Non-Intrusive Online Timing Analysis of Large Embedded Applications

Boris Dreyer, Christian Hochberger

Boris Dreyer
dreyer@rs.tu-darmstadt.de

Prof. Dr.-Ing. Christian Hochberger
Computer Systems Group
Department of Electrical Engineering and Information Technology
Technische Universität Darmstadt, Germany
Agenda

• Execution Time Profiles

• Histograms
  – Scalable histogram algorithm
  – Debie1 evaluation

• Hardware platform
  – Feasibility and performance

• Conclusion
What do we want to do?

Profile large embedded application

**non-intrusive**
- No code instrumentations
- No altering of timing behavior

**online**
- Profiling while the target system executes the application

**using ETPs**
- Execution time profiles (ETP)
- Execution time probability distribution
Design Decisions

Profile large embedded application

- non-intrusive: Using trace-port of modern microcontrollers
- online: Processing trace-events in hardware
- using ETPs: Generate histograms in hardware
Execution Time Profiles (ETPs)

Execution Time Profiles help us with:

- Probabilistic schedulability analysis: ETPs are used to calculate response time distribution
- Scheduling tasks while maintaining a given quality of service level
Timing Analysis Platform

How to compute execution time histograms in hardware?

How to compute histograms of data with unknown value ranges in hardware?
Scalable Histogram Algorithm Example

Requirements

- Small constant amount of bins
- Try to keep bin sizes small
- Assign each measured value to a bin

Event sequence <5,4,11,7,54,10>

One compression step
Statistics Storage

Histogram 1
Histogram 2
Histogram 3
Histogram 4
Histogram 5

<table>
<thead>
<tr>
<th></th>
<th>Bin 1</th>
<th>Bin 2</th>
<th>Bin 3</th>
<th>Bin 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Histogram 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Histogram 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Histogram 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Histogram 4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Histogram 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Timing Analysis Module

How fast can the Timing Analysis Module process waypoint edge events (WPEs)?
Timing Analysis Module
Loop and Function Automaton

Models one function or one loop
Timing Analysis Module
Timing Analysis Module
Timing Analysis Module
Timing Analysis Module

- Loop & Function Automaton
  - Edge Forwarding Tree
  - Runtime/Iteration Poll Tree
  - Measurement FIFO Buffer
  - Simple Statistics Storage Controller
  - Simple Statistics & Scalable Histogram Storage Controller
  - Function Runtime Loop Iterations Statistics Storage

- Remote Storage Access Handler
  - Edge Runtime Statistics Storage
  - Histograms & Simple Statistics

WPEs
Timing Analysis Module

How fast can the Timing Analysis Module process waypoint edge events (WPEs)?

Assuming one WPE per clock cycle
Measurement FIFO Buffer Utilization

Debie1 Benchmark

- 68 loops
- 172 functions
- 70,000,000 WPEs
Debie1 Benchmark

Using the Debie1 benchmark

- Onboard software of the DEBIE-1 satellite
- Consists of 4 tasks and 2 interrupt service routines

Target Hardware:

- Xilinx Zynq XC7Z020
  - dual-core ARM Cortex-A9 @667Mhz
  - separate L1 Caches, shared L2 Caches
- Debie1 on one core, on the second core a custom benchmark in a FreeRTOS instance for generating interference on the shared L2 cache
Debie1 Tasks

HandleAcquisition Task

HandleHealthMonitoring Task

HandleHitTrigger Task

HandleTelecommand Task

bin amount: 89258
bin size: 1024
bin size: 2048

bin amount: 23394
bin size: 256
bin size: 512

bin amount: 7503
bin size: 64
bin size: 128

bin amount: 3862
bin size: 32
bin size: 64

128 bins   64 bins
Conclusion

Achievements

✓ Histograms are an efficient way to store execution times
✓ Creation of histograms on dedicated hardware in real time
✓ Practical unlimited observation time

✓ No previous runtime information required
✓ Creation of histograms in one analyzing pass
Thank you!