Modeling Cross-Architecture Co-tenancy Performance Interference (CCGrid ’15)

Methodology:
1. Profile to collect sensitivity curves and pressures for a set of benchmark programs, \( x \in \{1, 2, \ldots, N \} \), on multiple architectures with different hardware configurations, \( n \in \{1, 2, \ldots, M \} \).
2. Create models for each program and architecture, \( f_{x,n} \), by regression analysis.
3. Create cross-architecture pressure models, \( h_{x,n} \), describing the relationship between pressures.
4. Given new program \( x \) on HW1, predict sensitivity functions and pressure on a hardware configuration HW2, \( \beta_{x} = h_{x}(P_{x}) \).
5. Given new program \( y \) on HW1, predict sensitivity functions and pressure on a hardware configuration HW2, \( \beta_{y} = h_{y}(P_{y}) \).

Results:
- Prediction error (%) on pairwise co-run benchmarks (core2 duo to i5):
  - Nehalem
  - Intel Core i5
  - Sandy Bridge
  - Intel Core i7
  - Kaby Lake

Conclusions:
- In general, predicted performance degradation is less than 2%.
- A few benchmarks are problematic with higher error, e.g., soplex.

Selective Switching Mechanism in VMs via Support Vector Machines and Transfer Learning (ML ’15)

Goal:
- Learn a decision model to predict switching mode between shadow pages (SP) and hardware assisted paging (HAP).
- Transfer model to new hardware with limited profiling required.

Methodology:
1. Collect training data from running SPEC CPU2006 INT benchmarks in HAP and SP mode; multiple training samples are collected running in one phase alone or switching modes for each phase of a benchmark.
2. A linear SVM model was used as the decision function for the dynamic switching mechanism:
   \[ 2.11TLB - 0.95SF - 24.79 = 0 \]
   This linear model was encoded in the XEN hypervisor to select switching modes.
3. A transfer learning algorithm TrAdaBoost is used to learn a model for a new architecture (AMD) using only a limited amount of new profile data combined with the original data from an Intel architecture.

Intelligent Sampling of Hardware Performance Counters for Performance Prediction

Overview:
- Present a method for fine-grained performance modeling through strategic sampling of available architecture events. Attribute Efficient Regression (AER)
  - Constrained by number of performance monitor counters available; limits number of events-hardware performance counters (HPCs) to observe
  - Assumes no a priori knowledge of event statistics
- AER’s regression coefficients, \( w \), are used to rank and select events for individual benchmarks or a common set across benchmarks
- With the set of selected events a model to predict performance is learned via least squares:

Results:
- Across three architectures, the rankings of events is mostly consistent (spearman rank correlation coefficient > 0.65)
- Prediction performance error decreases with the number of events sampled (Top k vs. Top 2k)
- In general, error decreases when selected events are restricted to a single element of each class (Top k, Unique); exceptions include Nehalem, mf and soplex
- Inclusion of higher-order relationships (pair-wise relationships – Top k, Pairwise), has the lowest error across architectures and benchmarks

References

Collaborators
- Michigan Tech: Wei Kuang, Jason Hebel
- Western Michigan University: Steven Carr