On the Path to Exascale: Deploying an Emerging HPC Architecture

Jeffrey Vetter

DOE Exascale Tools Workshop
Annapolis, MD
13 October 2011

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

In a nutshell

- Recap motivation for emerging architectures
- Productivity
- Keeneland – a case study
  - Software plays a critical roll in productivity!
- Top 10 Gaps in our Existing Toolset (wearing my project director hat)
  - Find a way to measure and demonstrate productivity and you will be well-rewarded

Technology Trends

Notional Exascale Architecture Targets

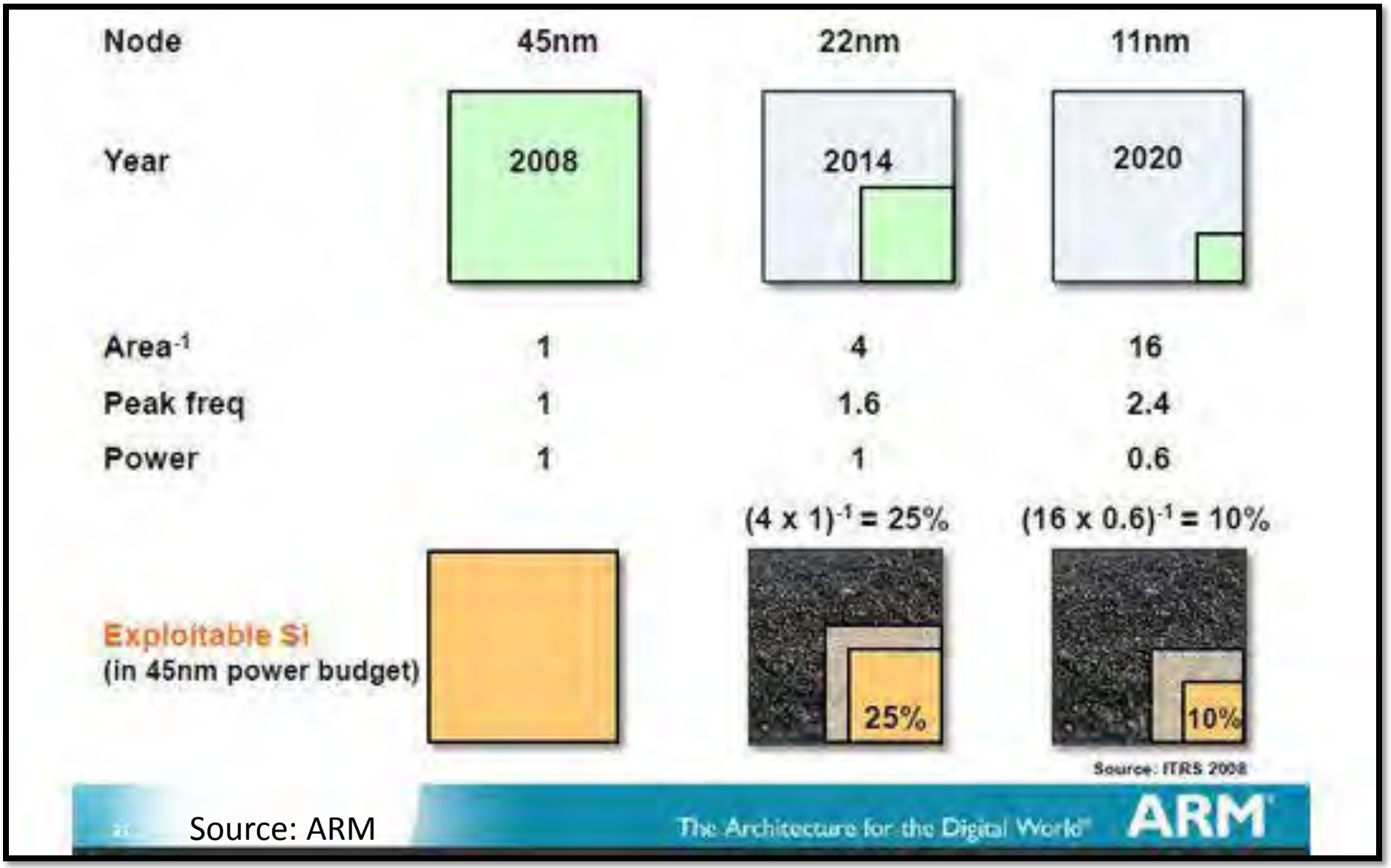
Exascale Arch Report 2009

System attributes	2001	2010	"2015"		"2018"	
System peak	10 Tera	2 Peta	200 Petaflop/sec		1 Exaflop/sec	
Power	~0.8 MW	6 MW	15 MW		20 MW	
System memory	0.006 PB	0.3 PB	5 PB		32-64 PB	
Node performance	0.024 TF	0.125 TF	0.5 TF	7 TF	1 TF	10 TF
Node memory BW		25 GB/s	0.1 TB/sec	1 TB/sec	0.4 TB/sec	4 TB/sec
Node concurrency	16	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	416	18,700	50,000	5,000	1,000,000	100,000
Total Node Interconnect BW		1.5 GB/s	150 GB/sec	1 TB/sec	250 GB/sec	2 TB/sec
MTTI		day	O(1 day)		O(1 day)	

Contemporary Systems

Date	System	Location	Comp	Comm	Peak (PF)	Power (MW)
2010	Tianhe-1A	NSC in Tianjin	Intel + NVIDIA	Proprietary	4.7	4.0
2010	Nebulae	NSC In Shenzhen	Intel + NVIDIA	IB	2.9	2.6
2010	Tsubame 2	TiTech	Intel + NVIDIA	IB	2.4	1.4
2011	K Computer (612 cabinets)	Kobe	SPARC64 VIIIfx	Tofu	8.7	9.8
~2012	Cray 'Titan'	ORNL	AMD + NVIDIA	Gemini	20?	7?
~2012	BlueGeneQ	ANL	SoC	IBM	10?	?
~2012	BlueGeneQ	LLNL	SoC	IBM	20?	?
~2012	BlueWaters Redux ??	NCSA	??	??	??	??
	Others…					

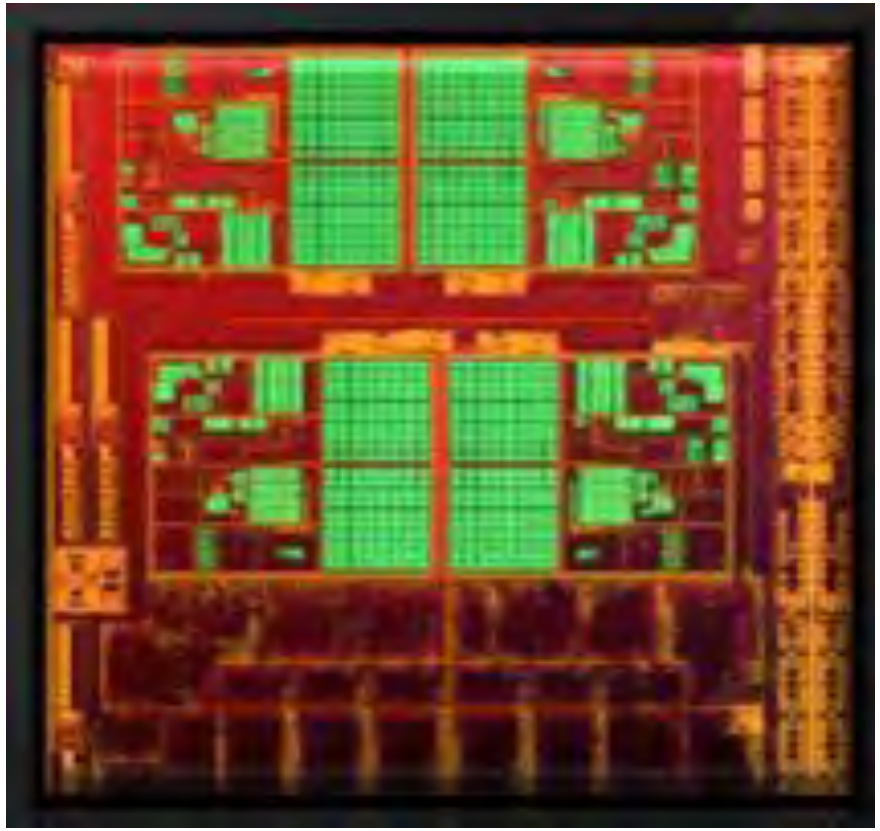
The Commodity Trend: Dark Silicon



Source: ARM

AMD's Llano: A-Series APU

- Combines
  - 4 x86 cores
  - Array of Radeon cores
  - Multimedia accelerators
  - Dual channel DDR3
- 32nm
- Up to 29 GB/s memory bandwidth
- Up to 500 Gflops SP
- 45W TDP



Source: AMD

Tianhe-1A uses 7000+ NVIDIA GPUs

- Tianhe-1A uses
  - 7,168 NVIDIA Tesla M2050 GPUs
  - 14,336 Intel Westmeres
- Performance
  - 4.7 PF peak
  - 2.5 PF sustained on HPL
- 4.04 MW
  - If Tesla GPU's were not used in the system, the whole machine could have needed 12 megawatts of energy to run with the same performance, which is equivalent to 5000 homes
- Custom fat-tree interconnect
  - 2x bandwidth of Infiniband QDR

The New York Times

Business Day
Technology

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

Search Technology | Go | Inside Technology | Internet | Start-Ups | Business Computing | Companies | Bits | Blog

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, and giving China bragging rights as a technology superpower.

RECOMMEND
TWITTER
SIGN IN TO E-MAIL
PRINT
REPRINTS
SHARE

Enlarge This Image

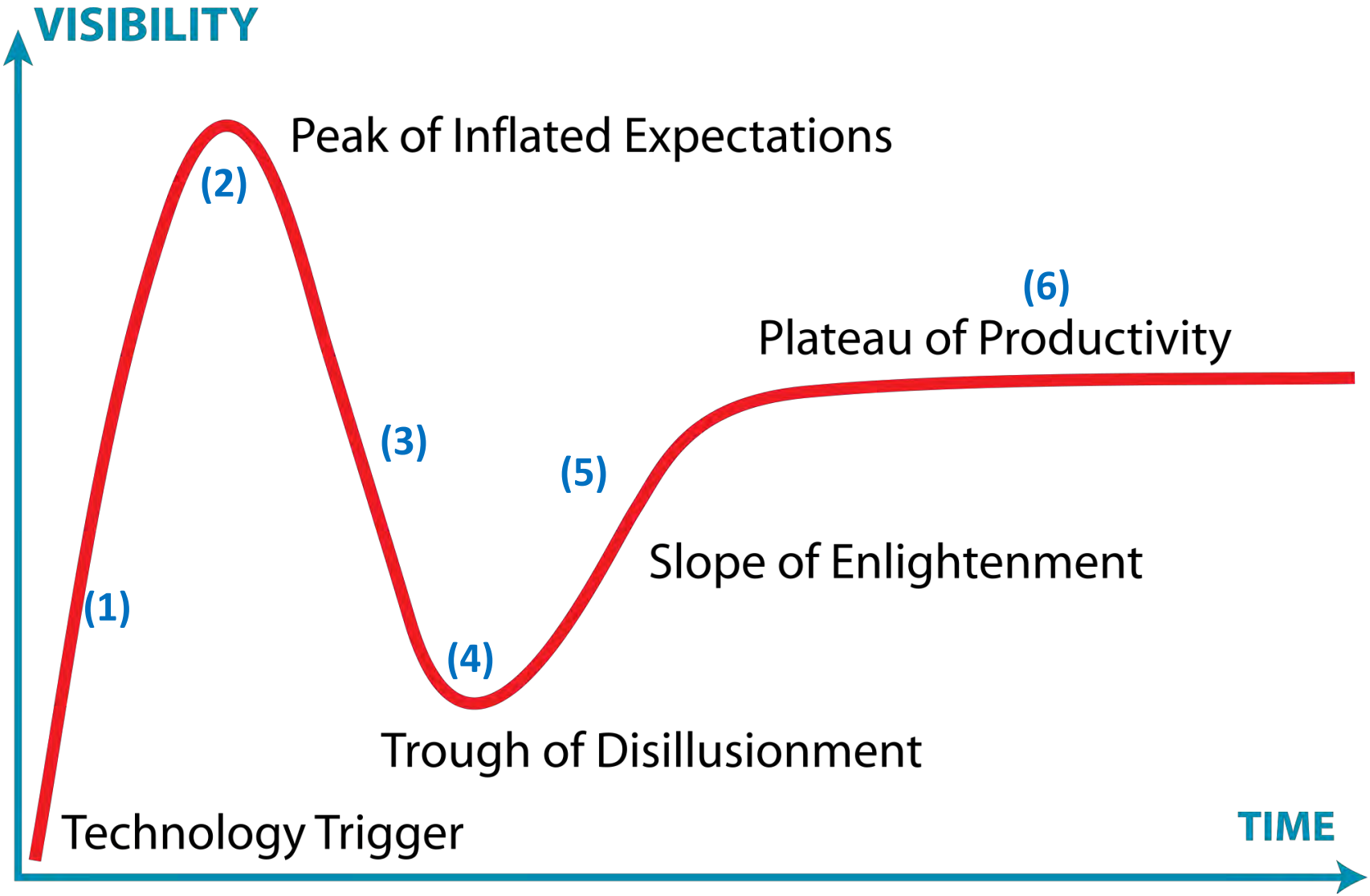
Nvidia

The Tianhe-1A computer in Tianjin, China, links thousands upon thousands of chips.

The computer, known as Tianhe-1A, has 1.4 times the horsepower of the current top computer, which is at a national laboratory in Tennessee, as measured by the standard test used to gauge how well the systems handle mathematical calculations, said Jack Dongarra, a University of Tennessee computer scientist who maintains the official supercomputer rankings.

Although the official list of the top 500 fastest machines, which comes out every six months, is not due to be completed by Mr. Dongarra until next week, he said the Chinese computer "blows away the existing No. 1 machine." He added, "We don't close the books until Nov. 1, but I would say it is unlikely we will see a system that is faster."

Quiz


VISIBILITY
Peak of Inflated Expectations
(2)
(6)
Plateau of Productivity
(3)
(5)
Slope of Enlightenment
(1)
(4)
Trough of Disillusionment
Technology Trigger
TIME

Productivity

“I know it when I see it”

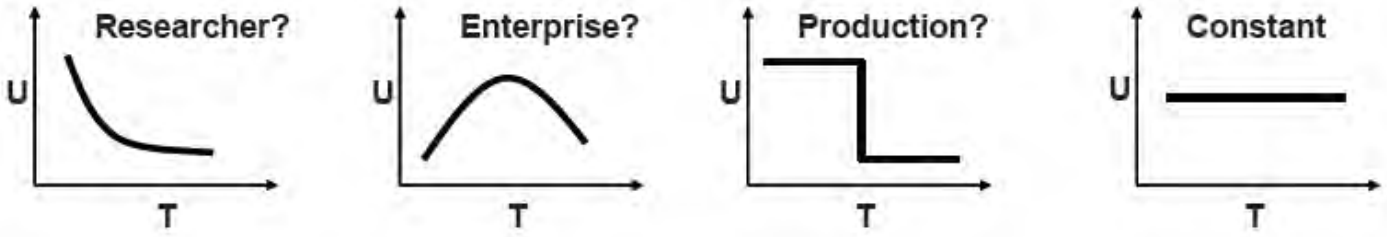
General Productivity Model

$$\Psi \equiv \frac{U}{C} = \frac{U(T)}{C_S + C_O + C_M}$$

ψ = productivity [utility/$]
U = utility [user specified]
T = time to solution [time]
C = total cost [$]

C_S = software cost [$]
C_O = operation cost [$]
C_M = machine cost [$]
$C_S+C_O+C_M = (c_S+c_O+c_M) \times T$

- Utility is value user places on getting a result at time T



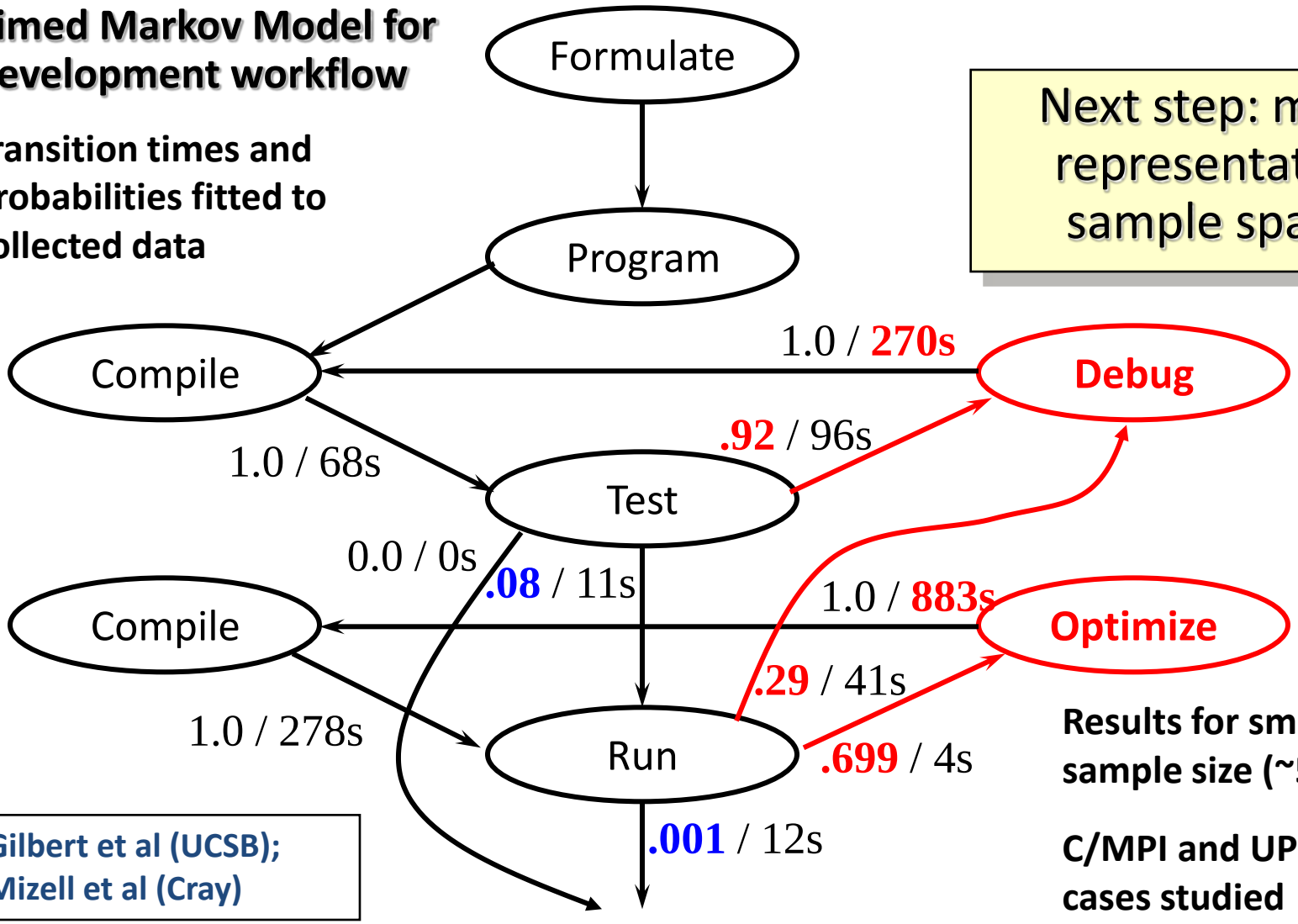
- T = T(P,Q) and C = C(P,Q) are functions system parameters P and application characteristics Q

Key challenges:
-Productivity is user/application specific
-Limited number of variables we can actually measure

Tracking development productivity

Timed Markov Model for development workflow

Transition times and probabilities fitted to collected data



Next step: more representative sample space

Results for small sample size (~5)

C/MPI and UPC cases studied

Gilbert et al (UCSB); Mizell et al (Cray)

Keeneland

A case study

Keeneland - Enabling Heterogeneous Computing for the Open Science Community

Jeffrey Vetter, Dick Glassbrook, Jack Dongarra, Karsten Schwan, Sudha Yalamanchili, Bruce Loftis, Stephen McNally, Jeremy Meredith, Patti Reed, Jim Rogers, Philip Roth, Kyle Spafford, Kevin Sharkey, and many others

http://keeneland.gatech.edu

Georgia Tech NICS THE UNIVERSITY of TENNESSEE OAK RIDGE National Laboratory NVIDIA hp

J.S. Vetter, R. Glassbrook, J. Dongarra, K. Schwan, B. Loftis, S. McNally, J. Meredith, J. Rogers, P. Roth, K. Spafford, and S. Yalamanchili, "Keeneland: Bringing heterogeneous GPU computing to the computational science community," *IEEE Computing in Science and Engineering, 13(5):90-5, 2011,* http://dx.doi.org/10.1109/MCSE.2011.83.

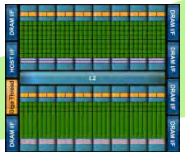
Keeneland High Level Goals (in one slide)

- Provide a new, innovative class of computing architecture to the NSF community for science
- Acquire, deploy, and operate two GPU clusters
  - Initial delivery - Operational
  - Full scale – Spring 2012
  - Operations, user support
- Ensure software tools, application development support for user productivity and success
- Perform technology assessment to track fast moving hardware and software
- Perform education, Outreach, Training for scientists, students, industry on these new architectures

Keeneland – Initial Delivery System Architecture

- **Initial Delivery** system procured and installed in Oct 2010
- 201 TFLOPS in 7 racks (90 sq ft incl service area)
- 677 MFLOPS per watt on HPL
- Final delivery system expected in early 2012

67 GFLOPS



515 GFLOPS

1679 GFLOPS
24/18 GB

Mellanox ConnectX-2

Full PCIe X16 bandwidth to all GPUs

6718 GFLOPS

40306 GFLOPS

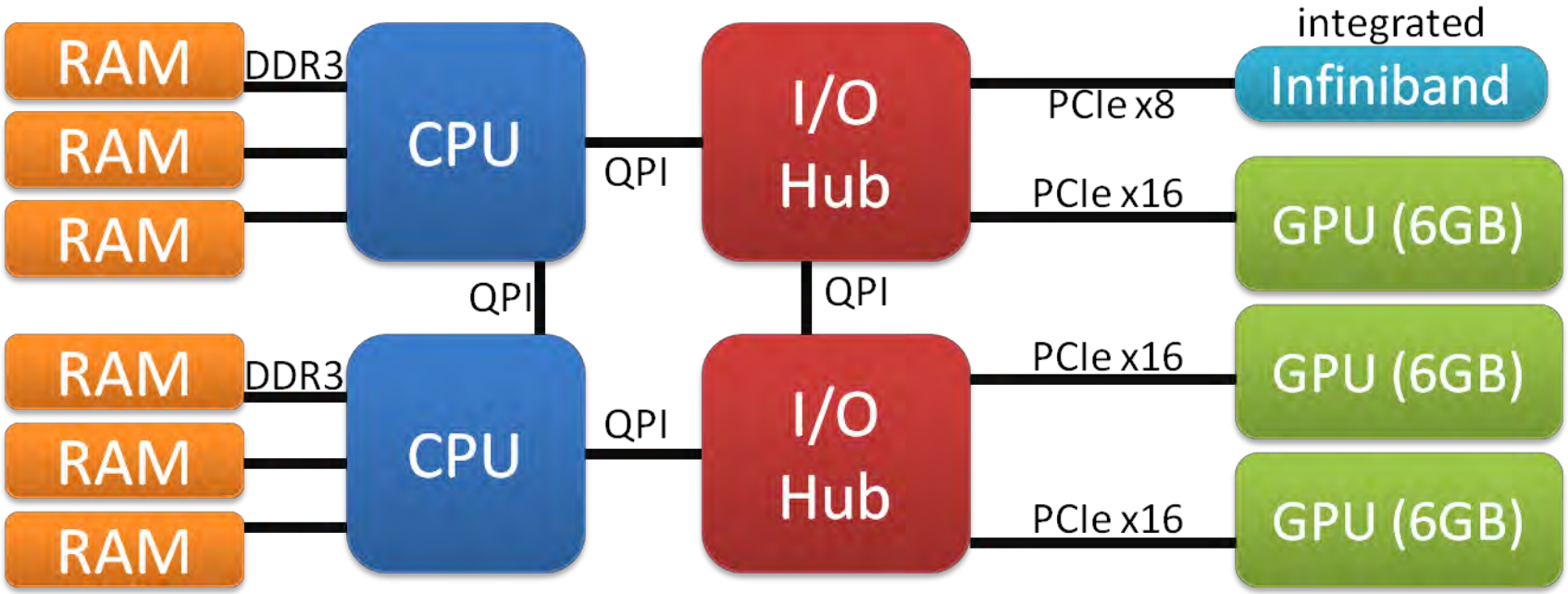
201528 GFLOPS

QLOGIC **12000-Series Director Switch**

Integrated with NICS Datacenter GPFS and TG

Georgia Tech · NICS · The University of Tennessee · Oak Ridge National Laboratory · NVIDIA · hp · NSF

Keeneland Node Architecture SL390

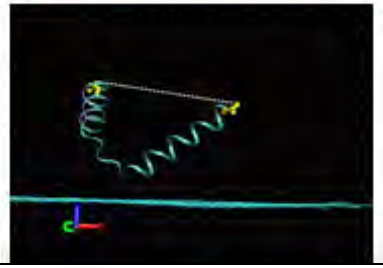


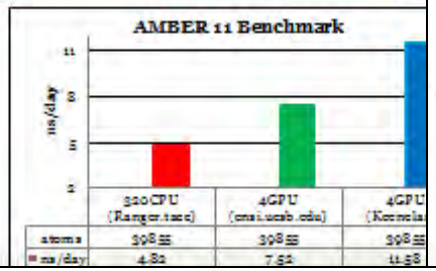
Peptide folding on surfaces

Joan-Emma Shea at

- Peptide folding on a hydrophobic surface
  - www.chem.ucsb.edu/~sheagroup
- Surfaces can modulate the folding and aggregation pathways of proteins. Here, we investigate the folding of a small helical peptide in the presence of a hydrophobic surface of graphite. Simulations are performed using explicit solvent and a fully atomic representation of the peptide and the surface.

- Benefits of running on a GPU cluster:
  - Reduction in the the number of computing nodes needed: one GPU is at least faster than 8 CPUs in GPU-accelerated AMBER Molecular Dynamics.
  - The large simulations that we are currently running would be prohibitive using CPUs. The efficiency of the CPU parallelization becomes poorer with increasing number of CPUs.
  - It can also decrease consumption of memory and network bandwidth in simulations with large number of atoms.

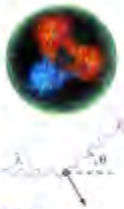




Hadron Polarizability in Lattice QCD

Andrei Alexandru
The George Washington University

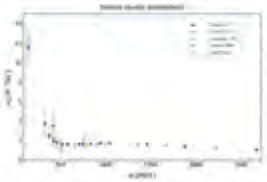
Understanding the structure of subnuclear particles represents the main challenge for today's nuclear physics. Photons are used to probe this structure in experiments carried out at laboratories around the world. To interpret the results of these experiments we need to understand how electromagnetic field interacts with subnuclear particles. Theoretically, the structure of subnuclear particles is described by Quantum Chromodynamics (QCD). Lattice QCD is a 4-dimensional discretized version of this theory that can be solved numerically. The focus of our project is to understand how the electric field deforms neutrons and protons by computing the polarizability using lattice QCD techniques.

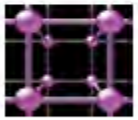


http://lamumbophys.gwu.edu/wiki/index.php/Hadron_polarizability

Why GPUs?

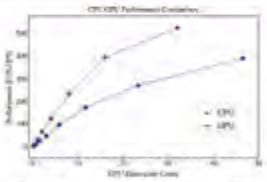
- Lattice QCD simulations require very large bandwidth to run efficiently. GPUs have 10-15 times larger memory bandwidth compared to CPUs.
- Lattice QCD simulations can be efficiently parallelized
  - Bulk of calculation spent on one kernel.
  - The kernel requires only nearest neighbor information.
  - Cut the lattice into equal sub-lattices. Effectively use single instruction multiple-data (SIMD) paradigm.





Experimental and current values for neutron electric polarizability in lattice QCD.

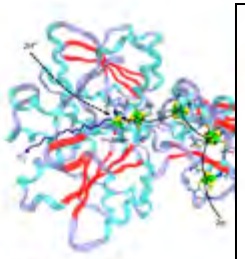
Alexandru and F. X. Lee (arXiv:0810.2833)



Performance comparison between Keeneland's GPU cluster and Kraken's Cray XT-5 machine. The CPU core count is translated to GPU equivalent count by dividing the total number of CPUs by 32, which is the number of CPU cores equivalent to a single-GPU performance.

A. Alexandru, et. al. (arXiv:1103.5103)

NAMD



- Biomolecular dynamics
- Public 2.7 Beta 4 pre-release
  - configuration flags to enable GPU support
  - minor benchmark input file modifications
- Single-node performance:
  - 4x faster than CPU on small benchmark, 9x faster on large

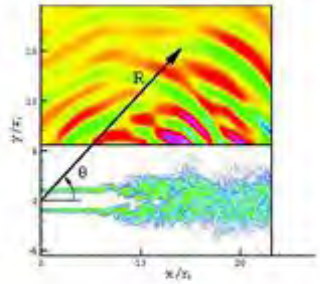
Benchmark	CPU Runtime	GPU Runti
dhfr	41.1 sec	11.5 s
apoa1	580.6 sec	67.56

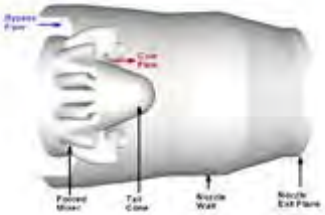
Georgia Tech NICS University of Tennessee Oak Ridge NVIDIA hp

Jet Engine Noise

User: Gregory Blaisdell, Purdue

- Large eddy simulation, computational aeroacoustics
- NSF PetaApps project
- Applications team:
  - function offload model results in communication bottlenecks
  - interested in pursuing other approaches to minimize communications

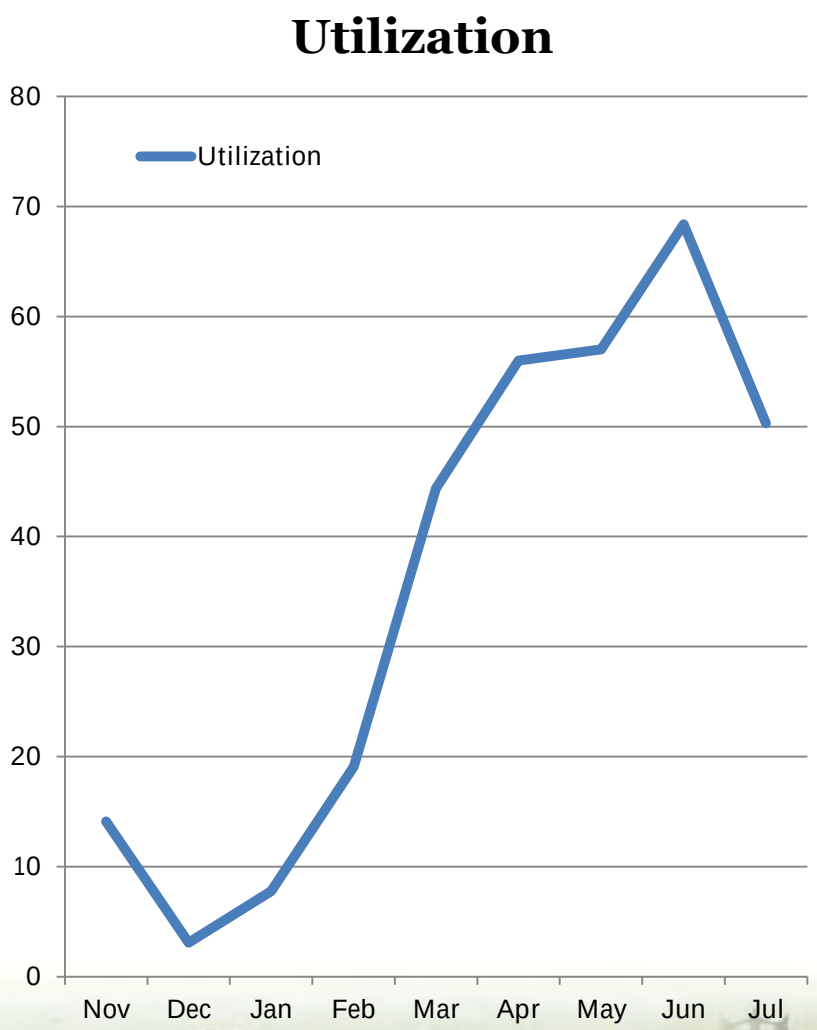




Georgia Tech NICS University of Tennessee Oak Ridge NVIDIA hp

Keeneland Usage At a Glance

- 110 nodes available to users with 8 reserved for software evaluation and 2 reserved for operational testing, totaling 120 compute nodes.
- 2 general purpose login nodes and 2 GSISSH login nodes, totaling 4 login nodes
- 190 total users with 130 active users
- 33,400 total jobs run since installation
- 28,500 completed jobs (due to system errors, user cancellations, or errors in codes)
- 85% of submitted jobs complete as expected



Software!

It’s the software stupid!

Holistic View of HPC

Performance, Resilience, Power, Programmability

Applications

- Materials
- Climate
- Fusion
- National Security
- Combustion
- Nuclear Energy
- Cybersecurity
- Biology
- High Energy Physics
- Energy Storage
- Photovoltaics
- National Competitiveness

- Usage Scenarios
  - Ensembles
  - UQ
  - Visualization
  - Analytics

Programming Environment

- Domain specific
  - Libraries
  - Frameworks
  - Templates
  - Domain specific languages
  - Patterns
  - Autotuners
- Platform specific
  - Languages
  - Compilers
  - Interpreters/Scripting
  - Performance and Correctness Tools
  - Source code control

Critical

System Software

- Resource Allocation
- Scheduling
- Security
- Communication
- Synchronization
- Filesystems
- Instrumentation
- Virtualization

Architectures

- Processors
  - Multicore
  - Graphics Processors
  - FPGA
  - DSP
- Memory and Storage
  - Shared (cc, scratchpad)
  - Distributed
  - RAM
  - Storage Class Memory
  - Disk
  - Archival
- Interconnects
  - Infiniband
  - IBM Torrent
  - Cray Gemini, Aires
  - BGL/P/Q
  - 1/10/100 GigE

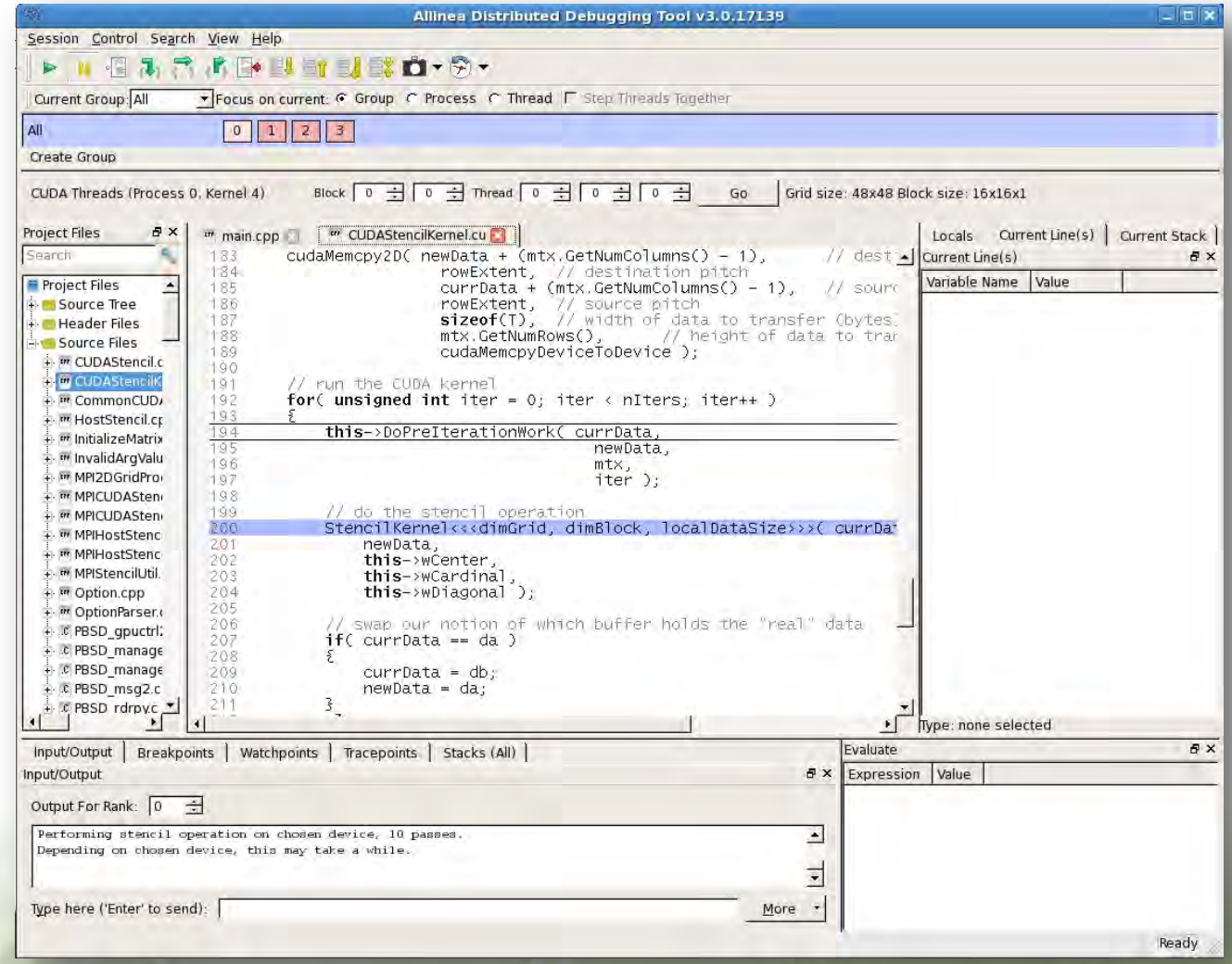
KIDS Software

- **CUDA 3.2, 4.0**
- **NVIDIA OpenCL 1.0**
- CentOS 5.5
- Intel, PGI, and GNU compilers
- OpenMPI, MVAPICH
- Torque (PBS) batch software

- GPU Enabled Compilers
  - PGI 11.x
  - CAPS HMPP Workbench 2.4.4
  - Reservoir Labs R-Stream 3.1.4.1
  - OpenMPC
  - PGI CUDA FORTRAN
  - Participating in OpenMP Accelerator working group
- Debuggers
  - Cuda-gdb
  - Allinea DDT

Allinea DDT with CUDA support

- Debugging Stencil2D from SHOC benchmark suite

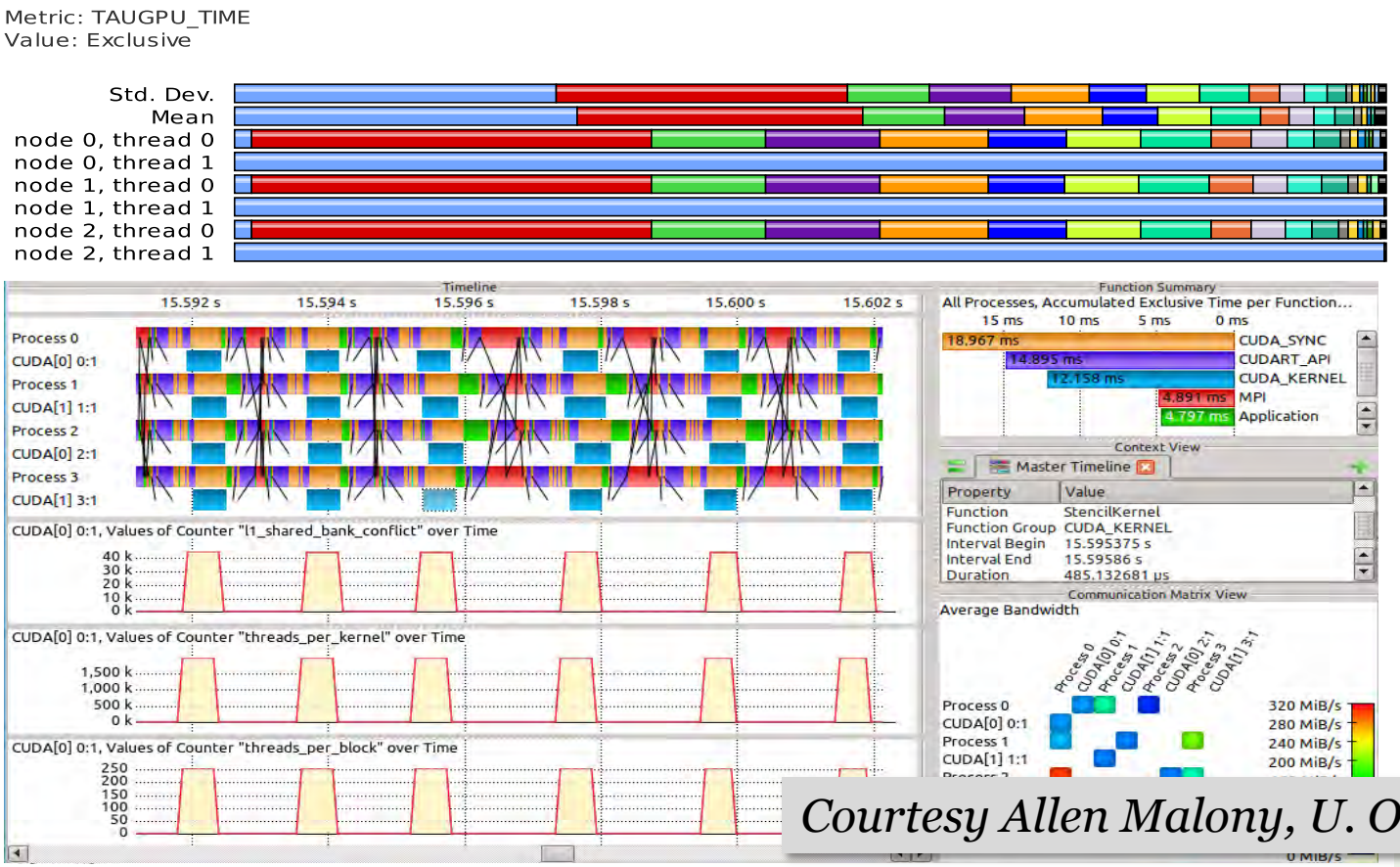


KIDS Software (2)

- Libraries
  - CUBLAS
  - CUFFT
  - CUSparse
  - CURNG
  - MAGMA
  - Jacket (eval)
- Frameworks
  - Thrust
- Performance
  - NVIDIA Visual Profiler
  - Tau
- Hardware counters
  - CUPTI
  - PAPI (on CUPTI)
- Other language bindings…

DOE Vancouver TAU Example

Stencil2D Parallel Profile / Trace



Courtesy Allen Malony, U. Oregon

Challenges

Top 10 Gaps in our Existing Toolset (wearing my facility project director hat)

1. *Anticipate the software availability time lag*

2. Systematic performance modeling to project future performance
   - Is my application a good fit for a GPU?
3. Locality management, data orchestration tools
   - Which memory should I use for my kernel?
   - NUMA, NUDA effects
4. Interoperability among multiple programming systems and tools
   - MPI, OpenMP, Threads, CUDA, OpenCL, PGAS,
   - E.g., GPUdirect
5. Self-aware runtime systems that help manage load-balance and device management (to hide 'system' differences), autotuning
   - Most every GPU system has a (drastically) different configuration

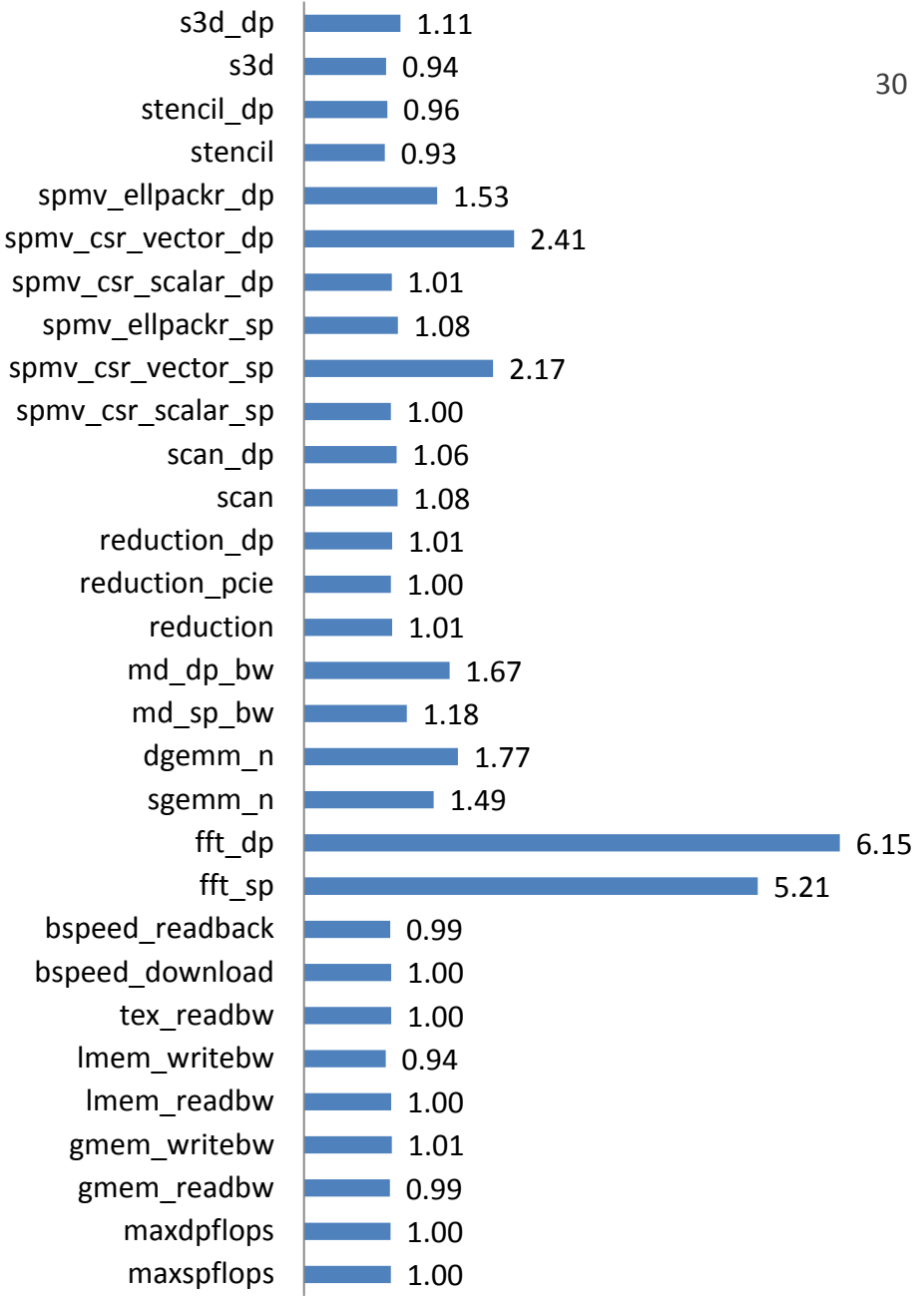
Top 10 Gaps in our Existing Toolset [2] (wearing my facility project director hat)

6. Correctness tools that automatically identify problems
   - Ocelot records memory conflicts, access errors
7. Resiliency and power information
   - Some of our users want to trade ECC for 50% performance improvement
8. Fine grained thread information
   - Most tools provide high level information about kernel performance (attribution)
9. Education, education, education!!
10. Performance contracts
11. Unified programming model
    - Many problem could be solved with a higher level programming model

Some possible solutions

CUDA v. OpenCL

- What does performance look like today?
- This chart shows the SHOC results of CUDA over OpenCL on a single Tesla M2070 on KIDS (CUDA 4.0, May 2011)
- Note that performance is (in most cases, close to equivalent)
- Cases where it's not tend to be related to texture memory or transcendental intrinsics

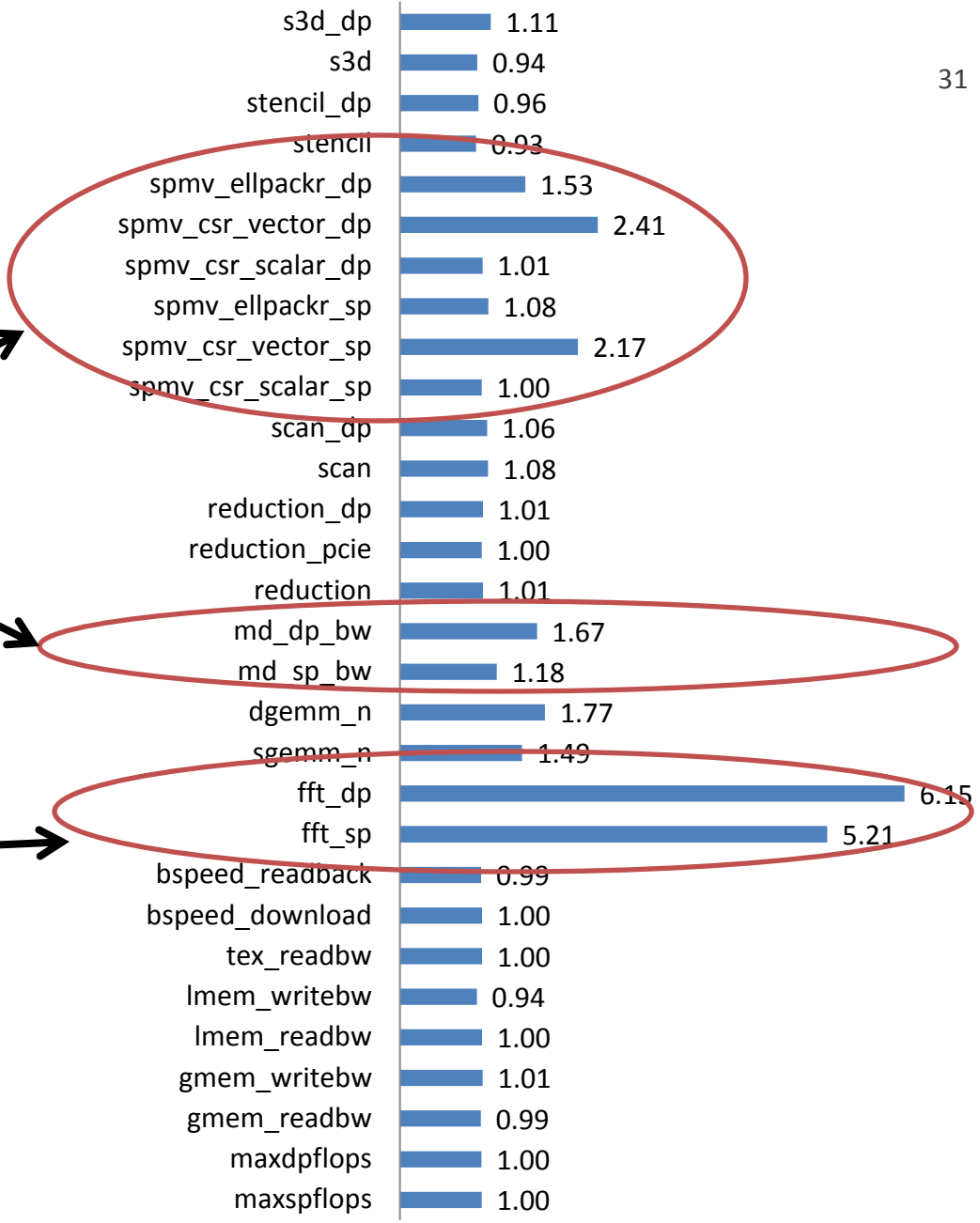


A. Danalis, G. Marin, C. McCurdy, J. Meredith, P.C. Roth, K. Spafford, V. Tipparaju, and J.S. Vetter, "The Scalable HeterOgeneous Computing (SHOC) Benchmark Suite," in Third Workshop on General-Purpose Computation on Graphics Processors (GPGPU 2010)`. Pittsburgh, 2010

A second look

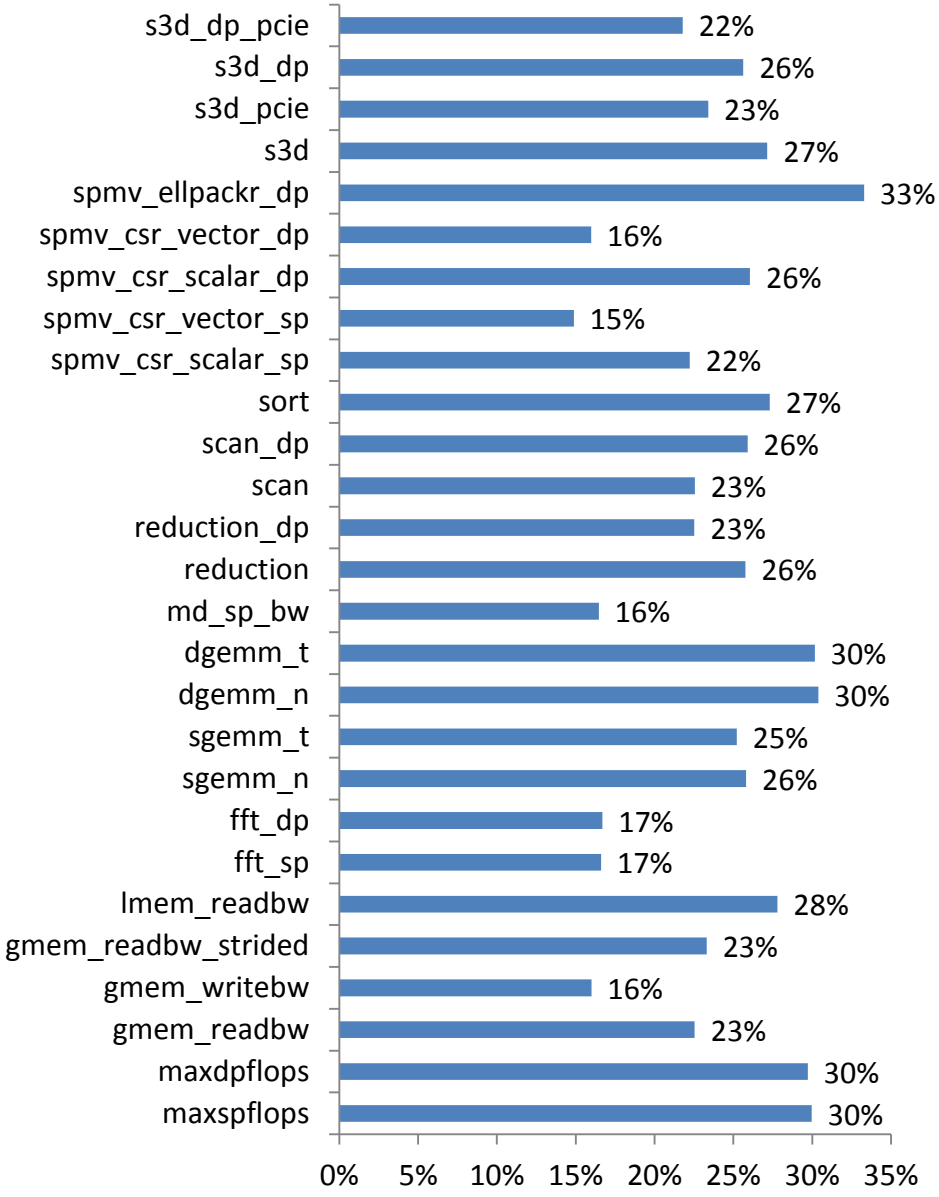
Reliant on texture memory

Dependent on transcendentals



M2090 v. M2070 SHOC Results

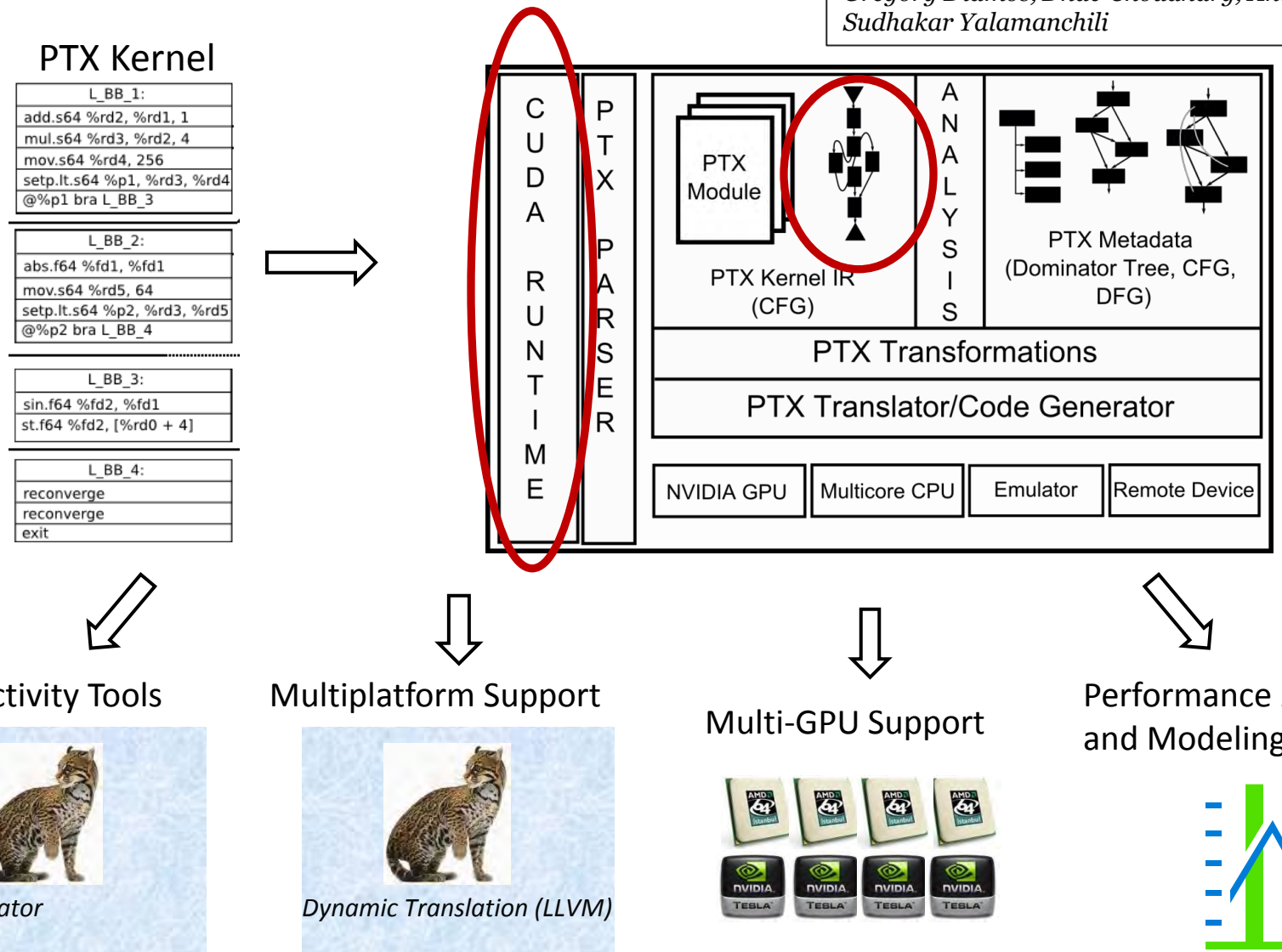
- M2090 v. M2070 in CUDA 4.0
- Performance improvements commensurate with expectation



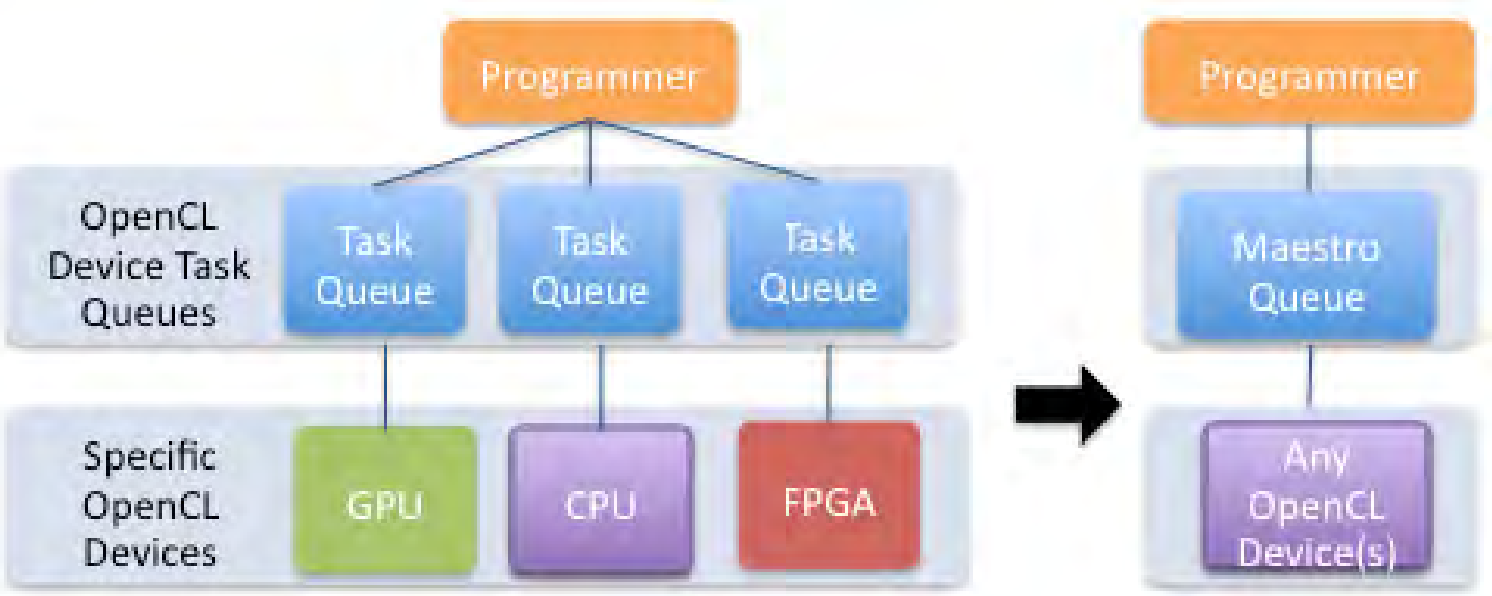
Ocelot: Dynamic Execution Infrastructure

http://code.google.com/p/gpuocelot/

Gregory Diamos, Dhuv Choudhary, Andrew Kerr, Sudhakar Yalamanchili



Maestro



- Portability
- Load balancing
- Autotuning

K. Spafford, J. Meredith, and J. Vetter, "Maestro: Data Orchestration and Tuning for OpenCL Devices," in *Euro-Par 2010 - Parallel Processing, vol. 6272, Lecture Notes in Computer Science, P. D'Ambra, M. Guarracino et al., Eds.: Springer Berlin / Heidelberg, 2010, pp. 275-86.*

Acknowledgements

- http://ft.ornl.gov
- http://keeneland.gatech.edu

U.S. DEPARTMENT OF ENERGY | Office of Science

NSF

Jeffrey Vetter, Dong Li, Anthony Danalis, Vinod Tipparaju, Philip Roth, Jeremy Meredith, Jan Hashmi, Kyle Spafford, Pat Worley, Gabriel Marin, Seyong Lee, Collin McCurdy, Olaf Storaasli

Not pictured: Dick Glassbrook (GT), Keeneland team