Center for Scalable Application Development Software - Overview

John Mellor-Crummey, Keith Cooper (Rice)
Peter Beckman, Ewing Lusk (ANL)
Jack Dongarra (UTK)
Bart Miller (Wisconsin)
Katherine Yelick (UCB/LBNL)
Goals

• Provide open source software systems, tools, and components that address a spectrum of needs
  – directly usable by application experts
  – provided to the CS community to enable development of other tools
• Engage directly with DOE application teams
• Target architectures of critical interest to DOE
  – Cray XT
  – Blue Gene/P
  – multicore processors in general
• Outreach
Scope of Activities

• Community engagement

• Research and development
  – system software
  – communication for partitioned global address space languages
  – math libraries for multicore
  – performance tools
  – compilers

• Open source software infrastructure
  – performance tool components
  – compilers

• Application outreach
Community Engagement

CScADS Summer Workshop Series

• Goals
  – identify challenges and open problems for leadership computing
  – brainstorm on promising approaches
  – foster collaborations between computer and application scientists
  – engage the broader community of enabling technology researchers

• Workshops to engage SciDAC and INCITE application teams
  – Leadership class machines, petascale applications, and performance
  – Scientific data analysis and visualization for petascale computing

• Workshops to foster development of enabling technologies
  – Autotuning for petascale systems
  – Performance tools for petascale computing
  – Libraries and algorithms for petascale applications
R&D: System Software

Developing open software stack for leadership computing platforms

• Focus
  – compute node operating system
  – I/O communication layer

• Benefits
  – facilitates infusion of software research into production systems
  – rapid (local) resolution of problems that might arise

• Results
  – Blue Gene/P compute node OS and I/O layer operational
  – supports BG/P for high throughput computing (HTC) as well as HPC
  – negligible performance penalty compared to IBM’s s/w stack
R&D: PGAS Communication Layer

Goals: low latency; high bandwidth; efficient collectives

- Planned SC08 release of GASNet and Berkeley UPC
  - updated Portals conduit for Cray XT3/4/5 platforms with “firehose”
  - new BG/P conduit based on low level DCMF layer
  - updated Infiniband conduit using new OpenIB/OpenFabrics verbs API
  - LAPI conduit for IBM Power uses RDMA
  - jointly supported by PModels and others

- Optimization of UPC collectives for multicore

BG/P: GASNet vs. MPI latency
(lower is better)

Autotuning collectives for Niagara2
(higher is better)
PLASMA: Parallel Linear Algebra software for Multicore Architectures

- **Objectives**
  - high utilization of each core
  - scaling to large number of cores
  - shared or distributed memory

- **Methodology**
  - DAG scheduling
  - explicit parallelism
  - implicit communication

- **Arbitrary DAG with fully dynamic scheduling**
R&D: HPCToolkit Performance Tools

Support measurement, analysis, and attribution of performance problems on petascale systems

- Partnership between
  - Performance Engineering Research Institute
  - Center for Scalable Application Development Software

- New capabilities
  - Sampling-based measurement of fully-optimized parallel codes on both Cray XT and Blue Gene systems
    - Uses on-the-fly binary analysis for stack unwinding of fully-optimized code
    - Supports different kinds of executables
      - Statically-linked: Blue Gene, Cray XT
      - Dynamically-linked: Linux
  - Strategies for pinpointing bottlenecks and quantifying inefficiencies
    - Across scalable parallel systems
    - Within multicore nodes
R&D: Performance Tool User Interfaces

**hpcviewer**
- correlates measurements with source
- provides actionable feedback
- supports scalability analysis on and between nodes with derived metrics

*(status: deployment fall 2008)*

**hpctraceviewer**
- displays temporal behavior of parallel applications
- provides hierarchical view call stack sample traces from HPCToolkit

*(status: prototype summer 2008)*
A source of inefficiency in large-scale applications is the “glue” that holds together code from different sources
- library code; code cribbed from other applications
- often different languages with different programming models

Classic compilers cannot improve this kind of code
- compiler never sees all the pieces; can’t optimize them together
- good application for runtime re-optimization

Opportunities in large-scale applications
- improve procedure calls & chains of calls (libraries, CCA)
  - runtime inlining and specialization of calls
  - runtime selection of library components

Ongoing work
- experimentation to quantify opportunities and estimate benefits
- compiler analysis for runtime estimation of benefits
- compiler analysis to support runtime optimization
Open Source: Performance Tools

Performance Tool Components

- **libmonitor**: first-party interface between performance tools and OS
  - manages process init/fork/exec/exit, thread create/init/join, signal delivery etc.
  - clients: HPCToolkit, Open|Speedshop, SciCortex

- **InstructionAPI**
  - abstract representation of instruction decode and address modes.

- **ControlFlowAPI**
  - platform independent representation of CFG, associated query routines, and extensible data structures
Open Source: Compiler Technology

- LLNL’s ROSE compiler: working with LLNL and LANL
  - adding full-featured Fortran support
  - adding support for Coarray Fortran 2.0

- LoopTool: memory hierarchy optimization of Fortran programs
  - source-to-source transformation of Fortran
  - capabilities include scalarization, loop fusion, blocking, unswitching
  - refined to ameliorate bottlenecks in S3D
GTC: simulates turbulent plasma in tokamak reactors
- 3D particle-in-cell code; 1D decomposition along toroidal direction
  - charge: deposit charge from particles to grid points
  - solve: compute the electrostatic potential and field on grid points
  - push: compute the force on each particle from nearby grid points

Grand challenge simulations require petascale systems
- microprocessor-based petascale systems are scarce resources
- efficient use requires effective use of multi-level memory hierarchies

Data locality optimization of GTC by CScADS & PERI @ Rice
- restructured program data and loops
- adaptively reorder ions at run time
  - at run time, locality degrades gradually as ions in the plasma become disordered
  - periodic particle reordering restores locality and performance

Reduces GTC shaped plasma simulation time by 21% on Cray XT
Application Engagement: S3D

- Direct numerical simulation (DNS) of turbulent combustion
  - state-of-the-art code developed at CRF/Sandia
    - PI: Jaqueline H. Chen, SNL
  - 2007/2008 INCITE awards at NCCS
  - pioneering application for 250TF system

- Identified node performance bottlenecks with HPCToolkit
  - low temporal reuse in diffusive flux calculation among others
  - unnecessary array copying at subroutine interfaces

- Improved loop nests with LoopTool’s semi-automatic transforms
Engagement: Other

- **Enabling technologies engagement**
  - APDEC: Chombo (structured AMR)
  - ITAPS/TASCS: Moab/iMESH (meshing)
  - PERI: performance tools development; Tiger teams

- **Application engagement using HPCToolkit**
  - UNEDF: MFDn (many Fermion dynamics - nuclear)
  - USQCD: Chroma (quantum chromodynamics)
  - Center for Turbulence Research: Hybrid (shock + turbulence)
  - NETL: MFiX (multiphase flow with interface exchanges)
  - Iowa State: CAM-EULAG (atmospheric modeling)
  - Gromacs (cellulosic ethanol)

- **Working with Fortran 2008 J3 standards committee on parallelism via coarrays**