Beyond Profiling

Chris Quackenbush
Google
cquackenbush@gmail.com

Mohamed Zahran
Computer Science Dept.
NYU
mzahran@cs.nyu.edu
Profiling

Program → Execution → Output

Profile Data

- Hardware optimization
- Hardware reconfiguration
- ....

- Feedback-directed optimization
- Code optimization
- ....
Hardware/Software Interaction

- Start Execution
  - No predictions
  - Gather patterns
- End Execution
  - Predict
  - Update patterns

Training

Inference

Examples: branch prediction, data value prediction, cache replacement, cache power management, ...
Hypothesis

It is all about patterns.
If we can learn the patterns, we can get useful info for ANY program and from ANY program.
Implementation

- List traversal
- Matrix transpose
- Array multiplication
- Quicksort
Average LLC size for PARSEC benchmarks when executed with each algorithm

- none (full cache)
- EWSS
- BLOOM
- ANN

Benchmark: blackscholes, canneal, dedup, facesim, ferret, fluidanimate, raytrace, swaptions, vips, x264
LLC miss rate for PARSEC benchmarks when executed with each algorithm

- blackscholes
- canneal
- dedup
- facesim
- ferret
- fluidanimate
- raytrace
- swaptions
- vips
- x264

Legend:
- none (full cache)
- EWSS
- BLOOM
- ANN
Roadmap

Continuous learning and re-learning

Branch prediction with offline data (Spectre?)

Automatically generating patterns

More test cases

More sensitivity analysis and signature formats
Conclusions

• The hardware (re)configuration depends on patterns.
• We can extra these patterns from any profile data.
• The more profile data we have the more patterns we learn.
Questions!

Chris Quackenbush
Google
cquackenbush@gmail.com

Mohamed Zahran
mzahran@cs.nyu.edu
http://www.mzahran.com