The Hybrid Multicore Consortium (HMC)

A multi-organizational partnership to support the effective development (productivity) and execution (performance) of high-end scientific codes on large-scale, accelerator based systems

http://computing.ornl.gov/HMC

Barney Maccabe, ORNL
March 9, 2010
SOS
Savannah, GA

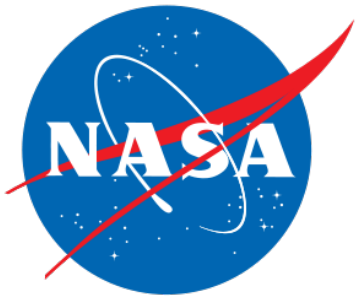
Membership is open to all parties with an interest in large-scale systems based on hybrid multicore technologies

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

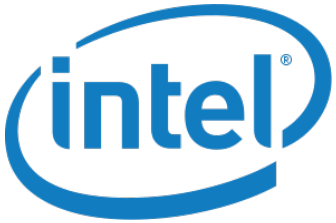
ORGANIZING PARTNERS



ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The organizing partners have made substantial investments in the deployment of large-scale, accelerator based systems

INDUSTRIAL AFFILIATES



Goal: Facilitate Production Readiness of Hybrid Multicore Systems

- Challenge
  - Existing applications require significant re-engineering to effectively manage the resources provided by large-scale, accelerator based systems
- Immediate goal
  - Identify obstacles to migrating high-end scientific applications to large-scale, accelerator based systems
  - Maintain long term perspective to ensure that today's efforts are not lost on tomorrow's platforms
- Long term goal
  - Identify strategies and processes, based on **co-design** among applications, programming models, and architectures, to support the effective development (**productivity**) and execution (**performance**) of large-scale scientific application

APPROACH

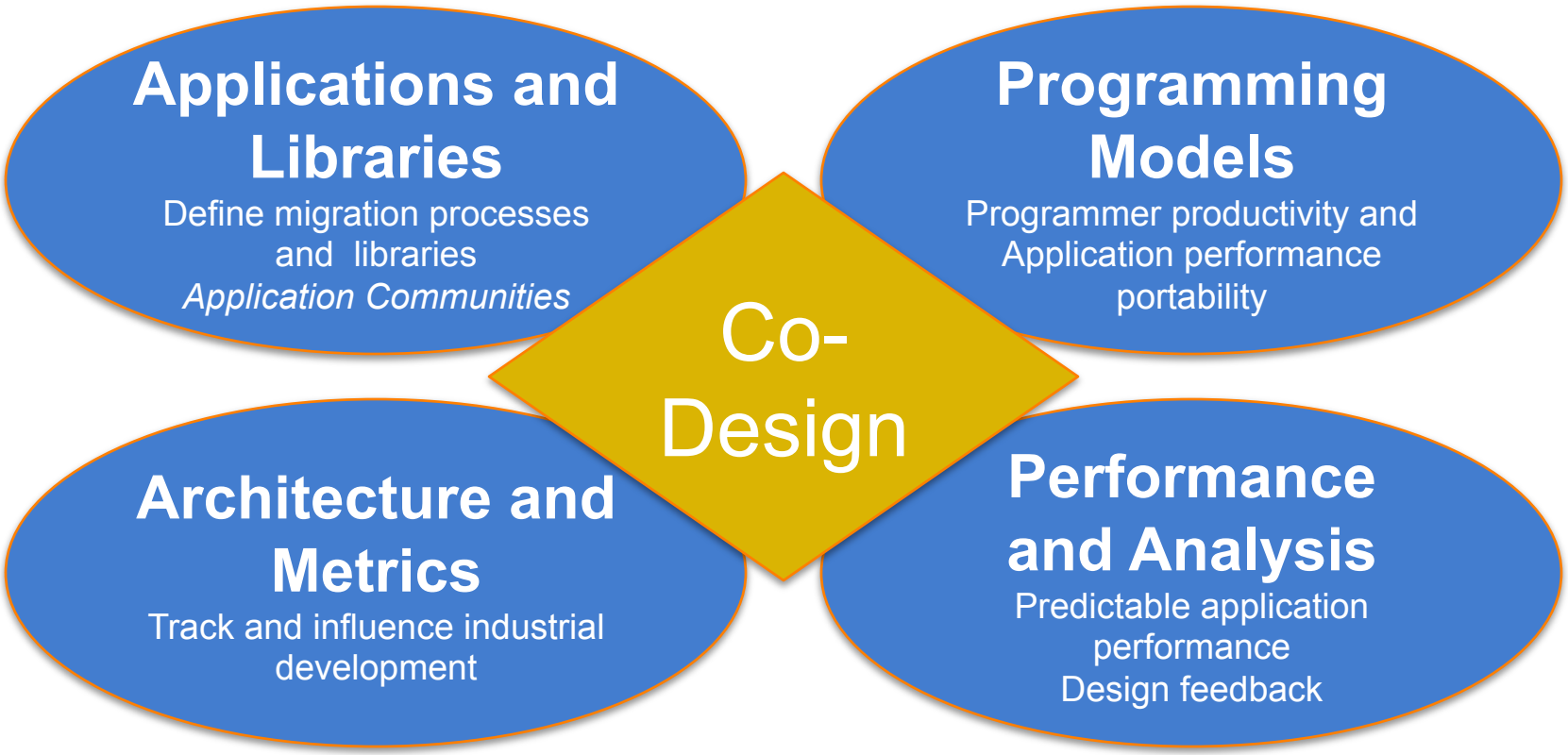
- Engage the broad community, including:
  - HW and SW developers (vendors),
  - Scientific computing community (users), and
  - Education / Training
- Maintain a **roadmap** documenting relevant projects and gaps
- Provide a unified voice to influence emerging standards and developers (both hardware and software)
- Serve as a clearinghouse to communicate successes and lessons learned
- Workshops and Web site
  - Define and update the roadmap
  - Support interactions (clearinghouse and engagement)
- Maintain long term vision while providing solutions for near term systems (“Think globally, act locally”)

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

TECHNICAL COMMITTEES (TC)



TC ORGANIZERS

- **Applications and Libraries (AL)**
  - John Turner (ORNL) and Sriram Swaminarayan (LANL), co-chairs
  - Erich Strohmaier (LBNL) and Thomas Schulthess (ETH)
- **Programming Models (PM)**
  - Kathy Yelick (LBNL), chair
  - Ken Koch (LANL) and John Turner (ORNL)
- **Architecture and Metrics (AM)**
  - Steve Poole (ORNL), chair
  - Jeff Broughton (LBNL) and Ken Koch (LANL)
- **Performance and Analysis (PA)**
  - Adolfy Hoisie (LANL), chair
  - Jeffrey Vetter (Georgia Tech, ORNL) and Costin Iancu (LBNL)

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

TECHNICAL OVERSIGHT COMMITTEE

- Barney Maccabe (ORNL), chair; Stephen Lee (LANL); John Shalf (LBNL); and TC chairs
- Responsible for
  - Managing consortium activities
    - Workshops
    - Web site
    - Roadmap
  - Internal communication within the consortium
- **Workshop Committee**
  - Al Geist (ORNL), chair
  - Technical Oversight Committee

OAK RIDGE National Laboratory

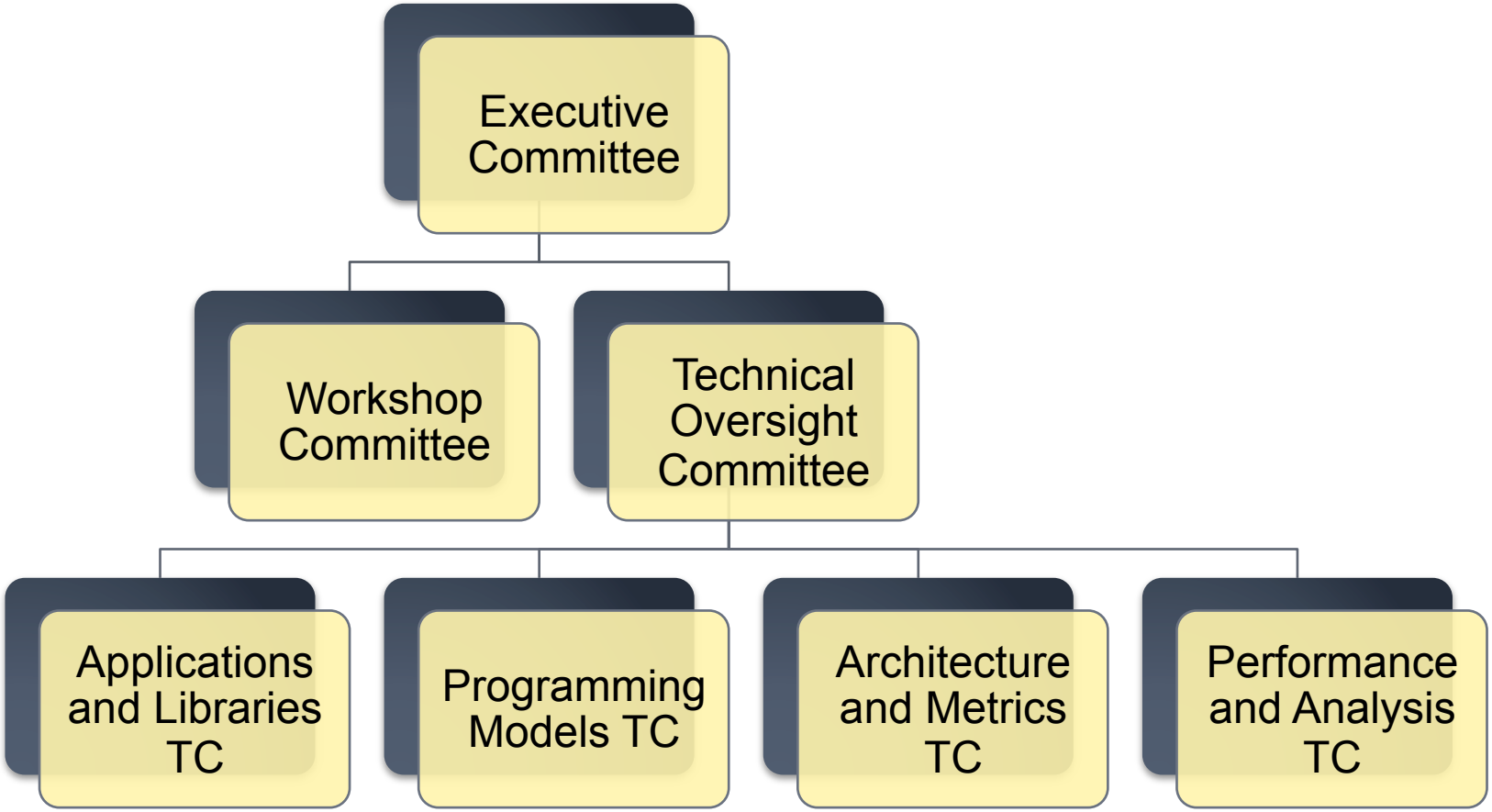
BERKELEY LAB

Los Alamos NATIONAL LABORATORY

EXECUTIVE COMMITTEE

- Jeff Nichols (ORNL), chair; Horst Simon (LBNL) and Andy White (LANL)
- Broad oversight of consortium activities
- Responsible for
  - Providing strategic direction and
  - External communication

Structure of the HMC Organizing Members



DEVELOPING THE ROADMAP

Roughly based on the HEC FSIO roadmap

http://institutes.lanl.gov/hec-fsio/

First Workshop held January 20-21, 2010
Hyatt Regency San Francisco Airport Hotel

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

THE ROADMAP

- Technologies we believe need to be developed to make large-scale, accelerator based systems **production** *ready*
- Document relevant projects
- Identify **gaps** and provide **grades**
- **Dashboard** might be a better name

Process of the Workshop

- Breakouts based on Technical Areas: Identify and grade topics
  - Applications and Libraries
  - Programming models
  - Architectures and Metrics
  - Performance and Analysis
- Report topics and grades from breakouts
- Pair wise breakouts to identify common topics
- Technical area breakouts to finalize topics and grades
- **Second report for each technical area**
- Crosscuts identified
  - Testbed Systems
  - Resilience
  - Operating Systems
  - I/O and Storage systems

GRADING CRITERIA

Urgency How soon is it needed?	**Duration** How long will it be useful?	**Responsive** Will adding resources help?	**Applicability** How broadly can it be used?	**Timeline** How soon can we expect it?
Critical Needed now	**Long** Useful for the foreseeable future	**High** Resources enable significant progress	**Broad** Applicable beyond scientific computing	**Immediate** Results within 1-2 years
Important Needed within 3 years	**Medium** Useful for Exascale	**Moderate** Resources enable progress	**Science** Applicable to general scientific computing	**Soon** Results within 2-5 years
Useful Needed after 3 years	**Near** Only useful for immediate systems	**Low** Resources have little affect on progress	**Narrow** Only applicable to HPC systems	**Eventually** Results after 5 years

Applications and Libraries

John Turner (ORNL)

Sriram Swaminarayan (LANL)

Notes from Applications and Libraries

- Hardware simulators are useful before hardware is available
- Once hardware is available, we need a few per site, or one per developer
- Small testbeds of 10-20 nodes within a year
- Larger platforms of 100-1000 nodes with promise of 10x the performance 5 years from now

Architecture and Metrics Summary

Topic	Urgency	Duration	Responsive	Applicability	Timeline
Math & I/O Libraries	Critical	Medium	Moderate	Broad	Immediate
Novel Algorithm Research	Critical	Long	High	Broad	Soon
Profiling Tools	Important	Long	High	Science	Eventually
Generic Scientific Toolkits	Useful	Long	High	Broad	Eventually
Fault tolerance tools	Important	Long	High	Science	Eventually

Math and I/O Libraries

- Description
  - Numerical libraries
    - BLAS, LAPACK, Trilinos, FFTW, BGL, grid operators, AMR
  - I/O libraries
- Notes from discussion
  - Building-blocks of apps
  - Scalable from desktop to HPC
  - Diffusion of knowledge beyond specific apps
  - Portability critical
- Relations to other TCs
  - Performance
  - Programming Models
  - Architecture
- Related Projects
  - MAGMA
  - cuBLAS
  - Trilinos
  - PETSc
  - Adios
  - PVFS, PLFS, GPFS, etc.

Urgency	Duration	Responsive	Applicability	Timeline
Critical	Medium	Moderate	Broad	Immediate

Novel Algorithm Research

- Description
  - Building-blocks of apps
  - Methods development
  - Algorithm is some version of above method that we can implement
  - Implementation is a specific instantiation of that method
- Notes from discussion
  - Implementations need to be architecture aware
  - Spatial and temporal locality is key
  - Time to solution should be kept in mind in addition to complexity and flops
- Relations to other TCs
  - Programming Models
- Related Projects
  - CFDNS on Cell
  - FEAST-GPU

Urgency	Duration	Responsive	Applicability	Timeline
Critical	Long	High	Broad	Soon

PROFILING TOOLS

- Description
  - Data motion feedback
  - Data location
  - Time to solution is critical
  - Energy to solution is critical
- Notes from discussion
  - Equal ownership with performance
  - Cache hits/misses
  - Retired operations
  - Dual-issue
  - Bus contention
  - Latency
  - Packet size
  - Ops/load can be useful
- Relations to other TCs
  - Performance
  - Architecture
- Related Projects
  - OpenSpeedshop
  - VTUNE
  - VAMPIR
  - Oprofile
  - gprof
  - Tau

Urgency	Duration	Responsive	Applicability	Timeline
Important	Long	High	Science	Eventually

Abstract Scientific Toolkits

- Description
  - High-level expression of math / physics
  - Physics resides in Applications, CS resides in Programming models
- Notes from discussion
  - Grid operation libraries
  - PDE libraries
  - Graph libraries
  - Success requires strong interaction between CS and physics experts
- Relations to other TCs
  - Programming Models
- Related Projects
  - SCOUT
  - libMesh
  - netCDF
  - Toolkits within matlab
  - BGL/PBGL

Urgency	Duration	Responsive	Applicability	Timeline
Useful	Long	High	Broad	Eventually

RESILIENCE / FAULT TOLERANCE

- Description
  - System reports faults so app can continue
- Notes from discussion
  - Must move beyond checkpoint / restart
  - Minimal impact on resources
  - Generic interaction with system
- Relations to other TCs
  - Performance
  - Programming Models
  - Architecture
- Related Projects
  - MAGMA
  - cuBLAS
  - Trilinos
  - PETSc
  - Adios
  - PVFS, PLFS, GPFS, etc.

Urgency	Duration	Responsive	Applicability	Timeline
Critical	Long	High	Science	Eventually

Programming Models

Paul Henning (LANL)
Sadaf Alam (CSCS)
Jonathan Carter (LBL)

CHARGE TO PROGRAMMING MODELS

- Identify and report on programming models for developing applications on large-scale (accelerator-based) hybrid computer systems in the near term and in the future.
- *Identify the types and degrees of parallelism provided by hybrid cores and to define key architectural metrics of this class of hybrid machine*.

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

Summary of Programming Models

- Areas of interest:
  - Code and performance portability
  - Developer productivity: tools, programming for "mere mortals"
  - Data layout & motion, multiple disjoint address spaces, SIMD length, etc.
- Relation to other TCs
  - Applications: algorithm design/selection
  - Architecture: design roadmaps
  - Performance: data motion costs, system modeling

Programming Models Summary

Topic	Urgency	Duration	Responsive	Applicability	Timeline
Technology Evaluation & Selection	Important	Medium	High	Narrow	Immediate
Translation Tools	Critical	Medium	High	Science	Soon
Debugging and Performance Support	Important	Long	High	Broad	Soon
HMC & non HMC Performance Portability	Important	Long	Moderate	Broad	Eventually
Expressive Programming Environments	Useful	Long	Moderate	Broad	Eventually

TECHNOLOGY EVALUATION AND SELECTION

- Description
  - Provide “honest broker” for evaluation of technologies
  - Match algorithms to hardware.
  - Influence future investments
- Notes from discussion
  - Reference implementations
  - Best practices
  - White papers & books
  - Benchmark suites
  - Illustrate range of available
- Relations to other TCs
  - Applications: get requirements
  - Libraries: serve as examples
  - Architecture: selection
- Related Projects
  - CUDA Zone, motifs, MAGMA project
  - Accelerator-oriented HPC benchmark suites

Urgency	Duration	Responsive	Applicability	Timeline
Important	Medium	High	Narrow (a plus!)	Immediate

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

TRANSITION TOOLS

- Description
  - Tools to facilitate refactoring existing code bases to new programming paradigms.
  - Tools for identifying acceleration opportunities
  - Choosing the right hardware for the application
- Notes from discussion
  - Language interoperability is crucial
- Relations to other TCs
  - Applications: requirements
  - Performance: modeling of systems
- Related Projects
  - Compiler directives (e.g. OpenMP)
  - Language translation (e.g. C-to-CUDA)
  - Performance analysis & modeling tool extensions (e.g. ROSE, TAU)

Urgency	Duration	Responsive	Applicability	Timeline
Critical	Medium	High	Science	Soon

Debugging and Performance Support

- Description
  - Capability to access debugging and performance data on HMC hardware
  - Couple low-level data to high-level languages
  - Bridging multiple ISAs, clocks, and address spaces
- Notes from discussion
  - Goal: Uniform interface between tools and architectural features for robust tools
  - Hardware changes on longer timeline
- Relations to other TCs
  - Architecture: need more hooks for
- Related Projects
  - PAPI, NVIDIA Nexus, vampir, oprofile, TAU, TotalView, Allinea DDT, Charm++ projection tool

Urgency	Duration	Responsive	Applicability	Timeline
Important	Long	High	Broad	Soon

HMC AND NON-HMC PERFORMANCE PORTABILITY

- Description
  - One code base for performance on multiple architectures
- Notes from discussion
  - What are the implications of maintaining multiple code bases?
  - What breadth of application space?
- Relations to other TCs
  - Applications: what is “acceptable” performance, when needed?
  - Architecture: compatibility
- Related Projects
  - MCUDA, OpenCL, CUDA-Fortran
  - Autotuning

Urgency	Duration	Responsive	Applicability	Timeline
Important	Long	Moderate	Broad	Eventually

EXPRESSIVE PROGRAMMING ENVIRONMENTS

- Description
  - Reduce effort to utilize accelerator hardware
  - Express developer's intent (more declarative)
- Notes from discussion
  - Accelerated PGAS expected within a year
  - Question about balance between research and development and the impact on timeline
- Relations to other TCs
  - Applications
- Related Projects
  - Thrust
  - MATLAB
  - Python (Copperhead,SciPy)
  - Domain specific languages
  - HPCS Languages
  - LabVIEW/FPGA Workflow

Urgency	Duration	Responsive	Applicability	Timeline
Useful	Long	Moderate	Broad	Eventually

Architecture and Metrics

Steve Poole (ORNL)
Ken Koch (LANL)
Jeff Broughton (LBNL)

SUMMARY OF ARCHITECTURE & METRICS

- Areas of interest to this TC
  - Accelerator/System Interfaces
  - Accelerator Design
  - System Software
  - System Design
  - Simulation & Modeling
  - Metrics
- Relation to other TCs
  - Programming Models: Ease of programming & debugging
  - Performance: Enhance throughput & provide measurement tools
  - Applications & Libraries: same

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

Architecture and Metrics Summary

Topic	Urgency	Duration	Responsive	Applicability	Timeline
Accelerator/ System Interface	Critical	Long	High	Broad	Soon
Accelerator Design	Critical	Long	High	Broad	Soon
System software	Critical	Long	High	Science	Immediate
Simulation & Modeling	Important	Long	High	Broad	Immediate
Metrics	Important	Long	High	Broad	Soon
System Design	Important	Medium	High	Broad	Immediate

ACCELERATOR/SYSTEM INTERFACE

- Description
  - [1] Enhance bandwidth, latency between CPU & Accel. e.g. On-die or in-socket or in-stack (peer or parent)]. Improve power eff.
  - [2] Shared address space between accelerator and host CPU
  - [4] Efficient synchronization
  - [3] Enhance BW, latency and power eff. between nodes required for balanced performance in hybrid systems
  - [5] Well-defined end-to-end error detection/correction
  - [6] Global RMA or shared address space across nodes
- Notes from Discussion
  - Interfaces to CPU, Memory, Interconnect
  - Work to focus on eng. studies / risk reduction
- Relations to other TCs
  - Improve performance
  - Improve programmability
- Related Projects
  - PCI-e Gen3
  - Larabee
  - Grand Fusion
  - QPI / HT
  - Networking research projects (photonics, etc)

Urgency	Duration	Responsive	Applicability	Timeline
Critical	Long	High	Broad	Soon

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

Accelerator Design

- Description
  - [1] Memory architecture (coherent plus hooks, increased addressable memory)
  - Latency hiding techniques
  - [2] Scalar performance
  - [3] Flexible synchronization
  - Improved thread scheduling
  - [5] System-level counters & status (performance & beyond)
  - Time correlation
  - User-level exception handling
  - Power/frequency scaling
  - [4] Improve fault detection and recovery
- Notes from Discussion
  - Homogeneity v. Heterogeneity
- Relations to other TCs
  - Enhance performance
  - Simplify programming
- Related Projects
  - Nvidia
  - ATI
  - Intel
  - IBM
  - FPGAs
  - Tilera

Urgency	Duration	Responsive	Applicability	Timeline
Critical	Long	High	Broad	Soon

SYSTEM SOFTWARE

- Description
  - [3] Scalable, Heterogeneous-aware OS
  - [4] Flexible management, monitoring, and scheduling of heterogeneous resources
  - [2] Optimization of data locality
  - [1] Direct data transfer (network, I/O, etc.) to/from accelerator
  - Scalable I/O in hybrid systems
  - [5] RAS system
- Notes from Discussion
- Relations to other TCs
  - All
- Related Projects
  - Lightweight OS projects
  - Heterogeneous-aware (e.g HyVM, Helios, Barrelfish)
  - OS/IO Function Shipping projects
  - Service Isolation projects

Urgency	Duration	Responsive	Applicability	Timeline
Critical	Long	High	Science	Immediate

System Design

- Description
  - [3] Packaging, density, scaling, cooling methods, serviceability
  - [1] Investigate proper balance of accelerators to cores, host interconnect, etc.
  - [2] Repeatable performance: eliminating sources of system variability
- Notes from Discussion
- Relations to other TCs
  - Performance
  - Programming models
- Related Projects
  - Roadrunner, Lincoln, etc.

Urgency	Duration	Responsive	Applicability	Timeline
Important	Medium	High	Broad	Immediate

SIMULATION & MODELING

- Description
  - [1] Support system level simulation using component level black box emulators and simulators (both cycle-accurate and not)
  - [3] Predictive, science based models and UQ methods for modeling and optimization of power, performance and reliability
  - [2] Validation of accuracy of simulators against real systems/ applications
- Notes from Discussion

- Relations to other TCs
  - Predictive performance models
- Related Projects
  - Ocelot (PTX emulation), SST, QEMU, MacSim, McPAT DRAMsim, Sim-Panalyzer, etc., Mambo (IBM - System Simulator), IAA (SNL, ORNL – Simulator project), BigSim, DiskSim
  - Etc.

Urgency	Duration	Responsive	Applicability	Timeline
Important	Long	High	Broad	Immediate

METRICS

- Description
  - [1] Establish common terminology
  - [3] Collect data to inform architecture evaluation and system/ accelerator design
  - [2] Quantify power efficiency, performance, and reliability
  - [4] Enhance usable instrumental in accelerators and hybrid systems
  - Examples:
    - Connectivity, BW, latency, I/O BW
    - Power efficiency, W/flop, MB/flop, Science/W (or throughput/W), Simulation Results/Joule
    - Reliability & availability of individual hybrid components and groups of components
- Notes from Discussion
- Relations to other TCs
  - Enable predictive performance models
  - Establish common terminology across TCs
- Related Projects
  - UHPC
  - HP Exascale group

Urgency	Duration	Responsive	Applicability	Timeline
Important	Long	High	Broad	Soon

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

Performance and Analysis

Adolfy Hoisie (LANL)
Jeffrey Vetter (ORNL)
Costin Iancu (LBNL)

SUMMARY OF PERFORMANCE AND ANALYSIS

- Areas of interest to this TC
  - Tools
  - Modeling
  - Code optimization
- Relation to other TCs
  - Architecture
  - Programming models
  - Apps

Performance is at the boundaries of all these areas, and spans the lifecycle/spectrum from R&D to design to implementation to optimization

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

PERFORMANCE AND ANALYSIS TOPICS

- Monitoring, observation and Analysis Tools for systems and applications
  - Memory, node, interconnect, apps
- Code optimization
  - Autotuning, compilation
- Predictive modeling
  - Optimal application-architecture mapping for hybrid
  - Application/architecture co-design
  - Methodology development (modeling of many flavors, simulation)
  - Dynamic (runtime) model-driven system/application optimization

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

Performance and Analysis Summary

Topic	Urgency	Duration	Responsive	Applicability	Timeline
Performance Instrumentation	Important	Medium	Moderate	Broad	Soon
Integrated measurements	Important	Medium	High	Science	Immediate
Tools for code optimization	Important	Long	Moderate	Broad	Soon
Predictive modeling	Critical	Long	High	Science	Immediate

PREDICTIVE MODELING

- Optimal app-arch mapping, app/arch co-design , methodology development, dynamic model-driven sys/ app optimization.
- Modeling power, reliability, performance in concert rather than independently
- Methodology development (modeling, simulation)
- “Should I port my code to hybrid? Is it worth it?”
- Representation for hybrid codes – programming. model ;
- Modeling hybrid applications - multiphysics
- Statistical techniques?
- Predict very large scale performance based on small scale measurements
- What is the measure of success for a model? (eg how precise to be useful? don’t always need more than coarse grained answer –“yes, porting is worthwhile”)
- Simulation: interoperability of simulators
- Fault modeling, prediction and detection; reliability modeling; error propagation. Focus on tools for this – what do we need, specific to accelerator based systems? How do accelerators influence reliability?
- Validation methodologies

- Relations to other TCs
  - Architecture, runtime SW, programming environment, apps
- Related Projects
  - LANL/PAL
  - ORNL
  - PMAC/SDSC
  - LBL/Roofline
  - P-Bound (ANL)
  - Rice

Urgency	Duration	Responsive	Applicability	Timeline
Critical	Long	High	Science	Immediate

INSTRUMENTATION

- Description: performance instrumentation for accelerators
- Binary / dynamic instrumentation for mixed codes
- Measuring buses
- HW (counters) & SW (system, application)
- Common interface for counters
- Memory subsystem analysis/ diagnosis
- MPI profile-like feedback at different levels (whole system, node level) about data movement
- Event tracing (clock; buffer)

- Relations to other TCs
  - Hooks into architecture and runtime system
- Related Projects
  - NVIDIA
  - PGI/TAU
  - UIUC
  - MIT
  - UC Berkeley

Urgency	Duration	Responsive	Applicability	Timeline
Important	Medium	Moderate	Broad	Soon

Integrated Measurements

- Infrastructure for migrating applications (performance portability)
- Tool perturbation
- Power consumption – sensors
- Diagnosis and attribution of root cause
- Resource contention and allocation / partitioning
- Mapping measurements to instructions or source code
- Performance variation / Noise for heterogeneous systems
- Data management & representation & volume
- Tool interoperability/composition/frameworks: hierarchy (intra- vs inter-node performance) and heterogeneity
- Scalability

- Relations to other TCs
  - Hooks into architecture and runtime system
- Related Projects
  - TAU
  - PGI
  - Dimemas

Urgency	Duration	Responsive	Applicability	Timeline
Important	Medium	High	Science	Immediate

TOOLS FOR CODE OPTIMIZATION

- Auto-tuning
- Dynamic Compilation
- "rules of thumb" "lessons learned" "Design Patterns" for hybrid devt, porting decisions ("should I port my code to GPU cluster?")
- Mixed precision: interactions with dynamic compilation; specifications for precision?
- Implications for correctness debugging – performance debugging interface
- Relations to other TCs
  - Fill this in
- Related Projects
  - Atlas/Magma
  - R-stream (Reservoir)
  - Ocelot
  - Parlab (Berkeley)
  - SCOUT
  - Rich Vuduc/GTech

Urgency	Duration	Responsive	Applicability	Timeline
Important	Long	Moderate	Broad	Soon

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY

THE HYBRID MULTICORE CONSORTIUM (HMC)

A multi-organizational partnership to support the effective development (productivity) and execution (performance) of high-end scientific codes on large-scale, accelerator based systems

http://computing.ornl.gov/HMC

Membership is open to all parties with an interest in large-scale systems based on hybrid multicore technologies

OAK RIDGE National Laboratory

BERKELEY LAB

Los Alamos NATIONAL LABORATORY