Roadmap

Background

Write-back Analysis

  Eviction-focussed Analysis
  Store-focussed Analysis

Evaluation

Conclusion and Future Work
Write-back Caches

Memory Accesses: 0

Write through

Memory Accesses: 0

Write back
Write-back Caches

Memory Accesses: 1

Write through

Memory Accesses: 0

Write back
Write-back Caches

Memory

2

Cache

CPU

Memory Accesses: 1

Write through

Memory

2

Cache

CPU

Memory Accesses: 1

Write back
Write-back Caches

Memory Accesses: 1

Write through

Memory Accesses: 0

Write back
Write-back Caches

Memory Accesses: 2

Write through

Memory Accesses: 0

Write back
Write-back Caches

Memory Accesses: 2

Write through

Memory Accesses: 1

Write back
Motivation and Challenge

The Good: write back delivers higher performance

The Ugly: write back requires more sophisticated analysis

“for real systems, write-through seems to always result in lower worst-case estimates”  
Wilhelm et al. (2010)
Motivation and Challenge

The Good: write back delivers higher performance

The Ugly: write back requires more sophisticated analysis

“for real systems, write-through seems to always result in lower worst-case estimates”

Wilhelm et al. (2010)
Major Events in the Life of a Cache Block

Eviction-Focused Analysis
Eviction causes write back

Store-Focused Analysis
store causes write back later
Major Events in the Life of a Cache Block

**Eviction-Focused Analysis**
Eviction causes write back

**Caching**
- Load
- Load
- Store

**Eviction**
- dirty

**Store-Focused Analysis**
store causes write back later
Major Events in the Life of a Cache Block

Eviction-Focused Analysis
Eviction causes write back

Store-Focused Analysis
store causes write back later
Major Events in the Life of a Cache Block

**Eviction-Focused Analysis**
Eviction causes write back if there was a store

**Store-Focused Analysis**
store causes write back later
Major Events in the Life of a Cache Block

Eviction-Focused Analysis
Eviction causes write back if there was a store

Store-Focused Analysis
store causes write back later
Major Events in the Life of a Cache Block

Eviction-Focused Analysis
Eviction causes write back if there was a store

Caching

Load
Load
Store
Store
Store
Eviction

dirty

Store-Focused Analysis
store causes write back later
Major Events in the Life of a Cache Block

**Eviction-Focused Analysis**
Eviction causes write back if there was a store

**Store-Focused Analysis**
*Dirtifying* store causes write back later
Least-Recently-Used Replacement (LRU)
Example

\[x = f(x)\]
\[\text{sum} = 0\]
repeat up to \(N\) times
\[\text{sum} += \text{arr}[\text{read\_sensor()}]\]
Example

\[ x = f(x) \]
\[ \text{sum} = 0 \]
repeat up to \( N \) times
\[ \text{sum} += \text{arr}[\text{read\_sensor()}] \]
may evict \( x \) in \textit{any} iteration
Example

\[ x = f(x) \]
\[ \text{sum} = 0 \]
repeat up to \( N \) times
\[ \text{sum} += \text{arr[read}\_\text{sensor()}\text{]} \]
may evict \( x \) in \textit{any} iteration
int arr[M]
repeat up to \( N \) times
\[\text{arr[read\_sensor()]}++\]

Assume the array is smaller than the cache

\[\implies \text{at most } M \text{ dirtifying stores, independently of } N\]
Persistence Analysis

```c
int arr[M]
repeat up to N times
    arr[read_sensor()]++
```

Assume the array is smaller than the cache

⇒ at most $M$ dirtifying stores, independently of $N$
Evaluation

1. Write back vs write through
2. Precision of write-back analysis
3. Store-focussed analysis vs eviction-focussed analysis
Evaluation setup

Benchmarks

- Mälardalen benchmark suite
- 5 SCADE benchmarks

Processor

- ARM(ish) processor with five-stage, in-order pipeline
- Single-level data cache
Write back yields lower bounds than write through

Normalized WCET Bounds \(\frac{WCET_{\text{WB}}}{WCET_{\text{WT}}}\)
WCET bounds are close to the theoretical lower bound.
Eviction-focussed approach is mostly orthogonal to store-focussed approach.
Most tasks spend little time accessing memory
Eviction-focussed approach is mostly orthogonal to store-focussed approach
Future Work

- Larger benchmarks
- Integrate into response-time analysis (Davis, Altmeyer, Reineke, RTNS 2016)
- Other replacement policies (PLRU, FIFO)
Conclusion

- Write-back caches are desirable for hard real-time systems
- Write-back analysis should focus on both, evictions and stores
Conclusion

- Write-back caches *are* desirable for hard real-time systems
- Write-back analysis should focus on both, evictions *and* stores

Thank you very much for your attention!