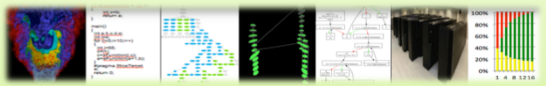
D-TEC



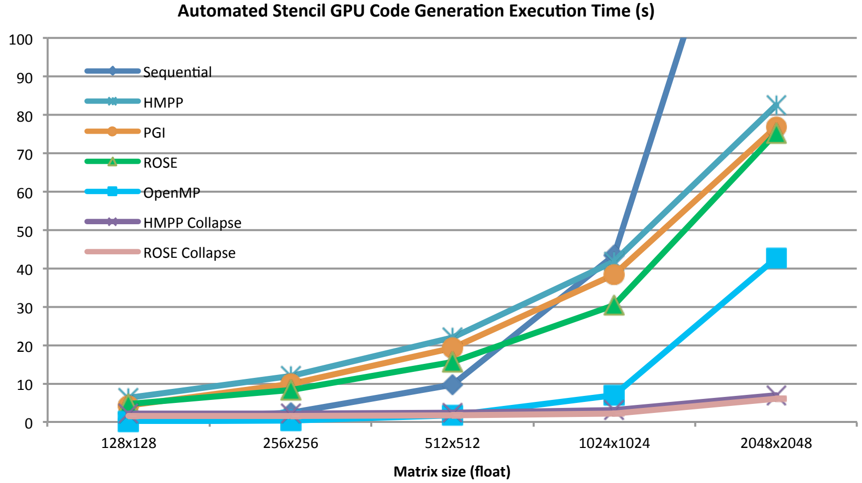
Techniques for Building Domain Specific Languages (DSLs)

Daniel J. Quinlan • Lawrence Livermore National Laboratory (Lead PI)

Co-PIs and Institutions: **Massachusetts Institute of Technology:** Saman Amarasinghe, Armando Solar-Lezama, Adam Chlipala, Srinivas Devadas, Una-May, O'Reilly, Nir Shavit, Youssef Marzouk; **Rice University:** John Mellor-Crummey & Vivek Sarkar; **IBM Watson:** Vijay Saraswat & David Grove; **Ohio State University:** P. Sadayappan & Atanas Rountev; **University of California at Berkeley:** Ras Bodik; **University of Oregon:** Craig Rasmussen; **Lawrence Berkeley National Laboratory:** Phil Colella; **University of California at San Diego:** Scott Baden.

Comparing ROSE to commercial compilers

(lower is better)



Novel Ideas

- Provide building blocks to improve the productivity of scientists and engineers:
  - Enabling custom programming models for specific problem domains
  - Tailoring applications for future architectures with compiler technology
  - Introducing automation to High Performance Computing (HPC)
- Develop an integrated solution that enables scientists to harness the power of emerging supercomputers

Impact

Simplify how to develop future applications on expected complex Exascale architectures. Domain Specific Languages (DSLs) provide tailored high-level approaches to the development of new software, and removes the burden of exploiting complex machine resources from the application developer.

DTEC makes it easier to build DSLs and have multiple DSLs cooperate.

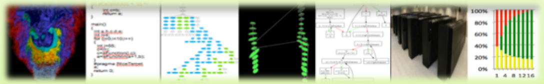
Accomplishments

- Building blocks for construction of programming models (GPU code gen, figure)
- A full Fortran 2008 SDF grammar for use in Rosebud
- Complete definition of a prototype DSL using RDL
- Definition of SDSL domain-specific language for stencil computations
- Polyhedral compiler optimization infrastructure in ROSE
- Public release of X10 2.4 (includes C++ API to APGAS runtime)
- X10 implementation of MCCK proxy app (CESAR)
- Demonstrated proxy applications: Geometric multi-grid (3D), 2D hydro method
- Stand-alone prototype of optimizing compiler for GPU code generation for SDSL
- Multi-target compiler optimization framework for stencil-based DSLs
- Initial implementations of "grammars" for Fortran, C++, and SDSL - a domain specific language for expressing common "stencil" computations within HPC
- Program analysis and source-to-source transformation support
- Compiler-based tools (e.g. auto-parallelization, support for HW simulators)

U.S. DEPARTMENT OF ENERGY | Office of Science | Lawrence Livermore National Laboratory | OREGON | RICE | IBM | MIT | BERKELEY LAB | THE OHIO STATE UNIVERSITY

September 2013

D-TEC



Techniques for Building Domain Specific Languages (DSLs)

Daniel J. Quinlan • Lawrence Livermore National Laboratory (Lead PI)

Co-PIs and Institutions: **Massachusetts Institute of Technology:** Saman Amarasinghe, Armando Solar-Lezama, Adam Chlipala, Srinivas Devadas, Una-May, O'Reilly, Nir Shavit, Youssef Marzouk; **Rice University:** John Mellor-Crummey & Vivek Sarkar; **IBM Watson:** Vijay Saraswat & David Grove; **Ohio State University:** P. Sadayappan & Atanas Rountev; **University of California at Berkeley:** Ras Bodik; **University of Oregon:** Craig Rasmussen; **Lawrence Berkeley National Laboratory:** Phil Colella; **University of California at San Diego:** Scott Baden.

Problem

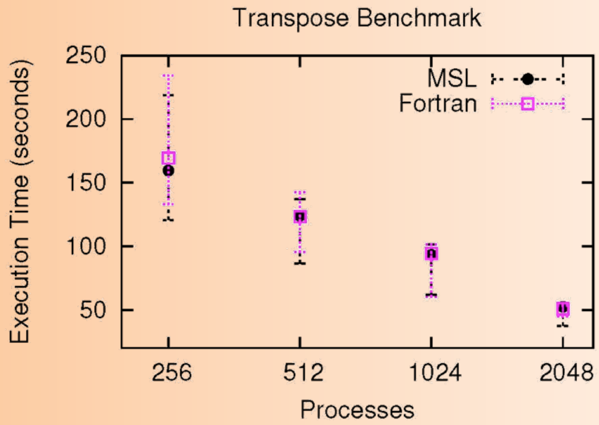
- Use of Domain Specific Languages supported by compilers and runtime systems is the most promising method for scientists to harness the full potential of next-generation Exascale supercomputers

Selected Solution Highlights

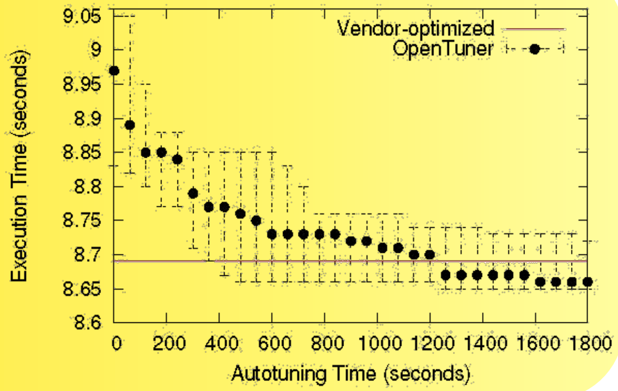
- The Sketch computer-aided programming system for refining high-level DSLs to high performance code supports parallel programming and can generate multiple variants of a program; each variant has potentially different performance
- We developed the first general and reusable auto-tuning framework, OpenTuner, that uses machine learning techniques to finding the best-performing variant of a program automatically for complex and heterogeneous systems

Recent Selected Accomplishments

- We were able for the first time to automatically synthesize the details of a distributed implementation of an irregular computational kernel and check it for the absence of bugs
- The resulting implementation scales as well as a hand crafted implementation at up to 2500 cores
- The Sketch generated version of Transpose from the FT NAS parallel benchmark is faster than the standard hand-parallelized FORTRAN version



- OpenTuner currently includes 22 machine learning techniques that are selected using a bandit meta technique, and thus is able to adopt to optimizing the performance of any new problem
- OpenTuner is able to outperform vendor hand-optimized high performance Linpack (HPL) benchmark after 150 iterations of learning



- OpenTuner and Sketch were released to the public and are now being used by many projects to automatically find best implementations of a program

U.S. DEPARTMENT OF ENERGY | Office of Science

Lawrence Livermore National Laboratory

OREGON

RICE

IBM

MIT

BERKELEY LAB

THE OHIO STATE UNIVERSITY

September 2013