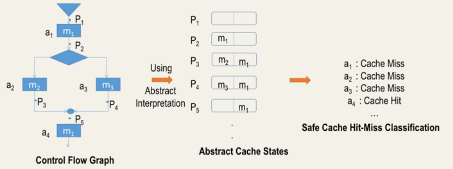
Precise Analysis of Private and Shared Caches for tight WCET estimates

Kartik Nagar, Y N Srikant

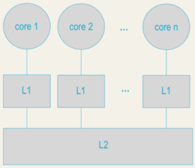
The WCET Problem

- WCET Problem : To determine W such that W ≥ Actual Execution Time of P on A
- Primarily in Real time systems, to prove all deadlines of tasks are always met.
- Execution time of a program depends heavily on the cache behavior of its memory accesses
- Precise prediction of **cache behavior** is necessary for obtaining tight WCET estimates.

Standard approach to cache analysis



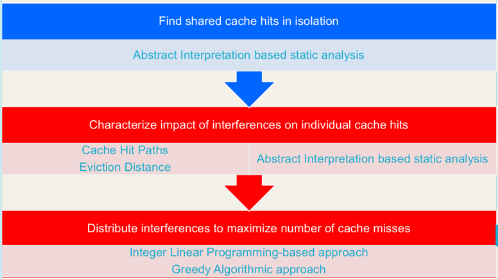
Shared Cache Analysis



- Given assignment of programs to cores, find the shared cache behavior of each program
- Primary Issue : Shared cache accesses made by other cores can evict cache blocks of program under analysis and cause extra cache misses.
- Due to interfering accesses, it is almost impossible to guarantee that an access will always hit the cache.

Worst Case Interference Placement

- Instead of classifying individual accesses as hit or miss, we try to find lower bounds on the number of cache hits, given the number of interfering accesses coming from other cores.
- To do so, we find the worst-case distribution of interferences in the program, which can cause the maximum number of shared cache misses.



Find shared cache hits in isolation

Abstract Interpretation based static analysis

Characterize impact of interferences on individual cache hits

Cache Hit Paths
Eviction Distance

Abstract Interpretation based static analysis

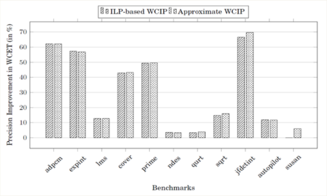
Distribute interferences to maximize number of cache misses

Integer Linear Programming-based approach
Greedy Algorithmic approach

Properties of Our Approach

- We break down WCIP into two inter-dependent problems:
  1. Finding the worst-case path in the program in the presence of interferences
  2. Finding the distribution of interferences on a program path which causes the maximum number of shared cache misses
- We show that problem 1 is NP-Hard by reduction from the 0-1 Knapsack problem.
- For problem 2, we propose an algorithmic greedy approach which chooses cache hits for interference distribution based on increasing order of eviction distance
  - This guarantees that the increase in WCET due to interferences would be linear in the number of interferences.
- The ILP-based approach encodes both problems in a single ILP, and directly provides the WCET of a program as the value of the objective function.
- The approximate algorithmic approach makes two assumptions to make the problem tractable, and provides a WCET estimate in polynomial time.

Experimental Results

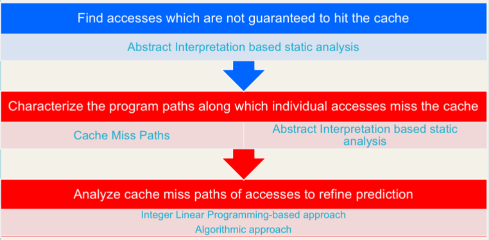


- 2-core architecture, with 4 KB Shared Cache.
- WCIP provides average precision improvement of 26 % over previous approach

Private Cache Analysis

- No interferences, but due to multiple program paths, an access may hit or miss the cache in different execution instances.
- The state-of-the-art approach classifies an access as a cache hit only if it is guaranteed to hit the cache across all execution instances.
- Cache hit-miss prediction can be refined in several ways
  1. Two accesses may never miss the cache together in the same execution instance.
  2. An access inside a loop may not miss the cache in all iterations.
  3. An access may not miss the cache in the worst-case execution instance.

Our Approach



Find accesses which are not guaranteed to hit the cache

Abstract Interpretation based static analysis

Characterize the program paths along which individual accesses miss the cache

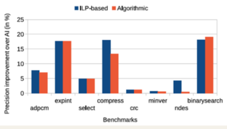
Cache Miss Paths

Abstract Interpretation based static analysis

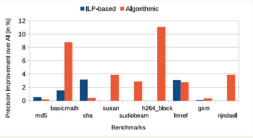
Analyze cache miss paths of accesses to refine prediction

Integer Linear Programming-based approach
Algorithmic approach

Experimental Results



Small Benchmarks (Code size ~ 10 KB)



Large Benchmarks (Code size ~ 100 KB)

- Average precision improvement of our approach over previous approaches – 8 %.
- Algorithmic approach matches the precision improvement of ILP-based approach, and also scales better for large programs.

Conclusion

- In this thesis, we have proposed precise, scalable approaches to cache analysis aimed towards tighter estimation of WCET.
- Shared cache analysis in multi-cores
  - Our approach, called Worst Case Interference Placement, is significantly precise than previous approaches.
- Private cache analysis
  - Reasonable precision improvement over previous approaches with a moderate increase in analysis time.

Publications

1. Precise shared cache analysis using optimal interference placement. Kartik Nagar and Y.N. Srikant. 20th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS), 2014.
2. Path sensitive cache analysis using cache miss paths. Kartik Nagar and Y.N. Srikant. 16th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2015.
3. Fast and Precise Worst Case Interference Placement for Shared Cache Analysis. Kartik Nagar and Y.N. Srikant. Accepted in ACM Transactions on Embedded Computing Systems (TECS), 2015.