Automatically Generating Performance Imperatives via Machine Learning

Biplab Kumar Saha
Texas State University
San Marcos, TX
b_s183@txstate.edu

Tiffany Connors
Texas State University
San Marcos, TX
tac117@txstate.edu

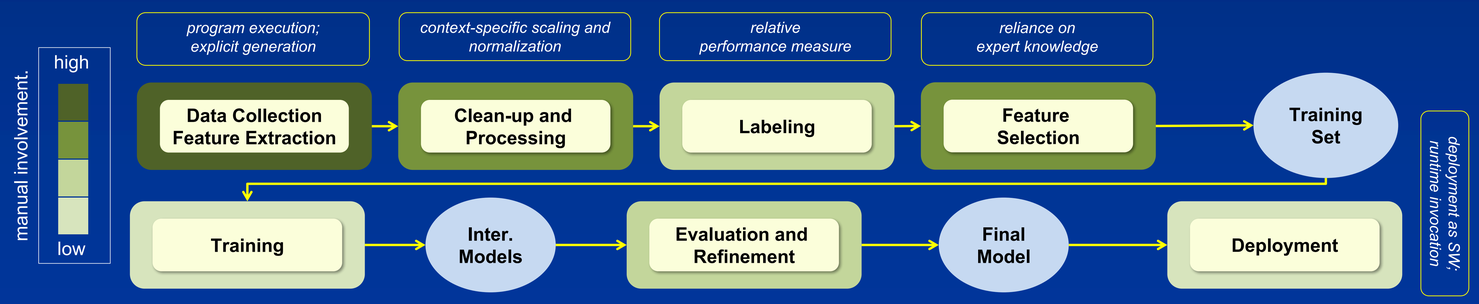
Saami Rahman
Texas State University
San Marcos, TX
saami.rahman@txstate.edu

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

Recent interest in machine-learning based methods have produced many sophisticated models for performance modeling and optimization. These models tend to be sensitive to architectural parameters and are most effective when trained on the target platform. Training of these models, however, is a fairly involved process and requires knowledge of statistics and machine learning that the end users of such models may not possess. This poster presents a framework for automatically generating machine-learning based performance models. Leveraging existing open-source software, we develop a tool-chain that provides automated mechanisms for sample generation, dynamic feature extraction and selection, data labeling, validation and model selection. The resulting models yield high prediction accuracy and can deliver improved performance and energy efficiency in many different contexts. We present summary experimental results that demonstrate the utility and efficiency of the system.

Conventional workflow for supervised learning and the specialized tasks required for building performance models

Degree of practitioner involvement highlighted in each stage



Survey of 112 papers on performance modeling and tuning via machine learning (MLMT) 2006-2016

Scores assigned based on work described in studied paper only

MLSPEC

Simple language interface allows practitioners to provide a structured description of the MLMT model to be built

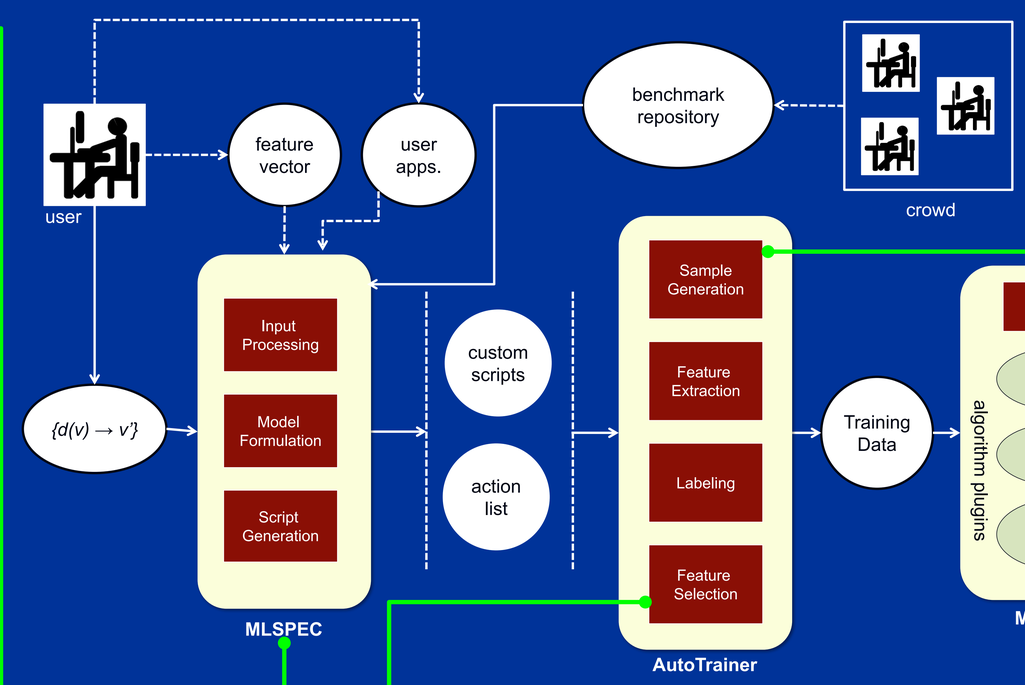
MLSPEC parser
- derives build and/or execute commands from model specification
- generates custom scripts to create sample instances and produce labelled training data

Diagnostic 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



Automatic Sample Generation

MLBENCH automatically generates synthetic micro-kernels with varying performance properties

Capable of controlling arithmetic intensity, memory divergence, ILP, data access patterns task granularity and FP-precision

Support CPU, GPU and hybrid kernels

Generated benchmark can be combined with applications from the benchmark repository to create sample instances the provide good coverage of feature space

Can simulate extreme performance behavior for automatic labeling

Validation and Visualization

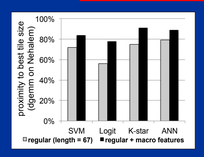
Automatic validation, evaluation and hyper parameter tuning

Analysis and visualization packages provide insight into both the feature space and the generated model

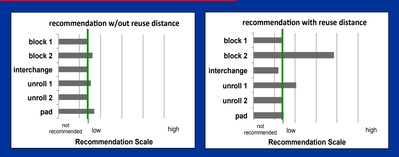
Clustering, PCA, decision tree

Experimental Results

Use of diagnostic and composite features

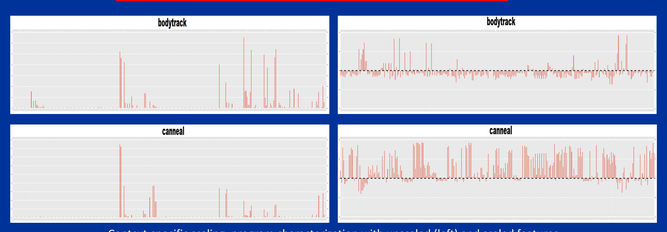


Inclusion of working set in feature vector and its impact on learning effectiveness

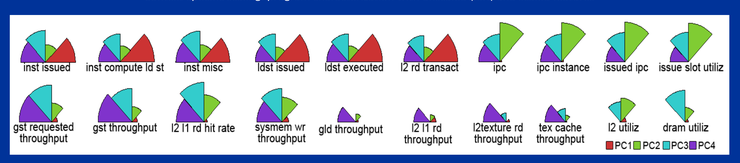


Differences in recommendation with and without composite features in feature vector

Exploring the feature space

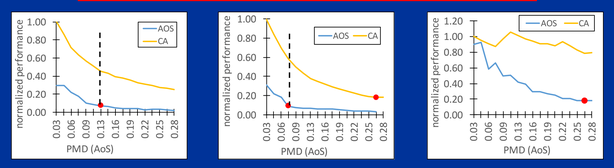


Context-specific scaling: program characterization with unscaled (left) and scaled features

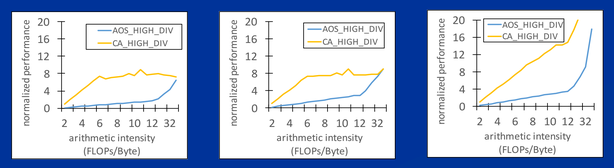


Visualizing the feature space: PCA Varimax Rotation diagrams for feature space containing GPU performance events

Bottleneck classification and performance insight

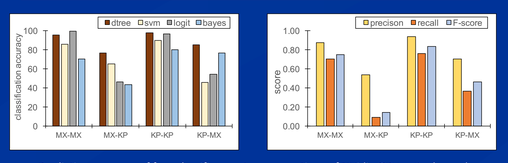


Pinpointing bandwidth saturation points for GPU kernels divergent memory access: (a) AMD 7850, (b) AMD8500 and (c) AMD Radeon Nano (CGO17)



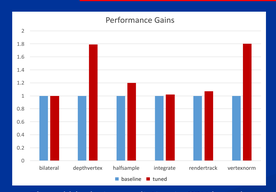
Identifying performance patterns in the feature space of arithmetic intensity: (a) AMD 7850, (b) AMD8500 and (c) AMD Radeon Nano (CGO17)

Cross-platform performance prediction

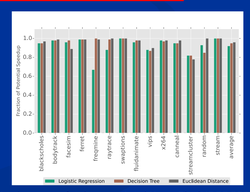


Prediction accuracy of four classifiers on two generations of NVidia GPUs : Kepler and Maxwell. X-Y indicates model trained on X and invoked on Y

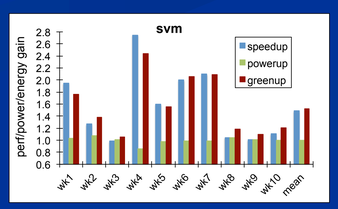
Performance and energy gains



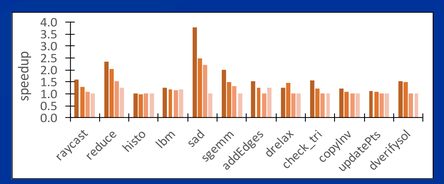
Thread block size prediction in GPU kernels (SC17, under review)



Hardware prefetching on Intel Xeon Phi (HPCC 2015)



Thread affinity prediction for multithreaded CPU workloads (IGSC15)



Register cap prediction for GPU kernels (under review)

What we have learned …

Can generate ML performance models with minimal user intervention
- Compiler optimizations sequence, specialized code transformations
- CPU, GPU and heterogeneous platforms

Auto-generated models exhibit comparable accuracy; can boost performance and improve energy efficiency

Models provide new insight about known performance problems

Acknowledgements

This research funded by NSF award CNS 1253292

Work on data layout was done while Apan Qasem was a Visiting Scholar at AMD Research

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