NSF Track 2D XHPC Keeneland Overview

Jeffrey Vetter, Jack Dongarra, Richard Fujimoto, Thomas Schulthess, Karsten Schwan, Sudha Yalamanchili, Kathlyn Boudwin, Jim Ferguson, Doug Hudson, Patricia Kovatch, Bruce Loftis, Jeremy Meredith, Jim Rogers, Philip Roth, Arlene Washington, Phil Andrews, Mark Fahey, Don Reed, Tracy Rafferty, Ursula Henderson, Terry Moore, and many others

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

http://keeneland.gatech.edu

BACKGROUND

NSF OCI RFP

- NSF 08-573 OCI Track 2D RFP in Fall 2008
  - Data Intensive
  - Experimental Grid testbed
  - Pool of loosely coupled grid-computing resources
  - Experimental HPC System of Innovative Design

An experimental high-performance computing system of innovative design. Proposals are sought for the development and deployment of a system with an architectural design that is outside the mainstream of what is routinely available from computer vendors. Such a project may be for a duration of up to five years and for a total award size of up to $12,000,000. It is not necessary that the system be deployed early in the project; for example, a lengthy development phase might be included. Proposals should explain why such a resource will expand the range of research projects that scientists and engineers can tackle and include some examples of science and engineering questions to which the system will be applied. It is not necessary that the design of the proposed system be useful for all classes of computational science and engineering problems. When finally deployed, the system should be integrated into the TeraGrid. It is is anticipated that the system, once deployed, will be an experimental TeraGrid resource, used by a smaller number of researchers than is typical for a large TeraGrid resource. (Up to 5 years duration. Up to $12,000,000 in total budget to include development and/or acquisition, operations and maintenance, including user support. First-year budget not to exceed $4,000,000.)

Oct 2008 Alternatives Analysis

- STI Cell (?)
- FGPAs
- Cyclops64 (?)
- Cray XMT
- Sun Rock/Niagara (?)
- ClearSpeed (?)
- Tensilica
- Tilera
- Anton
- SGI Molecule (?)
- Intel Larrabee (?)

- Graphics processors

- Others…

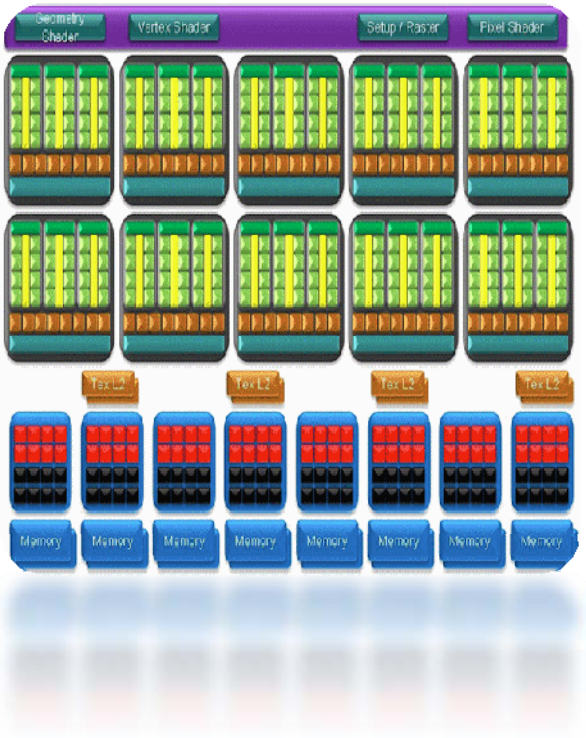
- Performance
  - Must show reasonable performance improvements at scale on real scientific applications of interest

- Programmability
  - Must be easy to re-port and re-optimize applications for each new architecture (generation) without large effort, delays
- Precision - Accuracy
  - Must provide impressive performance accurately
- Reliability
  - Must get high scientific throughput without job failures or inaccurate results

- Power and Facilities Cost
  - Must be reasonably affordable in terms of power and facilities costs

Alternatives analysis concluded GPUs were a competitive solution

- Success with various applications at DOE, NSF, government, industry
  - Signal processing, image processing, etc.
  - DCA++, S3D, NAMD, many others
- Community application experiences also positive
  - Frequent workshops, tutorials, software development, university classes
  - Many apps teams are excited about using GPGPUs

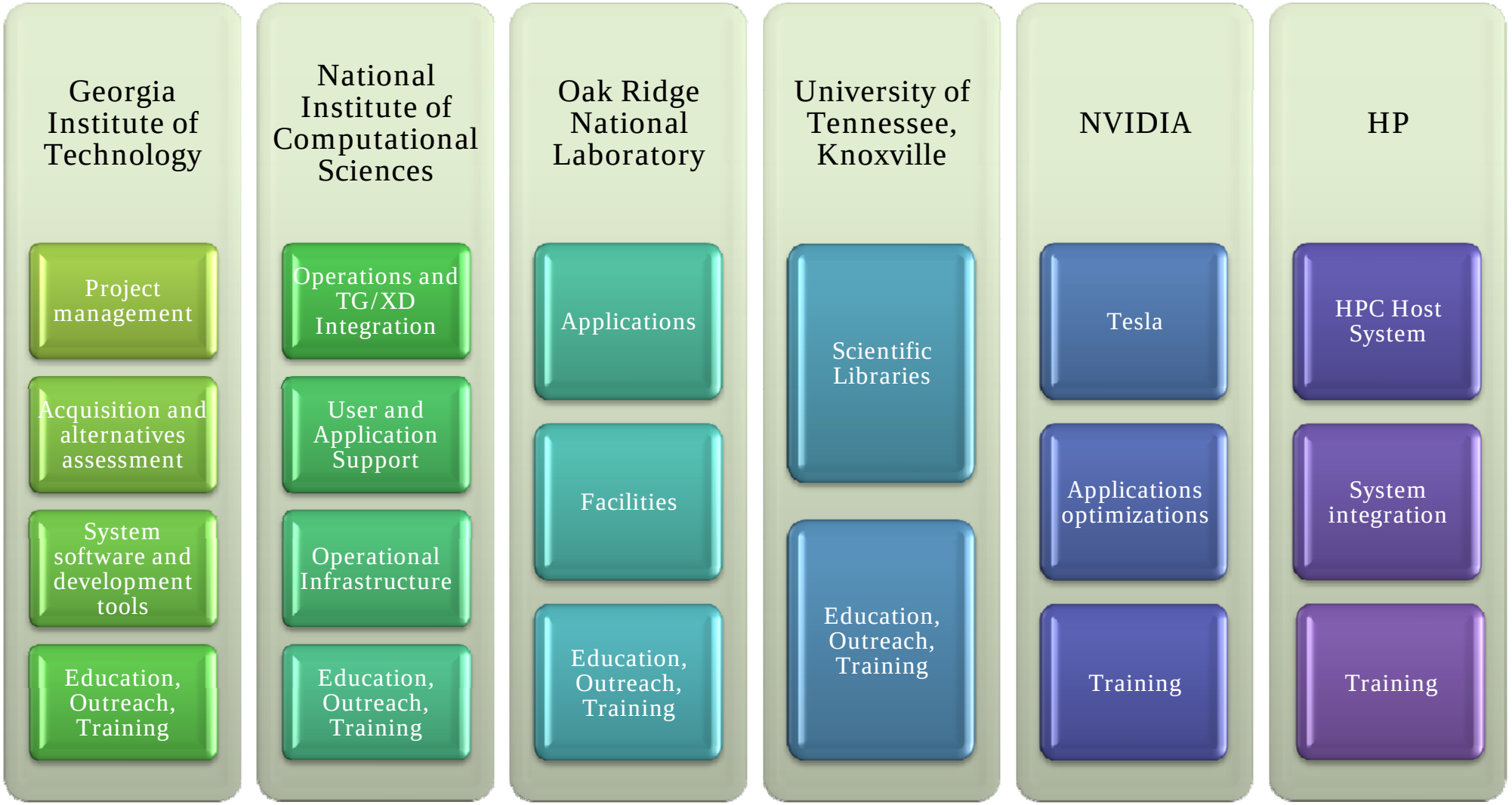


KEENELAND PROJECT OVERVIEW

Keeneland – An NSF-Funded Partnership to Enable Large-scale Computational Science on Heterogeneous Architectures

- Track 2D System of Innovative Design – Large GPU cluster
  - Initial delivery system – Spring 2010
  - Full scale system – Spring 2012
- Software tools, application development
- Operations, user support
- Education, Outreach, Training for scientists, students, industry

Keeneland Partners

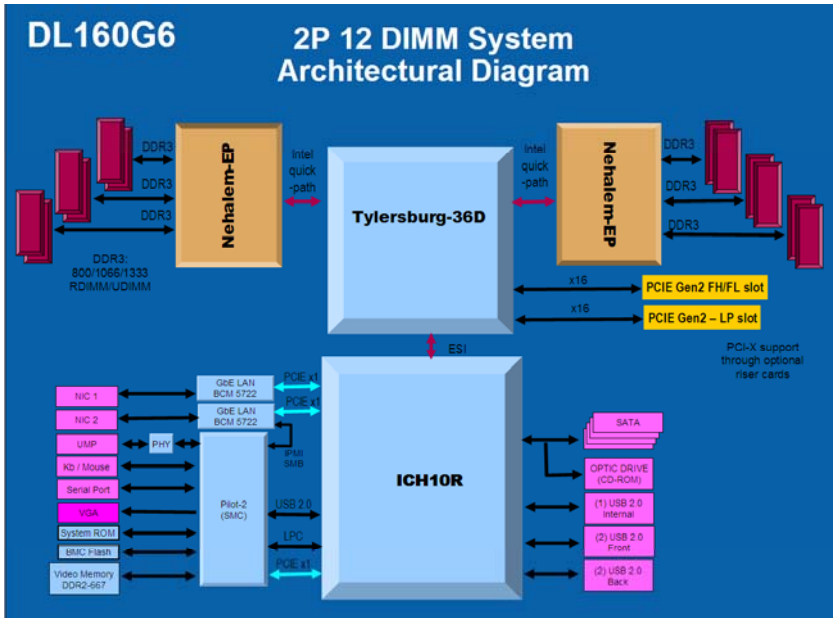


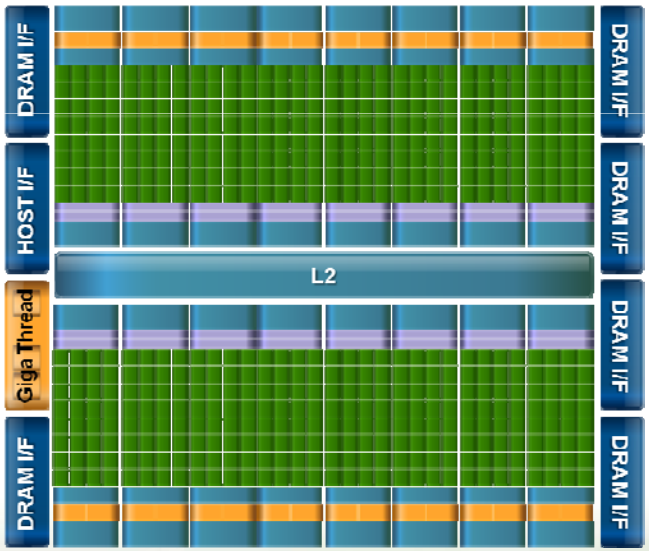
Keeneland Two Phase Deployment

- Why two phases?
  - Reliability, programmability in near-term GPGPUs was questionable
  - Rapidly changing area in both hardware and software
- Initial Delivery
  - Applications porting and refactoring
  - Early software development
    - OpenCL
    - Library Development
    - Tools
- Full Scale
  - 2012 delivery at ~3x budget
  - Depends on results of ID and architectural analysis

Keeneland Initial Delivery (ID) System

- Hewlett Packard Nodes
  - Dual socket Intel 2.8 GHz Nehalem-EP
  - 24 GB Main memory per node
- NVIDIA Servers
  - Fermi GPUs
- InfiniBand 4x QDR w/ full bisection interconnect
- Traditional Linux software stack augmented with GPU compilers, software tools, libraries
- Size: ~250 CPUs + ~250 GPUs
- Delivery and acceptance in Spring 2010

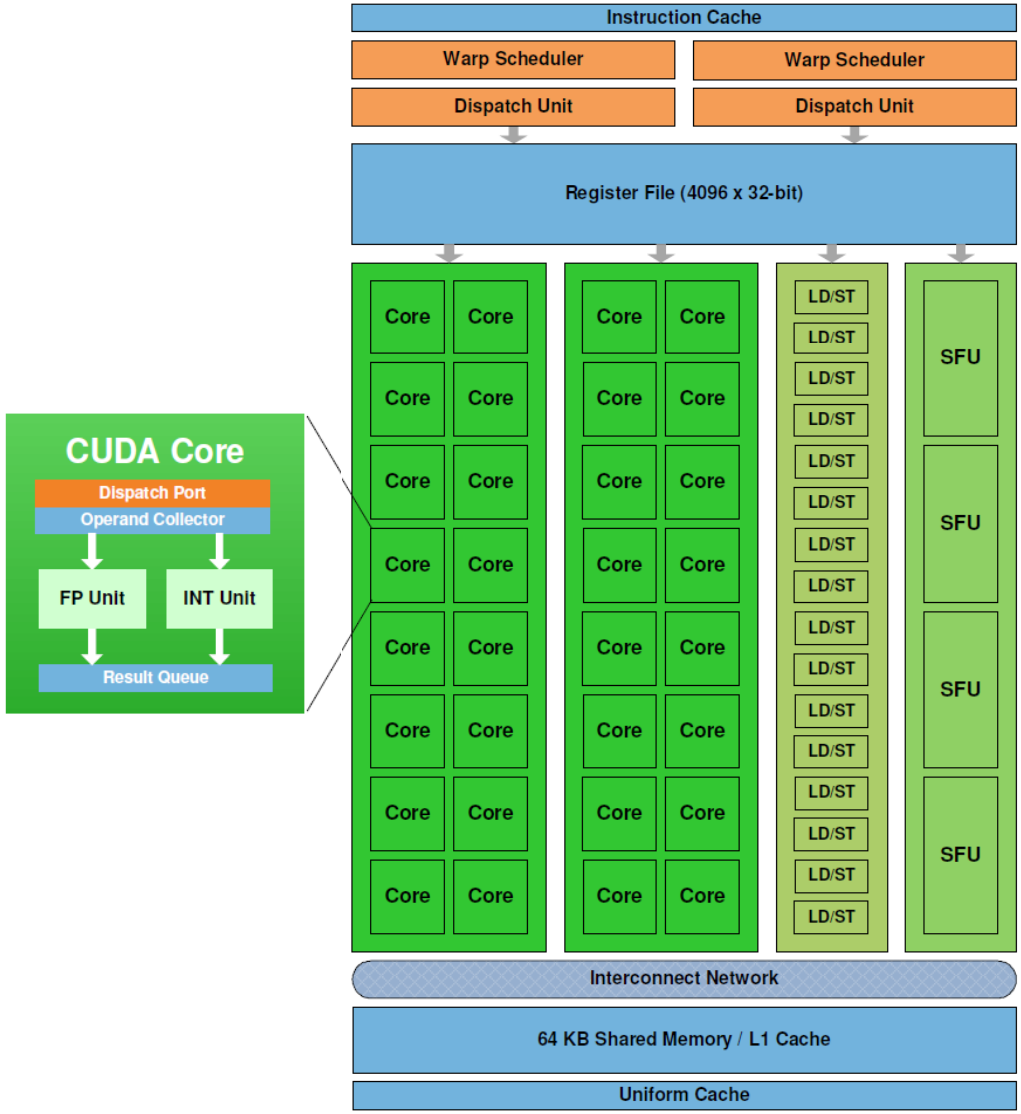




Georgia Tech NICS THE UNIVERSITY of TENNESSEE OAK RIDGE National Laboratory NVIDIA hp NSF

ID system will use NVIDIA's Fermi

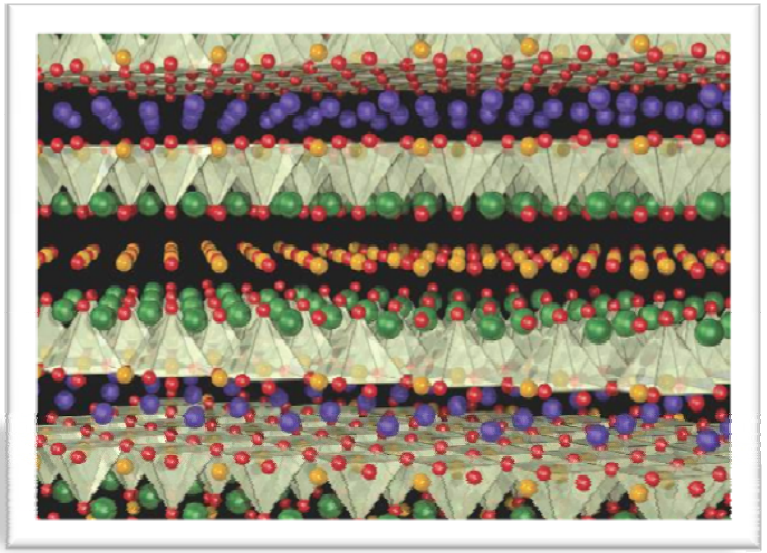
- 3B transistors
- ECC
- ~8x the peak double precision arithmetic performance over NVIDIA's last generation GPU.
- ~512 CUDA Cores featuring the new IEEE 754-2008 floating-point standard
- NVIDIA Parallel DataCache
- NVIDIA GigaThread Engine
- Debuggers, language support

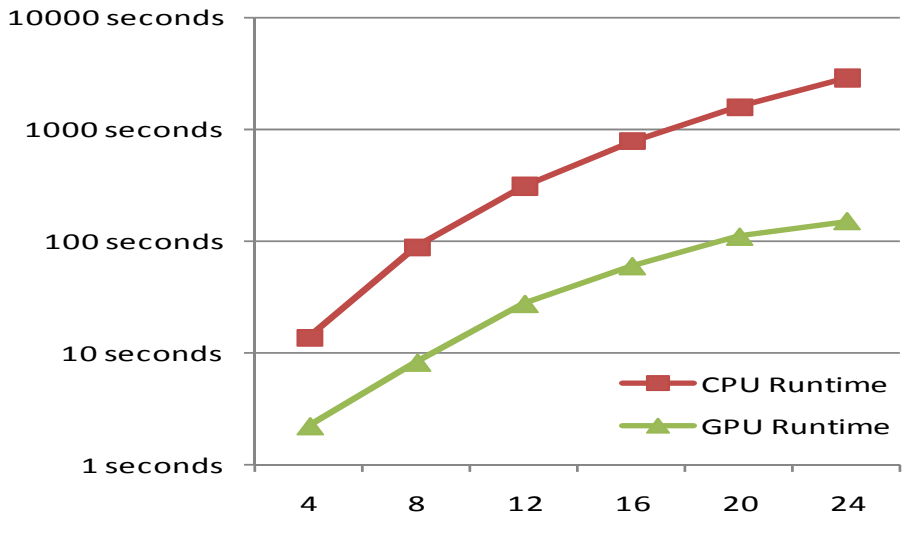


APPLICATIONS

Computational Materials - Case Study

- Quantum Monte Carlo simulation
  - High-temperature superconductivity and other materials science
  - 2008 Gordon Bell Prize
- GPU acceleration speedup of 19x in main QMC Update routine
  - Single precision for CPU and GPU: target single-precision only cards
  - Required detailed accuracy study and mixed precision port of app
- Full parallel app is 5x faster, start to finish, on a GPU-enabled cluster

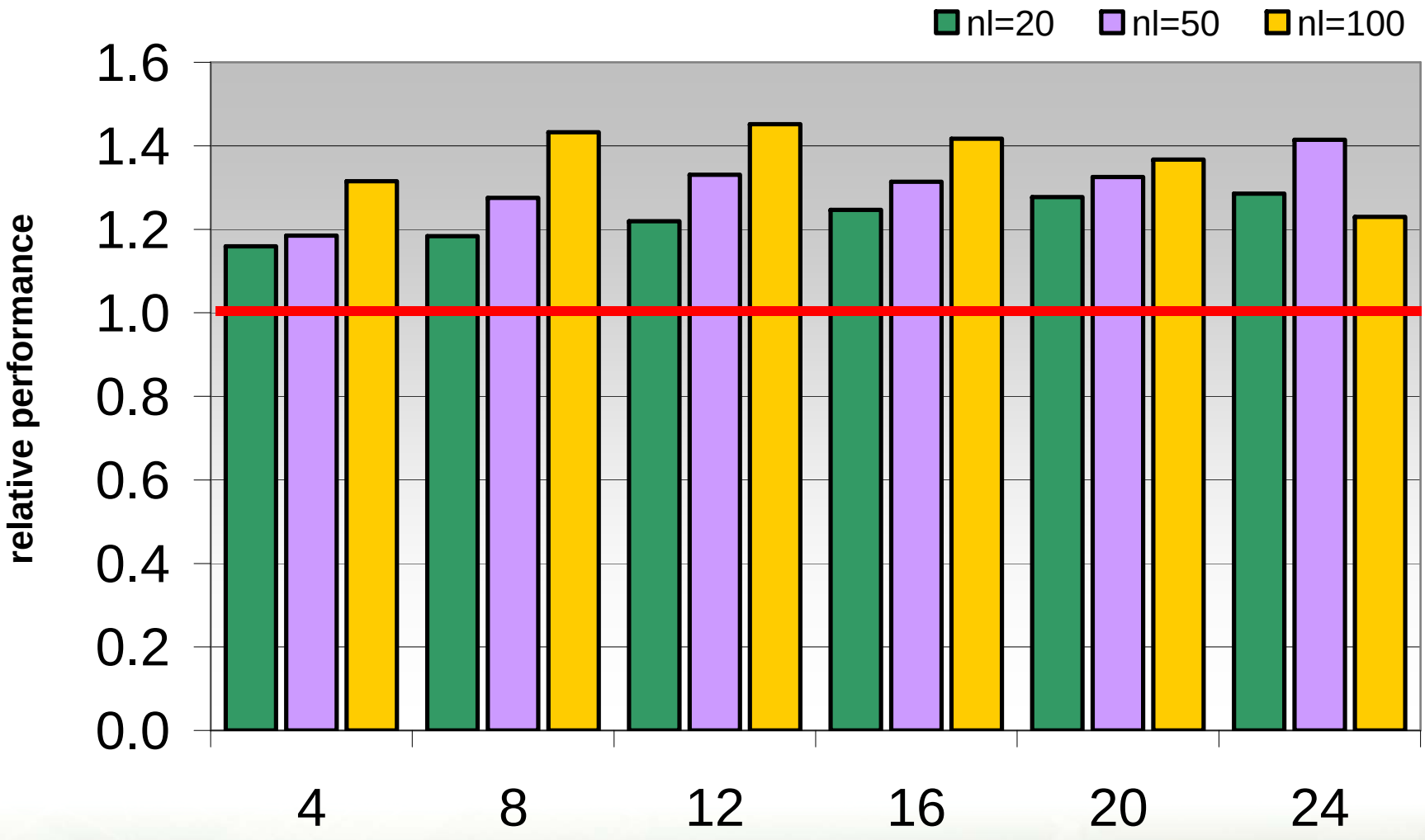




GPU study: J.S. Meredith, G. Alvarez, T.A. Maier, T.C. Schulthess, J.S. Vetter, "Accuracy and Performance of Graphics Processors: A Quantum Monte Carlo Application Case Study", *Parallel Comput., 35(3):151-63, 2009.*

Accuracy study: G. Alvarez, M.S. Summers, D.E. Maxwell, M. Eisenbach, J.S. Meredith, J. M. Larkin, J. Levesque, T. A. Maier, P.R.C. Kent, E.F. D'Azevedo, T.C. Schulthess, "New algorithm to enable 400+ TFlop/s sustained performance in simulations of disorder effects in high-Tc superconductors", SuperComputing, 2008. [*Gordon Bell Prize winner*]

Smart Data Transfers: Advantage over Naïve Transfers

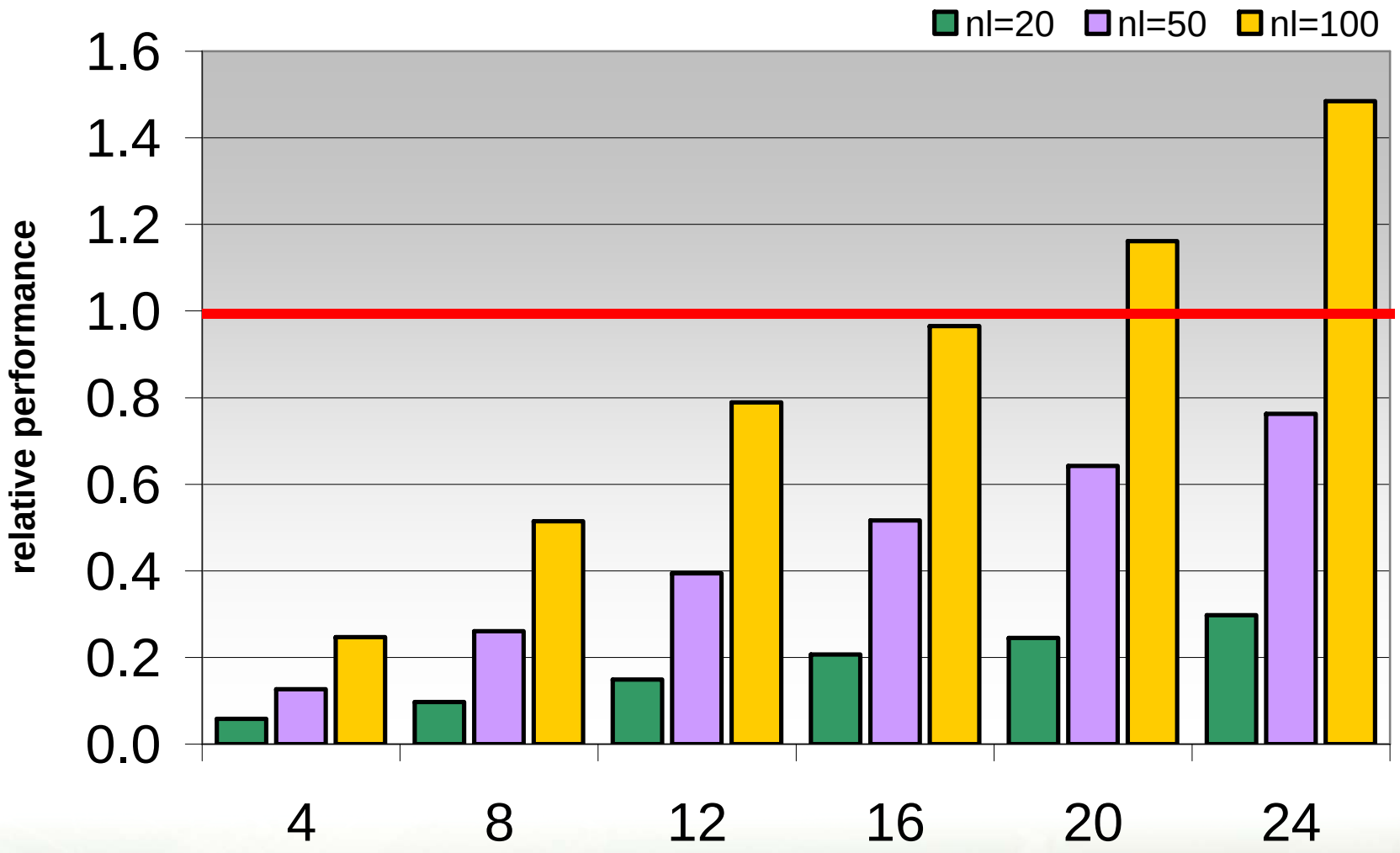


Georgia Tech NICS THE UNIVERSITY of TENNESSEE OAK RIDGE National Laboratory NVIDIA hp NSF

Exploiting Smarter Data Transfers

- Accelerating a single function is incomplete
  - Even if that function is 99% of your CPU runtime
- Functions that are otherwise insignificant become bottlenecks
  - If they require data to be on the CPU, they may be causing extra data transfers
  - Exploring functions that may even be a poor match for GPU acceleration is worthwhile

Accelerating Minor Functions: Advantage over GEMM-only



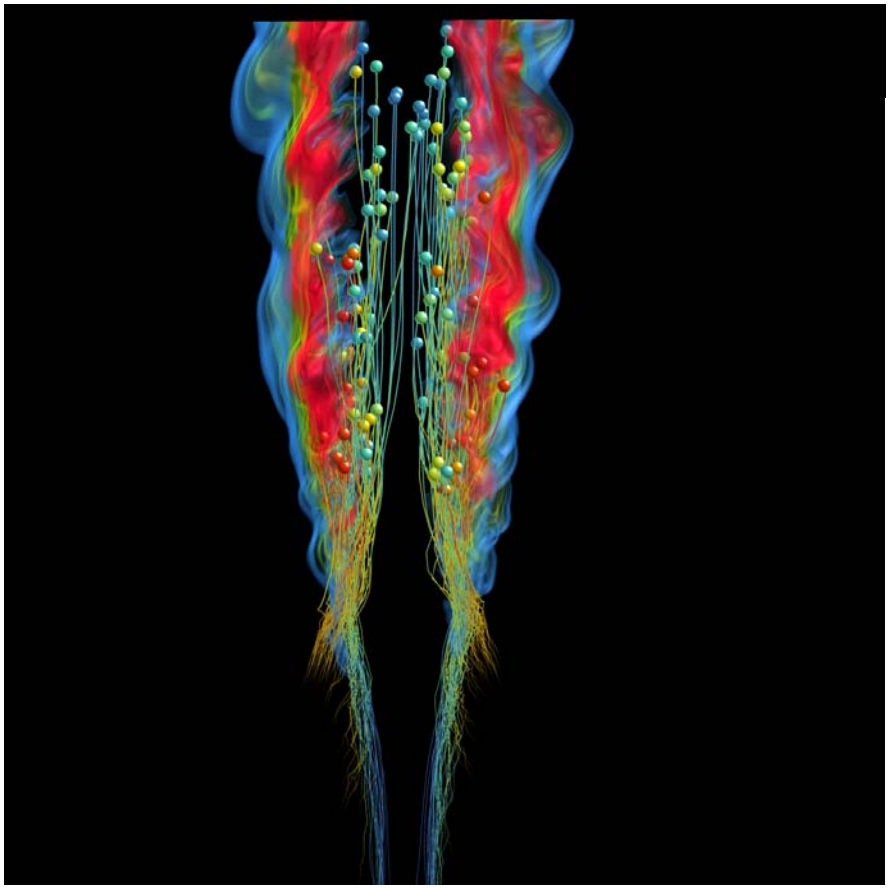
Accelerating Minor Functions

- Speedups:
  - 17x slower on small problems
  - 1.5x faster on large problems

- Effort:
  - more significant
    - these functions aren't necessarily a good match for GPUs
    - which makes them harder to port
  - and more dangerous
    - if they are not a good match, they can slow your code down

Combustion with S3D – Case Study

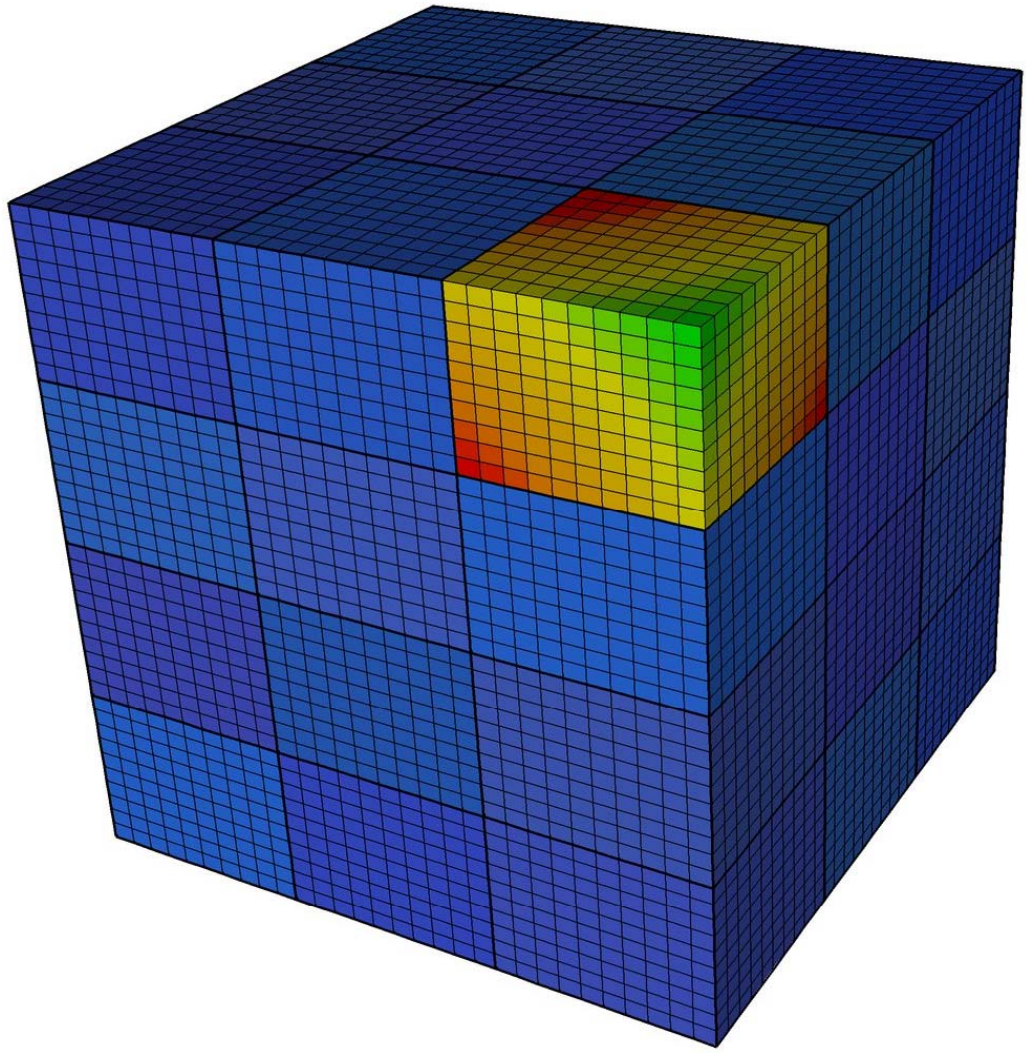
- Application for combustion - S3D
  - Massively parallel direct numerical solver (DNS) for the full compressible Navier-Stokes, total energy, species and mass continuity equations
  - Coupled with detailed chemistry
  - Scales to 150k cores on Jaguar
- Accelerated version of S3D's Getrates kernel in CUDA
  - 31.4x SP speedup
  - 16.2x DP speedup



K. Spafford, J. Meredith, J. S. Vetter, J. Chen, R. Grout, and R. Sankaran. Accelerating S3D: A GPGPU Case Study. Proceedings of the Seventh International Workshop on Algorithms, Models, and Tools for Parallel Computing on Heterogeneous Platforms (HeteroPar 2009) Delft, The Netherlands.

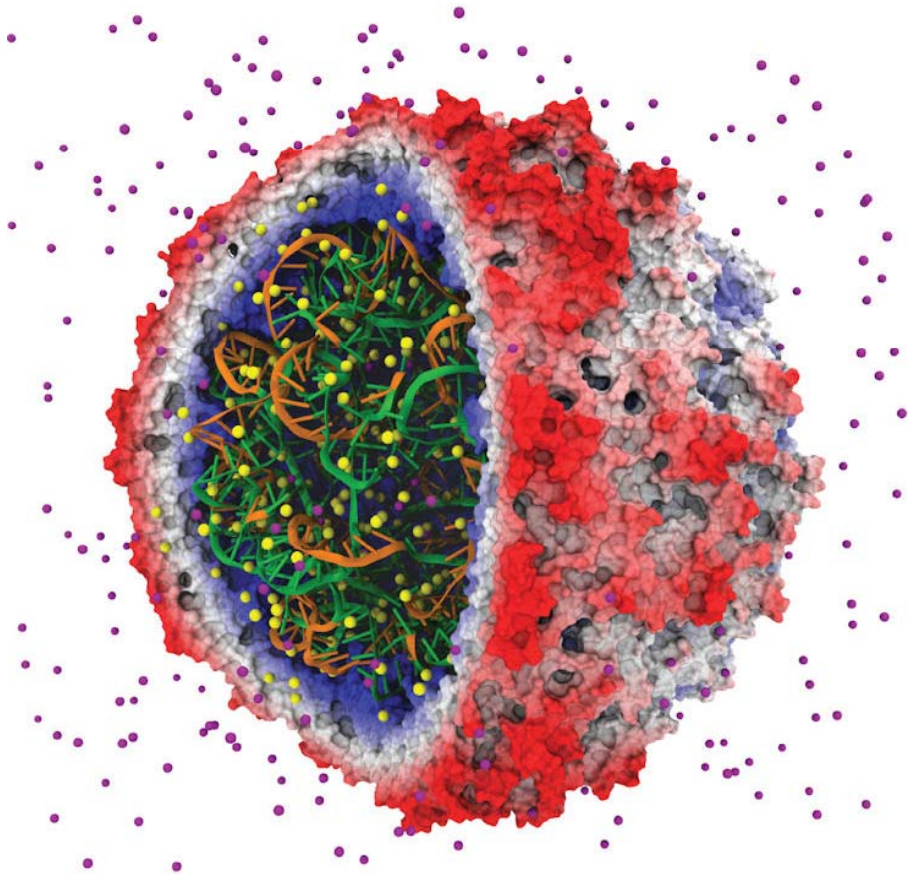
Challenges: Tailoring S3D to the GPU

- Main Tasks
  - S3D has a huge code base written in Fortran, porting the entire application without proof-of-concept is infeasible—must choose a kernel for acceleration
  - Expose fine-grained data parallelism
  - Modify algorithms to exploit the GPUs memory hierarchy
- Chemistry calculations – GetRates kernel
  - Based on temperature, pressure, and mass fraction, GetRates computes the speed at which chemical reactions are occurring
  - Comprises roughly 45-60% of runtime, varying with grid size
    - Expected to increase with more complex chemistry
  - Operates on a regular, structured 3-D grid



Biomolecular systems from NAMD Team – Not just us

- NAMD, VMD
  - Study of the structure and function of biological molecules
- Calculation of non-bonded forces on GPUs leads to 9x speedup
- Framework hides most of the GPU complexity from users



J.C. Phillips and J.E. Stone, "Probing biomolecular machines with graphics processors," *Commun. ACM, 52(10):34-41, 2009.*

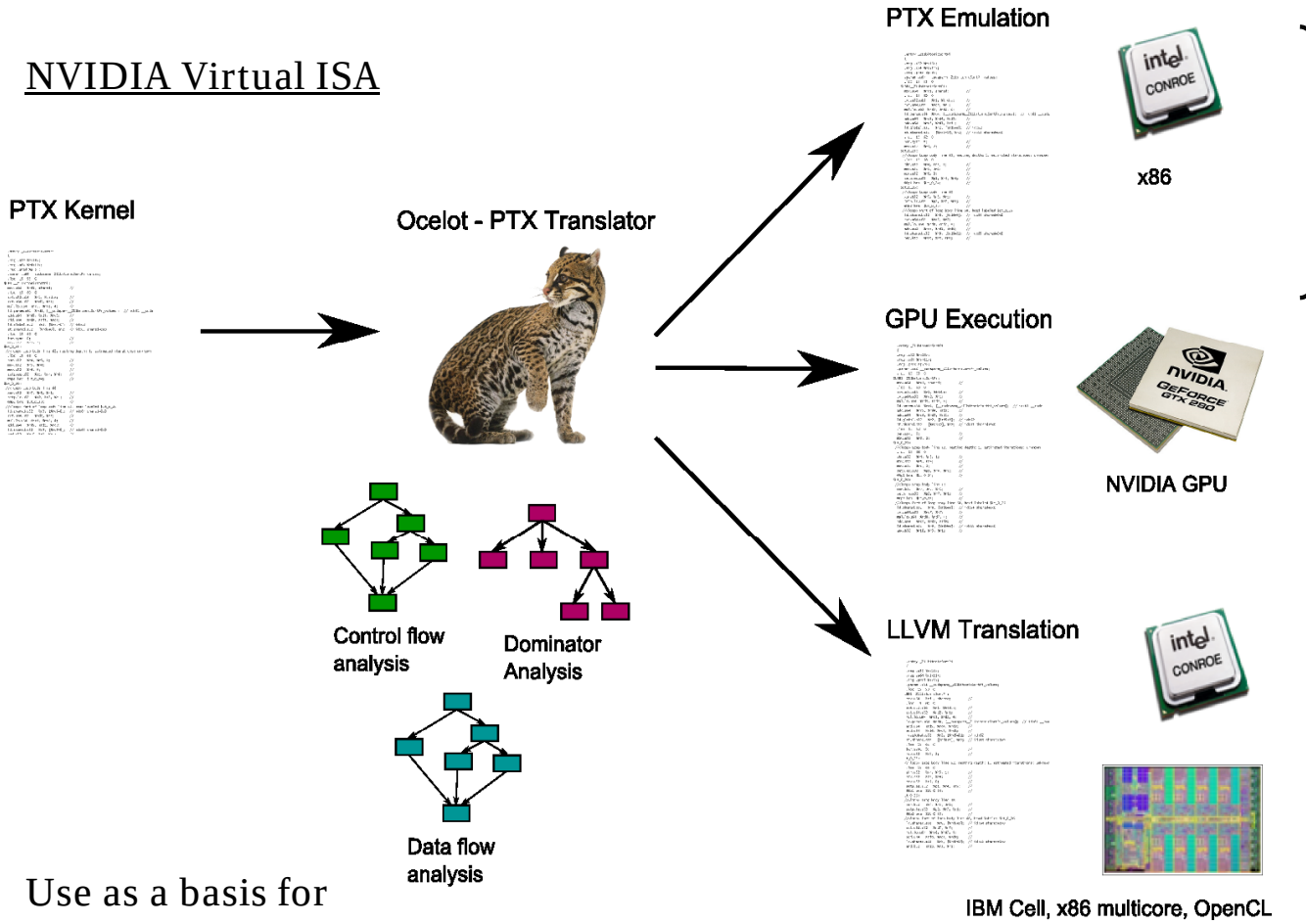
KEENELAND SOFTWARE

Keeneland Software Environment

- Integrated with NSF TeraGrid/XD
  - Including TG and NICS software stack
- Programming models
  - CUDA
  - OpenCL
  - NVIDIA Toolchain
  - PGI w/ accelerate
  - HMPP
- Additional software activities
  - Scientific libraries
  - Performance and correctness tools
  - Benchmarks
  - Virtualization

Ocelot: Dynamic Execution Infrastructure

NVIDIA Virtual ISA



PTX 1.4 compliant Emulation
- Validated on full CUDA SDK
- Open Source version released

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

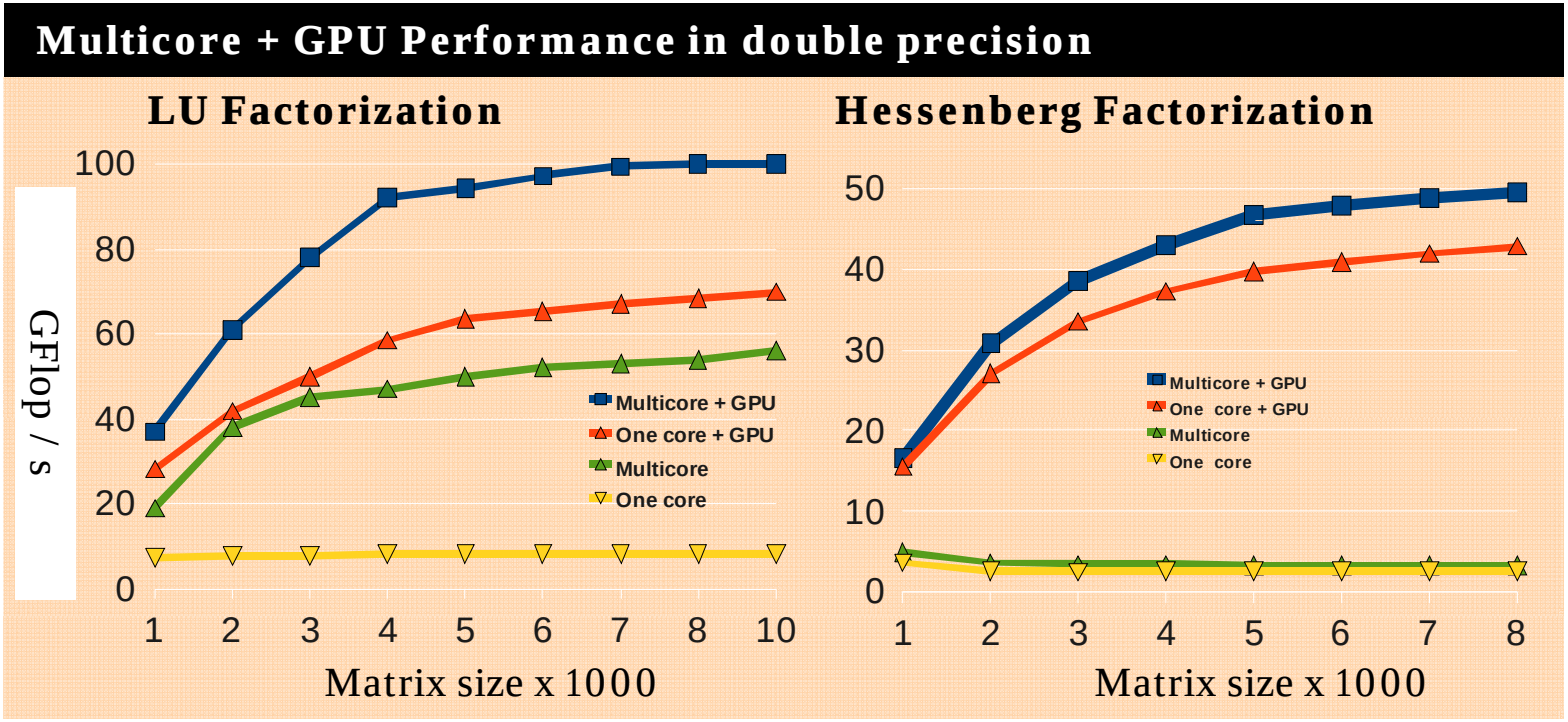
Use as a basis for

- Insight → workload characterization
- Performance tuning → detecting memory bank conflicts
- Debugging → illegal memory accesses, out of bounds checks, etc.

Gregory Diamos, Dhuv Choudhary, Andrew Kerr, Sudhakar Yalamanchili

Libraries: One and two-sided Multicore+GPU Factorizations

- These will be included in up-coming MAGMA releases
- Two-sided factorizations can not be efficiently accelerated on homogeneous x86-based multicores (above) because of memory-bound operations
  - MAGMA provided hybrid algorithms that overcome those bottlenecks (16x speedup!)



Jack Dongarra, Stan Tomov, and Rajib Nath

GPU : NVIDIA GeForce GTX 280
CPU : Intel Xeon dual socket quad-core @2.33 GHz

GPU BLAS : CUBLAS 2.2, dgemm peak: 75 GFlop/s
CPU BLAS : MKL 10.0 , dgemm peak: 65 GFlop/s

Risks

- Realized scientific results
- Fluid hardware environment
- Productive applications and users
  - Performance prediction and analysis
    - Selecting correct applications
    - Porting and optimizing them properly
  - Software tools:
    - Compilers
    - Libraries – aggregating library calls
    - Correctness, debugging, performance
  - Runtime
    - Schedulers, virtualization
  - Programming models
    - Programming large scale GPU clusters

Thank You!

More information

http://ft.ornl.gov

vetter@computer.org

Publications: http://ft.ornl.gov/pubs

http://keeneland.gatech.edu

http://www.cse.gatech.edu

http://www.cercs.gatech.edu

http://icl.cs.utk.edu

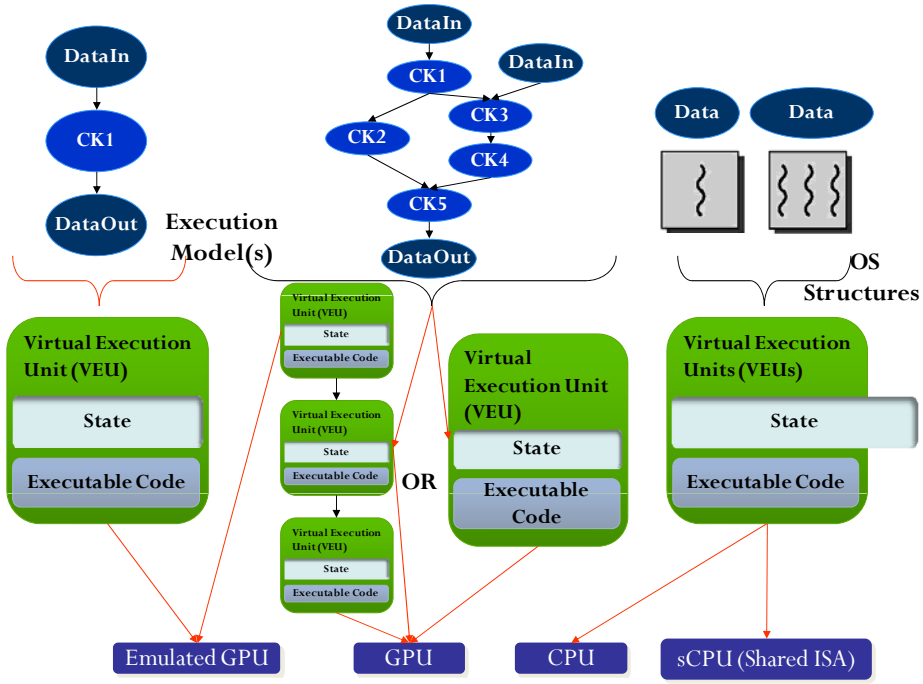
http://www.nics.tennessee.edu/

http://ft.ornl.gov

http://nsf.gov/dir/index.jsp?org=OCI

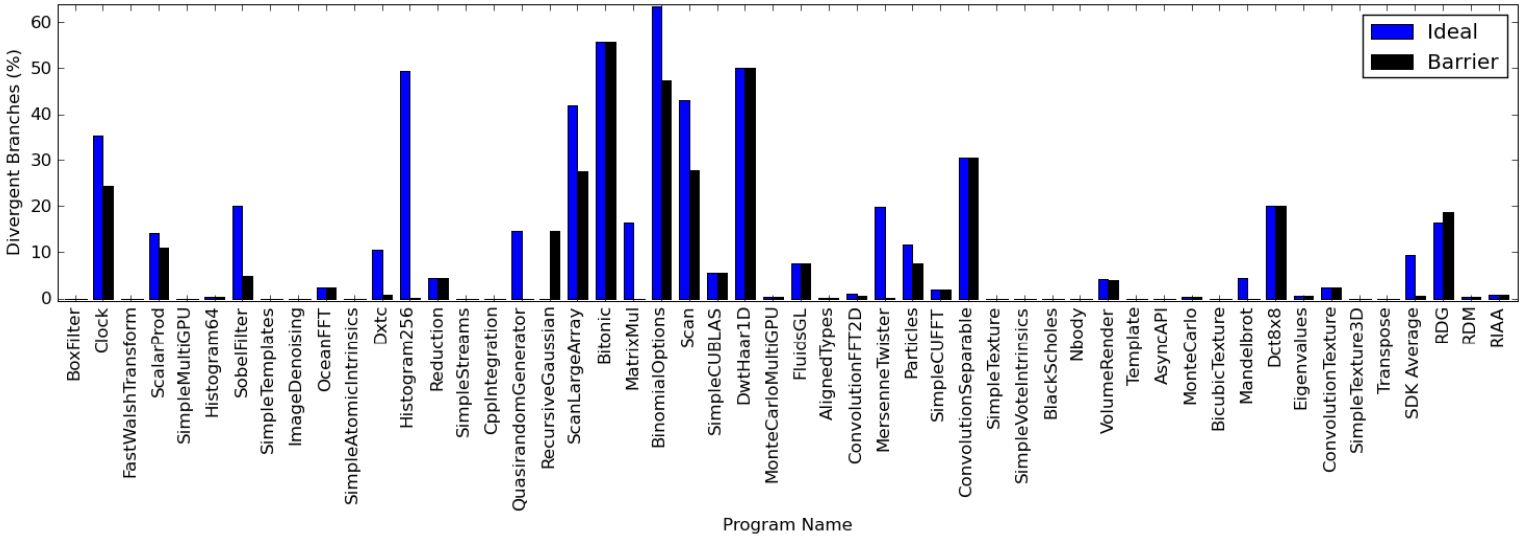
Hybrid Virtual Machine: HyVM

- Uniform runtime model for heterogeneous platforms:
- Hybrid Virtual Machines
  - Uniformity:
    - Virtual Execution Unit (VEU): hypervisor-level, uniform runtime representation for program executables
    - Heterogeneity-aware hypervisors: VMM-level management methods for improved platform utilization (incl. cache and energy) and application performance (SLAs)
    - Dynamic platform emulation: runtime CK compilation or re-writing for diverse accelerator targets (via LLVM)
  - High performance:
    - Commodity and custom VEU 'containers': Virtual Machines (VMs) – processes/threads – commodity cores; Special Execution Environments (e.g., NVIDIA) - Computational Kernels (CKs) - accelerators
    - Runtime and adaptive {CK} optimization for parallelism
    - Standards-compliant CK programming and runtime APIs (OpenCL, CUDA)
    - Compiler-based optimization techniques for {CK}



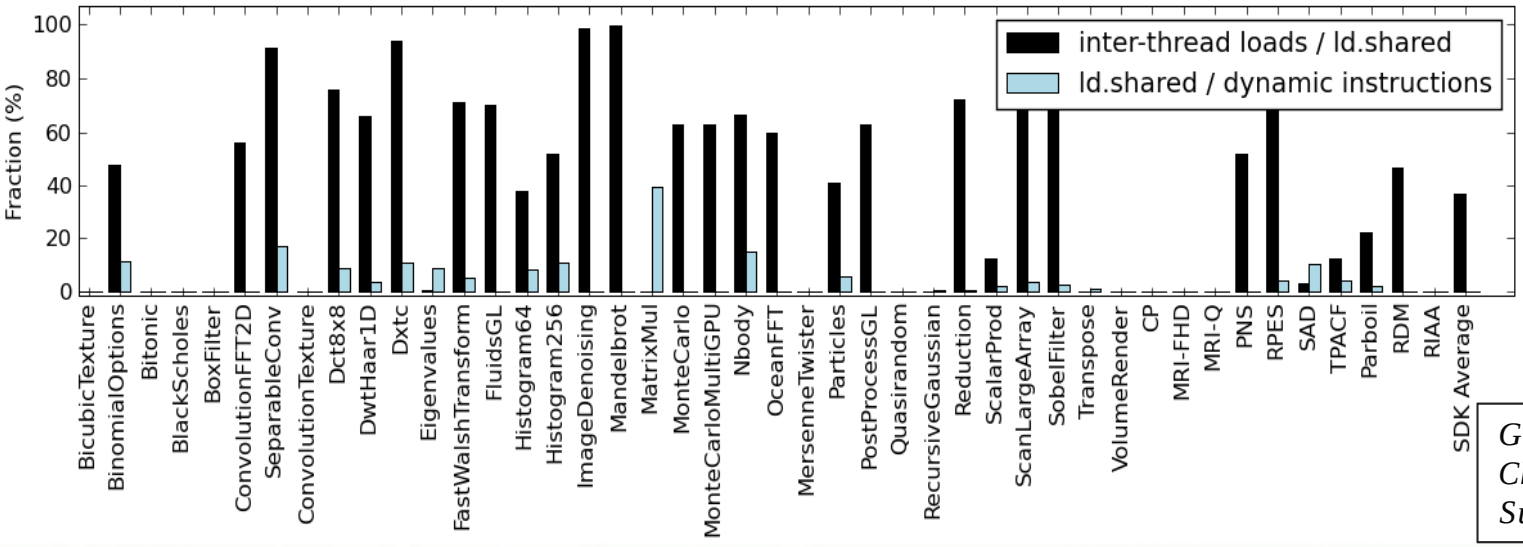
Ada Gavrilovska, Karsten Schwan, Sudha Yalamanchili, Jeff Vetter, and many PhD students

Workload Analysis: Examples



Branch Divergence

- Study of control Flow behavior
- Motivate synchronization support



Inter-thread Data Flow

- Study of data sharing patterns
- Motivate architectural support

Gregory Diamos, Dhuv Choudhary, Andrew Kerr, Sudhakar Yalamanchili