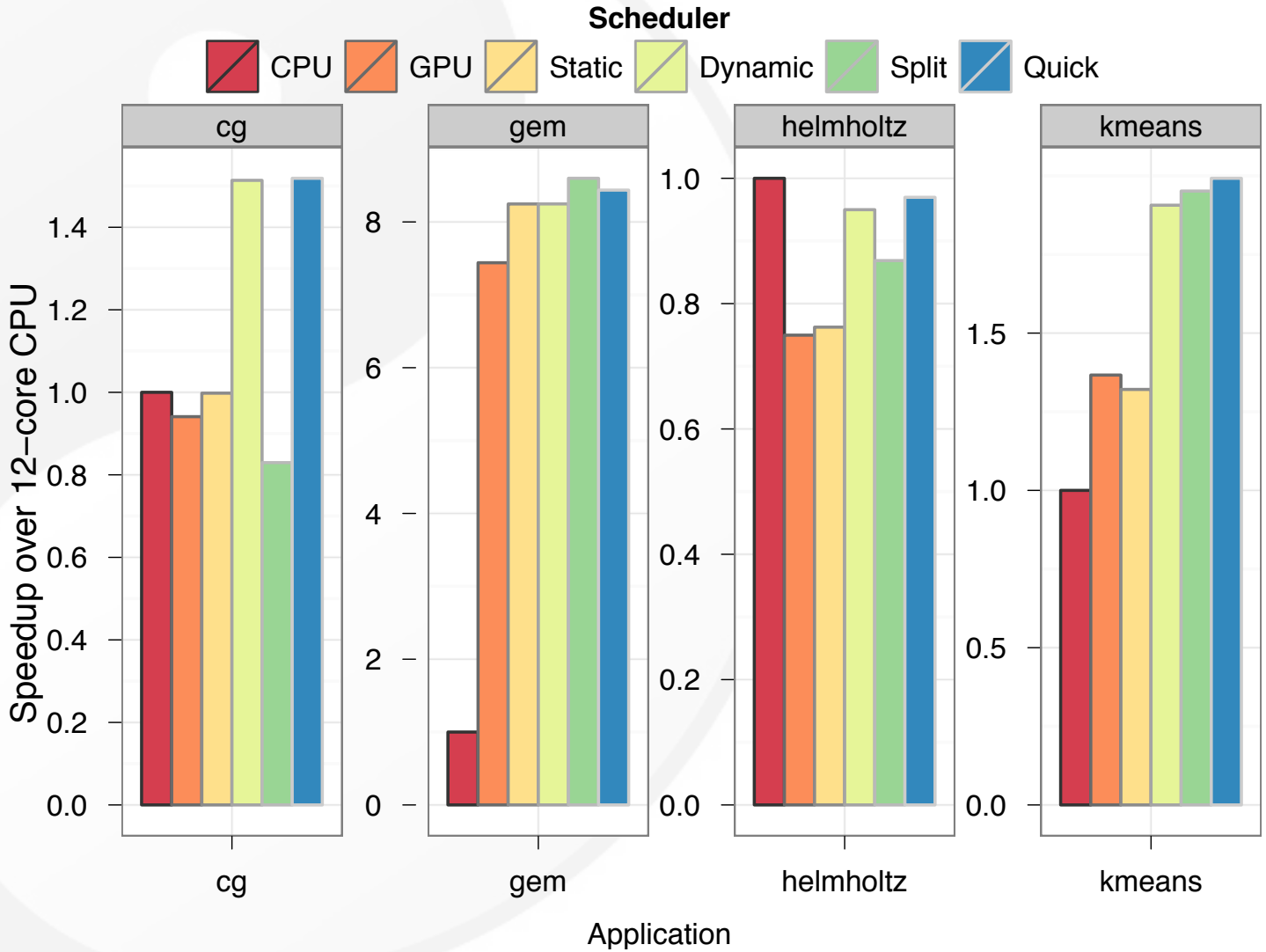
DESIRED OpenMP Accelerator Behavior



Source: T. Scogland et al., IPDPS 2012

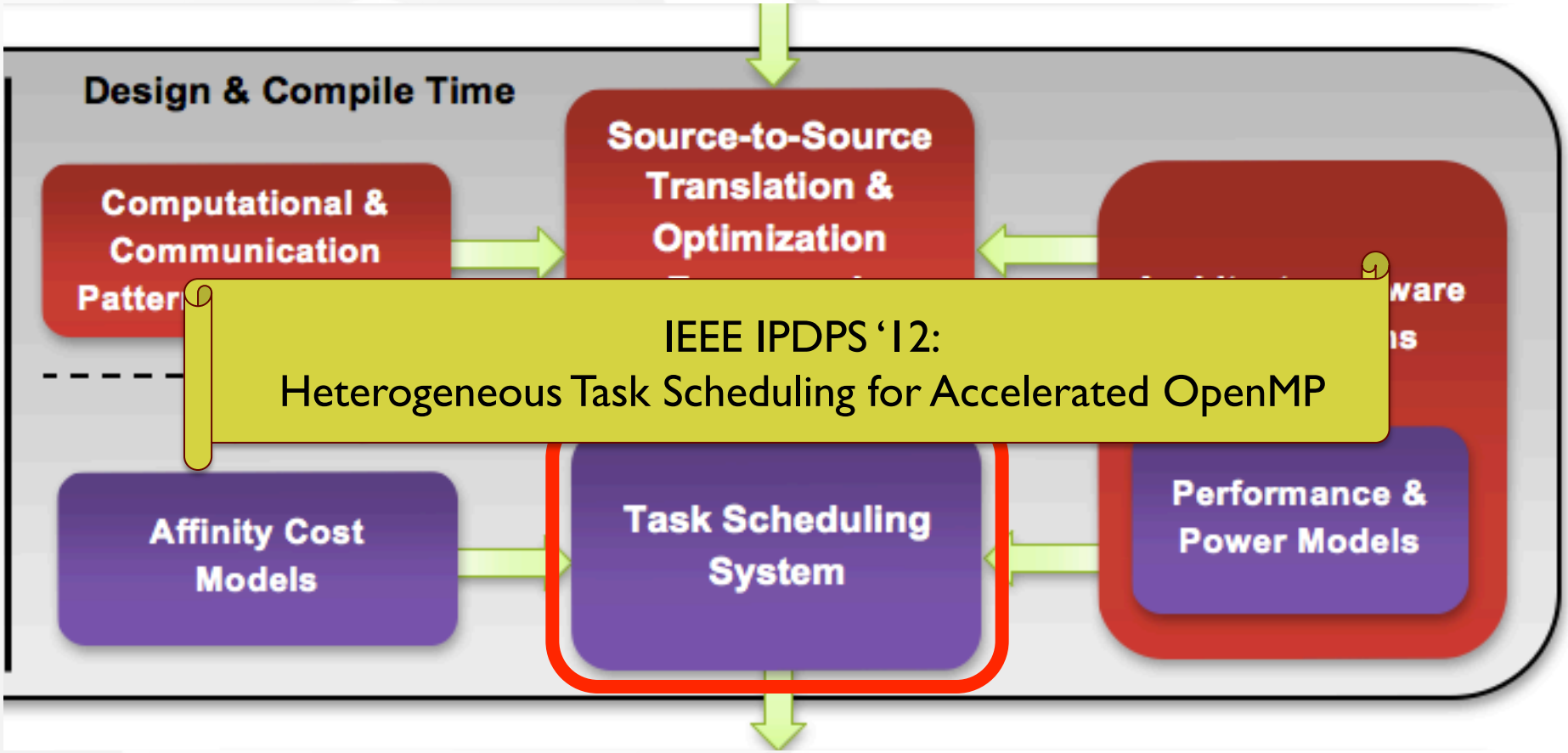
© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Results Across Schedulers for all Benchmarks



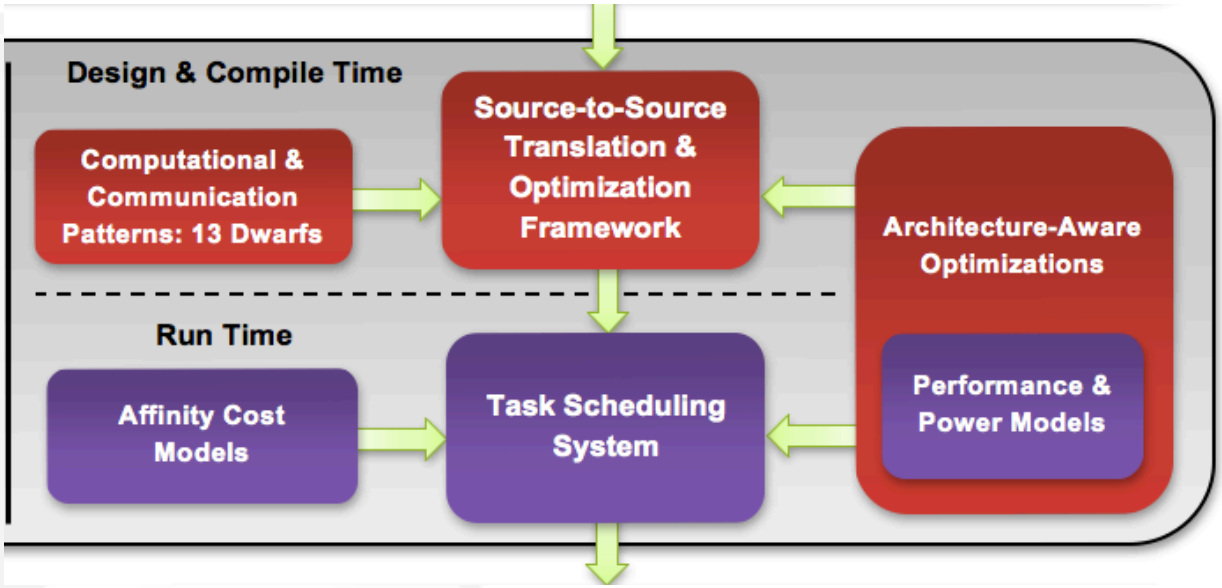
© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Roadmap



© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

An Ecosystem for the New HPC: Heterogeneous Parallel Computing



Much work still to be done.

• OpenCL and the 13 Dwarfs	Beta release pending
• Source-to-Source Translation	CU2CL only & no optimization
• Architecture-Aware Optimization	Only manual optimizations
• Performance & Power Modeling	Preliminary & pre-multicore
• Affinity-Based Cost Modeling	Empirical results; modeling in progress
• Heterogeneous Task Scheduling	Preliminary with OpenMP

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Recent Publications

- M. Daga, T. Scogland, W. Feng, "Architecture-Aware Mapping and Optimization on a 1600-Core GPU," *17th IEEE Int'l Conf. on Parallel & Distributed Systems*, Dec. 2011.
- M. Elteir, H. Lin, W. Feng, "StreamMR: An Optimized MapReduce Framework for AMD GPUs," *17th IEEE Int'l Conf. on Parallel & Distributed Systems*, Dec. 2011.
- W. Feng, Y. Cao, D. Patnaik, N. Ramakrishnan, "Temporal Data Mining for Neuroscience," *GPU Computing Gems,* Editor: W. Hwu, Elsevier/Morgan-Kaufmann, Feb. 2011.
- K. Bisset, A. Aji, M. Marathe, W. Feng, "High-Performance Biocomputing for Simulating the Spread of Contagion over Large Contact Networks," *BMC Genomics*, 2011.
- M. Elteir, H. Lin, W. Feng, "Performance Characterization and Optimization of Atomic Operations on AMD GPUs," *IEEE Cluster*, Sept. 2011.
- M. Daga, A. Aji, W. Feng, "On the Efficacy of a Fused CPU+GPU Processor for Parallel Computing," *Symp. on Application Accelerators in High Performance Computing*, Jul. 2011.
- A. Aji, M. Daga, and W. Feng, "Bounding the Effect of Partition Camping in Memory-Bound Kernels," *ACM Int'l Conf. on Computing Frontiers*, May 2011.
- S. Xiao, H. Lin, and W. Feng, "Accelerating Protein Sequence Search in a Heterogeneous Computing System," *25th Int'l Parallel & Distributed Processing Symp.*, May 2011.
- W. Feng with cast of many, "Accelerating Electrostatic Surface Potential Calculation with Multi-Scale Approximation on Graphics Processing Units," *J. Molecular Graphics and Modeling*, Jun. 2010.
- W. Feng and S. Xiao, "To GPU Synchronize or Not GPU Synchronize?" *IEEE Int'l Symp. on Circuits and Systems*, May-June 2010.

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Funding Acknowledgements

© W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

Wu Feng, wfeng@vt.edu, 540-231-1192

SyNeRG

http://synergy.cs.vt.edu/

CHREC
NSF Center for High-Performance Reconfigurable Computing

http://www.chrec.org/

mpiBLAST

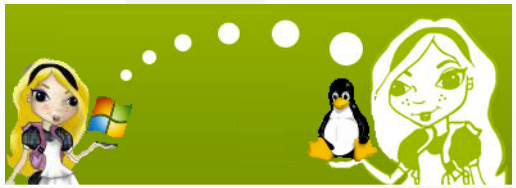
http://www.mpiblast.org/

SUPERCOMPUTING in SMALL SPACES

http://sss.cs.vt.edu/

THE GREEN 500

http://www.green500.org/



http://myvice.cs.vt.edu/

"Accelerators 'R Us"

http://accel.cs.vt.edu/

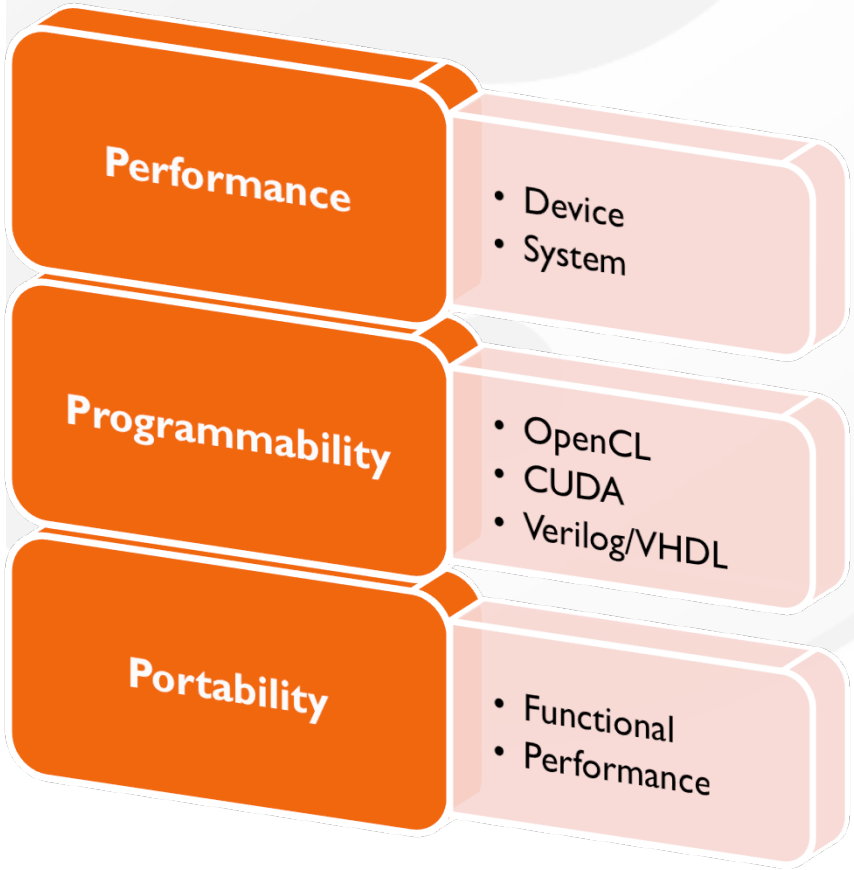
VirginiaTech
Invent the Future

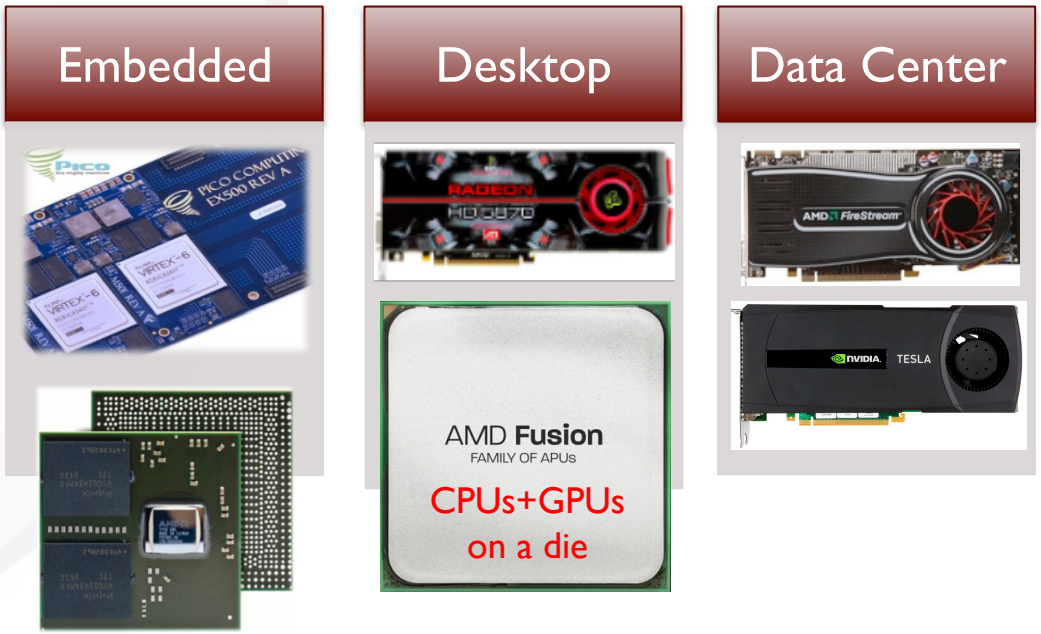
SyNeRG
synergy.cs.vt.edu

An Ecosystem for the New HPC: Heterogeneous Parallel Computing

- Deliver personalized supercomputing to the masses
  - Heterogeneity of hardware devices *plus* enabling software that tunes for *performance* (speed & power), *programmability, and portability*

via a benchmark suite of computational dwarfs and apps





©W. Feng, May 2012
wfeng@vt.edu, 540.231.1192

SyNeRG
synergy.cs.vt.edu