Analytically Bounding Data Cache Behavior for Real-Time Systems

Harini Ramaprasad and Frank Mueller

1 Motivation
- Real-time systems have temporal constraints
- A-priori schedulability guarantee required
- Schedulability analysis of real-time systems
- Worst-case execution time (WCET) of tasks required
- Static timing analysis (STA) used for this
- Data Caches (DS)
  - Difficult to characterize \(\rightarrow\) unpredictability in STA
  - Add to pessimism of WCET estimates
- Getting rid of data caches causes performance loss!
- Solution: Try to analyze and predict data cache behavior

2 Single Task Analysis
- Analyze loop-nest oriented code
- Enhance Cache Miss Equations (CME) framework
- Allow arbitrary loop nests
  - Forced loop fusion
  - Concatenate iteration spaces of sequential loops
  - Permit non-rectangular loops
- Derive exact DS reference patterns
- Miss/Hit patterns for each reference

3 Single Task Example
- Input
  - \(A[1..5][1..5]\) – base address = 151944
  - \(B[1..5][1..5]\) – base address = 153068
  - Loop nest:
    for(\(r = 0; r < 5; r++\))
    for(\(c = 0; c < 5; c++\))
- Cache configuration
  - 32 byte cache line, 1 KB cache, direct mapped
- Our output
  - Ref 1: MbMhMhMhMhMhMhMh – total misses = 6
  - Ref 2: MmMmMhMhMhMhMhMh – total misses = 6
  - Ref 3: hhhhhhhhhhhhhhhhhhh – total misses = 0

4 Multiple Task Analysis
- Multiple prioritized tasks share DS \(\rightarrow\) preemption occurs
- Data of preempted task may be evicted from DS
- DS related preemption delay (D-CRPD) required
- Response time of task requires upper bound on
  - Number of feasible preemption points
    - Use execution range (BCET-WCET) of tasks
  - Progress in execution of task at each point
    - Map execution time range of task to iteration range
  - Preemption delay at each point
    - Use access chains (represent memory accesses)

5 Multiple Task Example

<table>
<thead>
<tr>
<th>Task</th>
<th>Period</th>
<th>WCET</th>
<th>BCET</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>20</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>T1</td>
<td>50</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>T2</td>
<td>200</td>
<td>30</td>
<td>15</td>
</tr>
</tbody>
</table>

6 Results (U = 80%)

- Response time increases with decrease in priority
- Our method has least rate of increase
- All task sets deemed schedulable by our method

7 Conclusions
- Schedulability analysis for hard real-time systems
- Single task analysis
  - Hit/miss pattern \(\rightarrow\) number & position of misses
  - Safe and tight WCET estimate
- Multiple task analysis
  - Eliminate infeasible preemptions
  - Tight WCET & response time estimates
  - Increased schedulability of task sets