Non-Intrusive Dynamic Profiler for Multicore Embedded Systems

Sudarshan Sargur and Roman Lysecky
Electrical and Computer Engineering
University of Arizona, Tucson, AZ
sudarshansl@email.arizona.edu, rlysecky@ece.arizona.edu
The Past: Runtime-adaptive/Self-aware Systems

1. APPLICATION INITIALLY EXECUTES ON MICROPROCESSOR
2. PROFILER MONITORS APPLICATION EXECUTION
3. OPTIMIZER CHOOSES BETWEEN SOFTWARE AND HARDWARE ALTERNATIVES
4. RECONFIGURE FPGA, UPDATE PROCESSOR & FPGA V/F
5. RINSE AND REPEAT

Needs accurate profile of application execution
The Past: Non-intrusive Profiling

- Dynamic Application Profiler (DAProf)
  - Provides loop/kernel-level profiling
  - Greater than 95% accuracy
  - Interfaced to microprocessor trace port
  - Nonintrusive
  - 5-10% area overhead

DYNAMIC APPLICATION PROFILER (DAProf)
The Present: Runtime-adaptive/Self-aware Systems

- Multicore and many core systems
  - Thermal and power management
  - Dynamic data-driven application systems
  - Thermal/aging aware dynamic task allocation and scheduling
  - Dependable systems
  - Malware detection
  - Energy optimization using heterogeneous/asymmetric processors
  - Reconfigurable computing
  - Runtime SOC tuning (i.e., tuning cache, memory, NOC, priority encoders, DAM, etc.)

Needs accurate profile of system execution
The Present: Profiling Inaccuracy

- **DAPprof profiler in multicore systems**

Direct application of DAPprof to individual processor cores can lead to inaccurate *system* profile
The Present: Profiling Inaccuracy

- **DAProf profiler in multicore systems**

  - 3% peak error for single-tasked application executing on single processor core (ST-SC)
  - 35% peak error for multi-tasked application executing on multicore processor (MT-MC)
**Dynamic Application Profiler (DAPprof)**

- **Profile Task Filter**
  - Programmable component storing start and end address of each task (or region) to be profiled
  - Monitors trace port bus to detect context switches
    - Asserts CS signal if iAddr falls outside of the current task’s address range
Dynamic Application Profiler (DAPprof)

- **Profile Cache**
  - **Tag**: Address of the short backwards branch
  - **Offset**: Negative branch offset, corresponding to loop size
  - **CurrIter**: Number of iterations for the current loop execution
  - **AvgIter**: Average Iterations per execution of the loop
    - 17-bit fixed point representation with 14 bits integer and 3 bits fractional
Dynamic Application Profiler (DAProf)

- **Profile Cache**
  - **Execs**: 16-entry storing the number of times a loop executes
  - Maintains relative execution count
    - If Execs counter saturates, profiler controller adjusts all loop execution counts by dividing by 2 (implemented as right shift)
    - Applications may saturate at different times on different processor cores

**Profiler Cache**

- **Profile Cache**
  - **Found**
  - **Found Index**
  - **Replace Index**

**Profile FIFO**

- **Profiler Task Filter**
- **Profiler Controller**

**Profile Controller**

- **Tag** (30)
- **Offset** (8)
- **Curriterr** (14)
- **Avgiter** (17)
- **Execs** (16)
- **Inloop** (1)
- **Infunc** (1)
- **Inc** (1)
Dynamic Application Profiler (DAPProf)

- **Profile Cache**
  - **Execs**: 16-entry storing the number of times a loop executes
  - Maintains relative execution count
    - If Execs counter saturates, profiler controller adjusts all loop execution counts by dividing by 2 (implemented as right shift)
    - Applications may saturate at different times on different processor cores

### Local Profile

<table>
<thead>
<tr>
<th>Loop ID</th>
<th>Execs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>42500</td>
</tr>
<tr>
<td>B</td>
<td>22000</td>
</tr>
<tr>
<td>C</td>
<td>34000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Loop ID</th>
<th>Execs</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>34196</td>
</tr>
<tr>
<td>Y</td>
<td>800</td>
</tr>
<tr>
<td>Z</td>
<td>10500</td>
</tr>
</tbody>
</table>

### Global Profile

<table>
<thead>
<tr>
<th>Loop ID</th>
<th>Execs (Actual)</th>
<th>Execs (DAPProf)</th>
<th>% ET (Actual)</th>
<th>% ET (DAPProf)</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>42500</td>
<td>42500</td>
<td>17</td>
<td>29</td>
<td>12</td>
</tr>
<tr>
<td>B</td>
<td>22000</td>
<td>22000</td>
<td>9</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>C</td>
<td>34000</td>
<td>34000</td>
<td>14</td>
<td>24</td>
<td>10</td>
</tr>
<tr>
<td>X</td>
<td>132500</td>
<td>34196</td>
<td>53</td>
<td>24</td>
<td>29</td>
</tr>
<tr>
<td>Y</td>
<td>1100</td>
<td>800</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Z</td>
<td>17500</td>
<td>10500</td>
<td>7</td>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>

Profile Results from Individual Cores

Naively Combined System Profile
Dynamic Application Profiler (DAProf)

- **Options for Improving Multicore Profiling using DAProf**
  - Larger DAProf entry for executions
    - Requires large area increase, and loops saturation are still not be synced across cores
  - Global loops saturations
    - Requires increase in both execution entry and number of loops to maintain profile accuracy
    - Introduce timing and layout challenges

---

**DYNAMIC APPLICATION PROFILER (DAProf)**

**PROFILE CACHE**

<table>
<thead>
<tr>
<th>Tag (30)</th>
<th>Offset (8)</th>
<th>Current (14)</th>
<th>Average (17)</th>
<th>Execs (16)</th>
<th>Inloop (1)</th>
<th>Infunc (1)</th>
<th>Incs (1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**DAProf with multitask and multicore scaling (DAProf-MT-MC)**

- Extend profile task filter with *per task* instruction counts
  - \( pcValid \): Processor trace port indicating current PC is a valid instruction
  - **Instruction Count**: 64-bit register storing per task count of instructions executed
  - Instructions counts used to scale the estimated percentage of execution (%ET) across tasks

**DAProf with multicore scaling (DAProf-MC)**

- Extend profile task filter with *per core* instruction counts
  - Instructions counts used to scale the estimated percentage of execution (%ET) across tasks
Experimental Setup

- Multitasked benchmarks
  - Individual application tasks taken from the MiBench benchmark suite
  - Labeled $MT_x.y$, where $x$ indicates the number of tasks within the benchmark and $y$ is a unique ID

- Multicore benchmarks
  - 10 multicore benchmarks
  - Two multitasked application mapped onto separate processor core

- Evaluate the accuracy of DAProf, DAProf-MC, and DAProf-MT-MC

<table>
<thead>
<tr>
<th></th>
<th>MT2.3</th>
<th>MT2.5</th>
<th>MT3.1</th>
<th>MT3.3</th>
<th>MT4.1</th>
<th>MT4.2</th>
<th>MT5.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CJPEG</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DJPEG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FFT</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIFF2BW</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUSAN</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIJKSTRA</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIT COUNT</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRINGSEARCH</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSORT</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAWCAUDIO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAWDAUDIO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

| MC1            | ✓     | ✓     |       |       |       |       |       |
| MC2            | ✓     | ✓     |       |       |       |       |       |
| MC3            | ✓     |       |       |       | ✓     |       |       |
| MC4            | ✓     | ✓     | ✓     |       |       |       |       |
| MC5            | ✓     |       |       |       |       |       |       |
| MC6            | ✓     | ✓     | ✓     | ✓     |       |       |       |
| MC7            | ✓     |       |       |       |       | ✓     |       |
| MC8            |       | ✓     | ✓     |       |       |       |       |
| MC9            |       | ✓     | ✓     |       |       |       |       |
| MC10           |       |       |       |       | ✓     | ✓     | ✓     |
Experimental Results – Profiling Accuracy

DAProf-MT-MC’s average error ranges from 0% to 0.7%
DAProf-MT-MC’s reduces peak error by as much as 33%
Experimental Results – Area Requirements

DAProf-MC’s requires only 0.5% larger Profiler Task Filter

DAProf-MT-MC’s requires 9.1% larger Profiler Task Filter
The Distant Future (The Year 2017)

- Investigate methods to support for task migration
- Integration with operating system
- Adaptive profiling resolution
- Investigate the impact on the dynamic optimization techniques
  - i.e., How much improvement can be achieved with increased profiling accuracy?
  - Particularly expect that worst case scenario can be better detected and mitigated
Thank You, Gracias, Danke, Merci, Tak, ありがとう, Kittos