Enhancing Learning-based Autotuning With Composite and Diagnostic Feature Vectors

Saami Rahman
Texas State University
San Marcos, TX
saami.rahman@txstate.edu

Richard Hay
Texas State University
San Marcos, TX
rh1357@txstate.edu

Mario A. Gutierrez
Texas State University
San Marcos, TX
mag262@txstate.edu

Apan Qasem
Texas State University
San Marcos, TX
apan@txstate.edu

The use of machine learning techniques has emerged as a promising strategy for autotuning. Central to the success of such learned heuristics is the construction of a feature vector that accurately captures program behavior. Although the salient features of certain domain specific kernels are well understood, at least in principle, automatically deriving suitable features for general numerical applications remains a significant challenge. This poster presents a learning-based autotuning system that introduces two new classes of features that encapsulate key architecture-sensitive performance characteristics for a broad range of scientific applications. The first class of features are high-level, composite and derived from compiler models for data locality and parallelization. The second class of features is a series of synthesized and normalized HW performance counter values that diagnose causes of program inefficiencies.

Composite Features

Derived from compiler models, program AST and execution profiling

Covers key performance imperative including data locality, parallelism and use of specialized hardware units

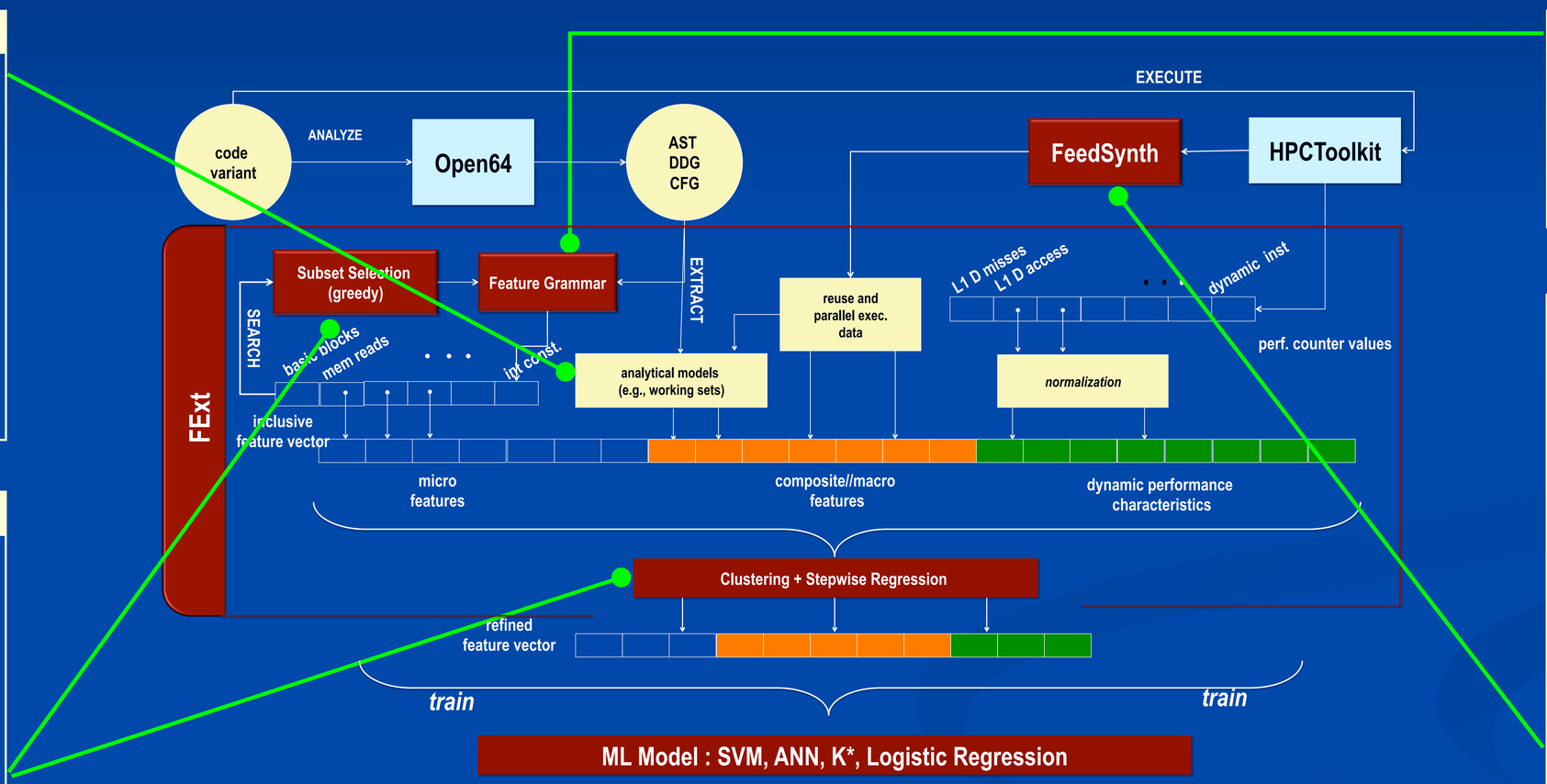
Example include reuse distance, stride information, thread granularity, and prefetch candidacy

Feature Set Pruning

Subset selection with greedy heuristic to gather elementary features with highest ranking

Clustering to eliminate strongly correlated and redundant features

Stepwise regression to filter out irrelevant features



Feature Grammar

Automatically derived from program's internal representation

Extracts source-level elementary features

Diagnostic Dynamic Features

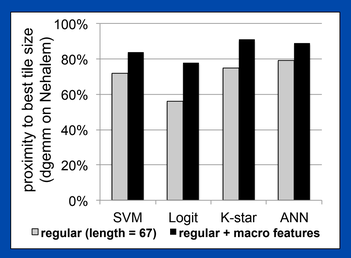
Synthesis of raw performance metrics to produce meaningful performance indicators

- Cache miss classification (compulsory, capacity, conflict)
- Hardware prefetch activity
- Cache coherence and sharing : true and false sharing, replication, pollution
- Synchronization cost

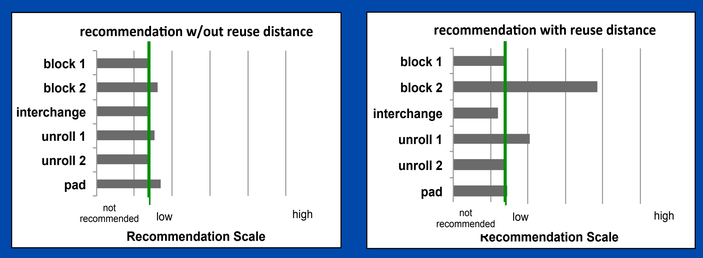
Identify causes rather than symptoms of performance issues

Experimental Results

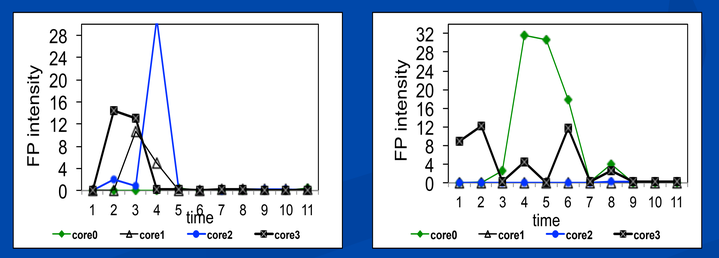
Inclusion of working set in feature Vector and its impact on ML effectiveness



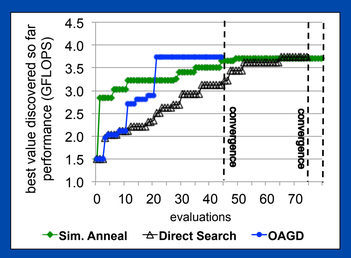
Differences in recommendation with and without composite features in feature vector



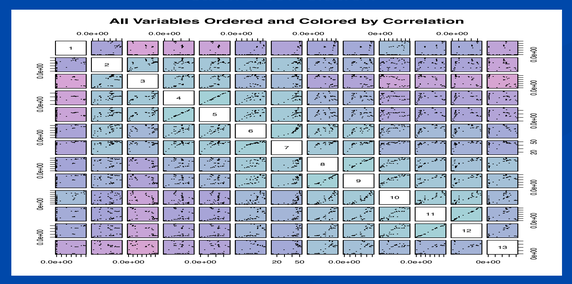
FP-intensity as a feature and its relationship to thread affinity configurations



Conflict misses and register spills as dynamic features and Its impact on heuristic search



Feature correlation



Conclusions

High-level composite features pertaining to data locality and parallelism combined with diagnostic dynamic performance metrics can enhance the predictive capabilities of learning algorithms

This research funded by NSF award CNS 1253292

We thank John Mellor-Crummey and Nathan Tallent for their help with the HPCToolkit interface