HPC Software Initiatives at Rice

Jan E. Odegard
Executive Director, CITI
Rice University
odegard@rice.edu

Center for Scalable Application Development Software (CScADS)

Overview

- Rice University (Lead Institution)
  - —Ken Kennedy, Lead PI
  - —John Mellor-Crummey
  - —Keith Cooper
- Argonne National Laboratory
  - —Peter Beckman, Lead Site PI
  - —William Gropp
  - —Ewing Lusk
- University of California, Berkeley
  - —Katherine Yelick
- University of Tennessee, Knoxville
  - —Jack Dongarra
- University of Wisconsin, Madison
  - —Barton Miller

Goals

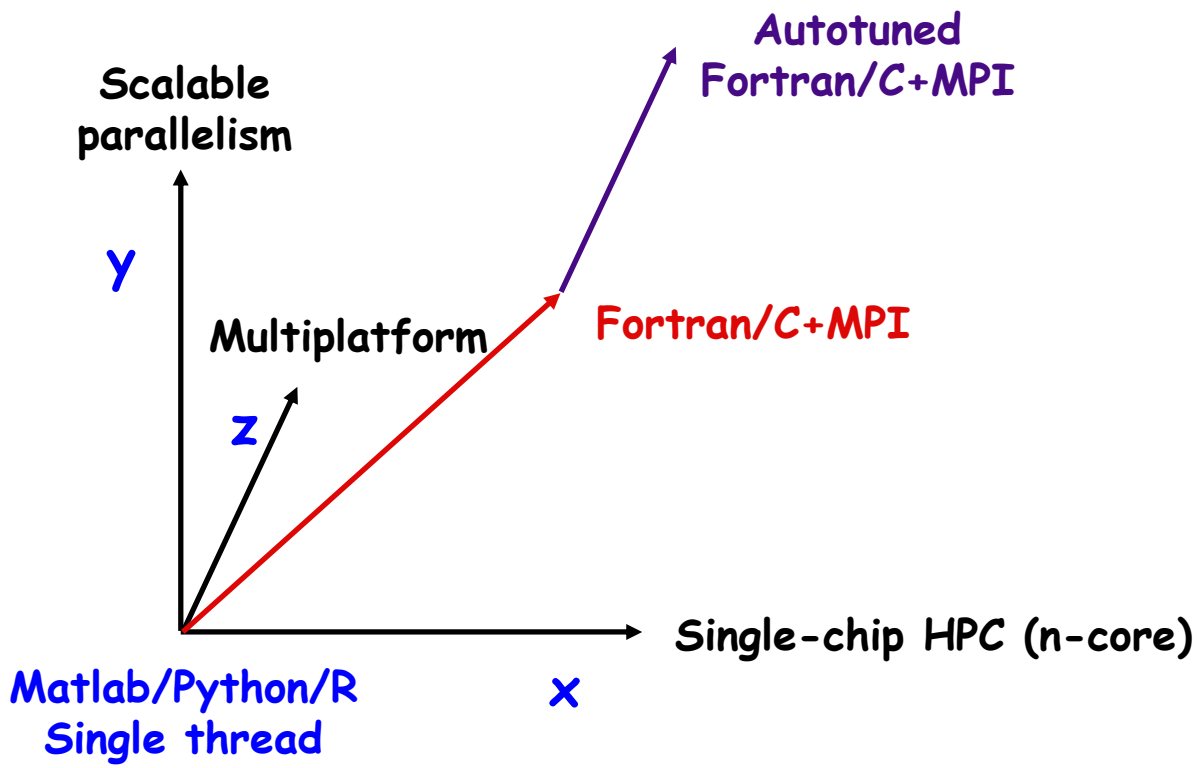
- To conduct research leading to the design and construction of software tools and systems to facilitate scalability of applications to the petascale and beyond
  - Special emphasis on DOE Leadership Class Facilities (LCFs) and parallel systems composed of multicore processors
- To catalyze activities within the computer science community that will lead to visionary new ideas for application development support software
  - Emphasis on interactions with systems vendors, application developers, and library designers
- To foster the development of new tools by the computer science community through support of common software infrastructures and standards

Scalability Dimensions

- Scaling from Domain Language to Fast Uniprocessor Application
  - Example: Translate Matlab to C or Fortran
    - Constraint-based type analysis: determine array sizes and types
    - Replace calls to domain library with calls to preoptimized versions, specialized to parameter type signature
    - Scalarize array statements
  - Example: Source-to-source performance optimization
    - Inlining, loop fusion, tiling for multiple levels of cache
- Scaling from One to Many Processors
  - Example: Matlab with data distributions
    - Direct translation to Fortran+MPI via HPF compilation strategy
  - Example: Tools for performance understanding
    - HPCToolkit reveals where parallel efficiency is lost
- Scaling to Many Platforms
  - Example: Automatic tuning to specific machine's architecture

Scalable Application Development

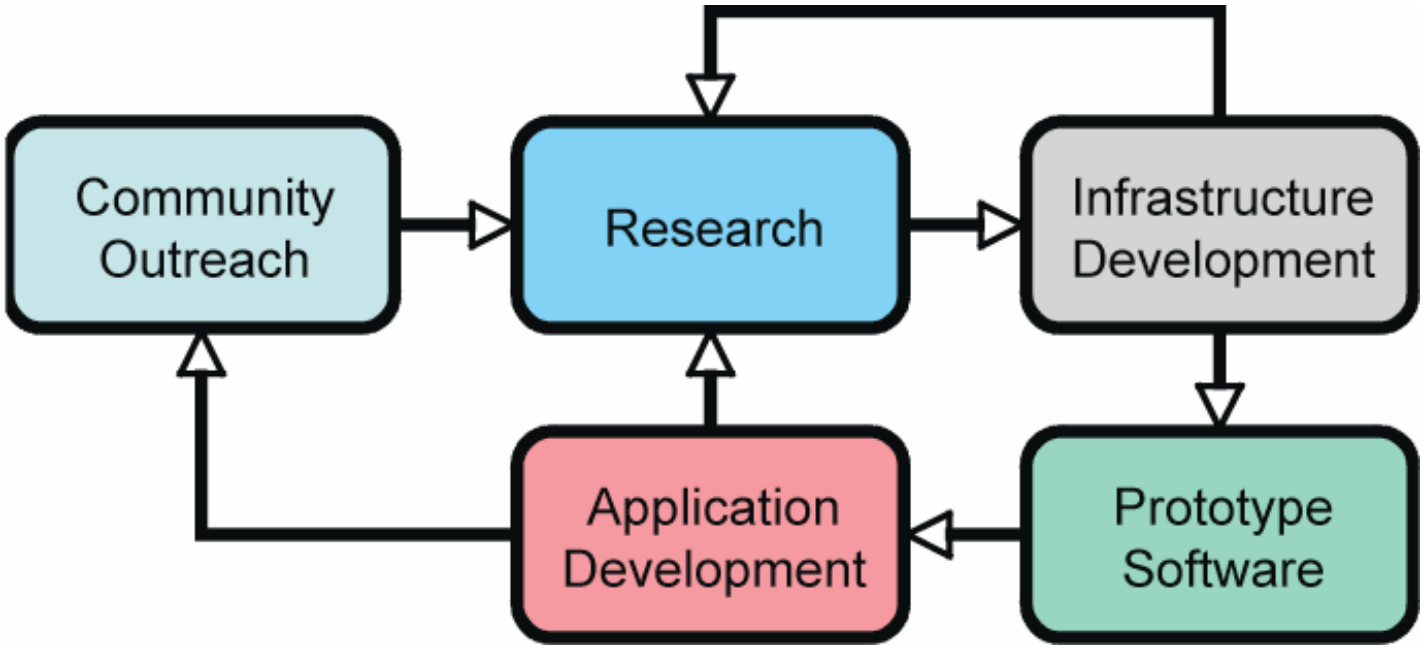
- Scalable from high-level to high-performance
- Scalable from uniprocessor (or multicore) to petascale
- Scalable from single platform to multiple platforms



CScADS Model

- Research
  - Application-driven software systems research
- Community vision building
  - Summer institutes on focused topics of relevance to scalability
- Open-source software development and integration
  - Software infrastructures: Open64, ROSE (collab with LLNL), Telescoping Languages, D System, HPCToolkit
- Collaborations
  - DOE
    - Labs: Oak Ridge, Argonne, LBNL (+ PNL, LANL, LLNL, SNL)
    - Application, library, and system software developers
    - Other Centers and Institutes: e.g., PERI, APDEC, TASCS
  - NSF
    - TeraGrid, NSF Petascale Track 1 and 2
  - Industry
    - Vendors of systems and software

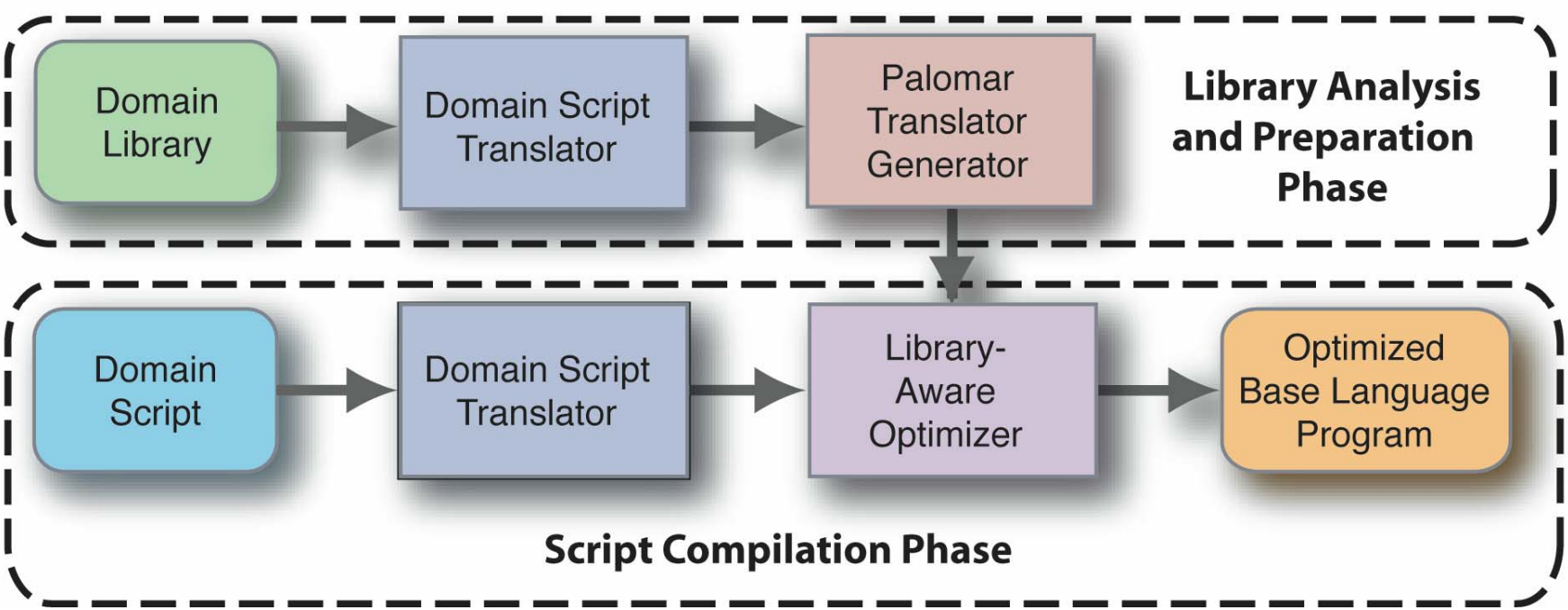
Interrelationships Among Activities



Research

- Compilers, libraries, performance analysis tools, and tuning systems for new architectures
  - Key focus: Scalable systems based on multicore chips
    - With heterogeneous components
  - Automatic tuning multicore chips and scalable systems
- Programming models, languages and compilers for scalable parallel computing
  - Co-Array Fortran, UPC, Titanium, HPCS languages
  - Extensions to standard languages (C, C++, Fortran)
  - High-level domain languages based on scripting languages
    - Example: Matlab (with parallelism) plus domain libraries
- High performance component integration frameworks
  - Application of telescoping languages
    - Trade off flexibility for performance

Domain Language Support



Plan

- First two years:
  - Concentrate on tools to support scalable application development for DOE Petascale Facilities (ORNL, ANL, LBNL)
    - Improvement of node performance
    - Efficiency of scaling
    - Research and development driven by critical applications
  - Summer Institutes
    - Familiarize community with challenges of DOE petascale systems
    - Interface workshops between tool and application developers
- Later years:
  - Increased focus on application developer productivity
    - High productivity languages, scalable anomaly detection, automatic tuning
  - Seed research infrastructures to the CS community to accelerate research

Research Supported by

Scientific Discovery through Advanced Computing

ST-HEC

LACSI