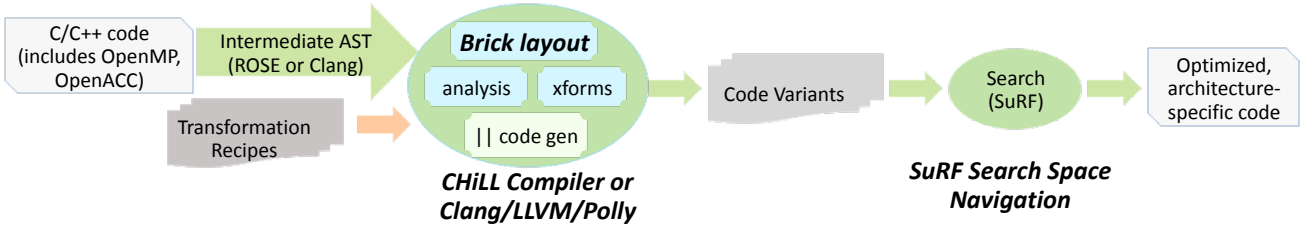
Autotuning Interface and its Application Pragma Autotuning of OpenMP and Loop Transformations [2.3.2.10 PROTEAS-TUNE]

EXASCALE COMPUTING PROJECT

Mary Hall, Vinu Sreenivasan (University of Utah)
Prasanna Balaprakash, Hal Finkel, Paul Hovland, Michael Kruse, Xingfu Wu (Argonne)

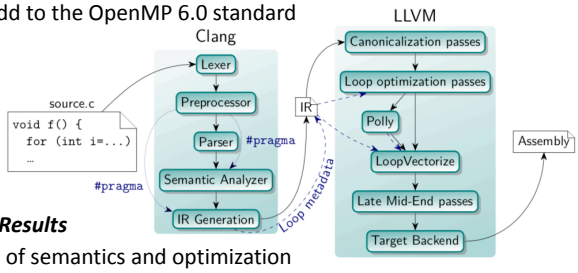
Overview



LLVM Extensions

Loop transformations with Polly, Clang & LLVM

- Improve support for loop transformations in Clang/LLVM
  - Add more transformations
  - Composable transformations
  - Interaction with parallelism
- Controllable using in-source #pragma annotations
  - `e.g. #pragma clang loop transform sizes(32,16)`
- Using Polly in the implementation
  - Code modifications become tree transformations
- Effort to add to the OpenMP 6.0 standard



Benefits and Results

- Separation of semantics and optimization
- Use case *dgemm*: Already reached 42% of theoretical peak performance with tiling, interchange and array packing.

Contact: Michael Kruse, michael.kruse@anl.gov

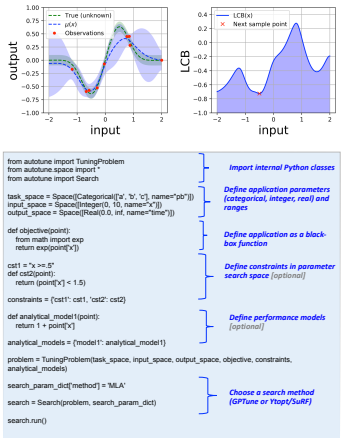
SuRF / Autotuning

SuRF: Search Using Random Forests for autotuning search

- Asynchronous search method with manager-workers approach
- Random forest as a surrogate model to navigate the search space
- Bayesian methods for exploration and exploitation of the search history
- Interface for expressing wide range of problems, constraints and performance models (with xSDK4ECP)

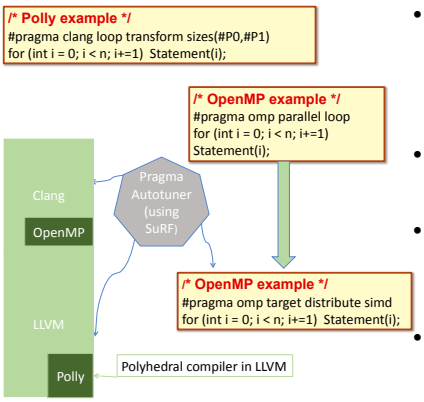
Benefits and Results

- Finds high-performing parameter configurations in short wall-clock time
- Demonstrated effectiveness on superlu library and qmcpack application autotuning
- Scaled up to 64 worker nodes to reduce the search time significantly
- Interface allows to prune the search space
  - user defined constraints
  - performance models
  - other search methods



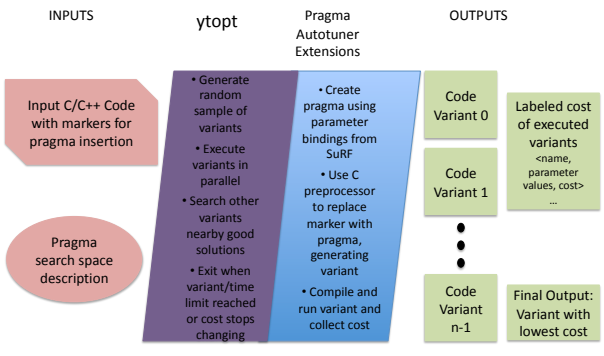
Contact: Prasanna Balaprakash, pbalapra@anl.gov

Pragma Autotuning



Benefits and Impact

- Employs autotuning to achieve performance portability (CPU and GPU) of compiler transformations and OpenMP
- Replace descriptive OpenMP pragmas with prescriptive ones
- Vary Clang loop transformation pragmas, profitable settings for compiler and loop level parameters
- ***Improve application programmer productivity!***



Contact: Mary Hall, mhall@cs.utah.edu