

Online Analysis of Debug Trace Data for Embedded Systems

CONIRAS

Normann Decker, Boris Dreyer, <u>Philip Gottschling</u>, Christian Hochberger, Alexander Lange, Martin Leucker, Torben Scheffel, Simon Wegener, Alexander Weiss







GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

CONIRAS



GEFÖRDERT VOM



Bundesministerium für Bildung und Forschung



This work was funded by the German Federal Ministry for Education and Research within the project CONIRAS (funding ID 01|S13029) and ARAMIS II (funding ID 01|S16025) and by the European Union's Horizon 2020 research and innovation programme within the project COEMS (grant agreement no. 732016). The responsibility for the Content remains with the authors.



Outline



- Debugging Techniques and Problems
- Online Trace Reconstruction
- Online Trace Analysis
 - Execution Time
 - Runtime Verification
- Conclusion



Debugging/Profiling





21.03.2018 | TU Darmstadt | Computer Systems Group | Philip Gottschling | 4





Debugging Tools – Desktop Development

- Behavioral
 - Unit tests (golden values, corner cases)
 - Conditional breakpoints (gdb, jdb)
- Performance
 - Valgrind
 - Sanitizers (leaks, locks, ...)
 - ..
- Assumption
 - Nearly no time constraints
 - Lots of computing power





Debugging Tools – Embedded Development

- Behavior
 - Register Values
 - Breakpoints and single stepping
 - Printf
- Simulation/Emulation
- Model based static profiling

Does this work for multi cores? They have much more power...



FCHNISCHE

UNIVERSITÄT DARMSTADT



Debugging Tools – Embedded Development

- Multicore
 - Registers and breakpoints as before
 - Modeling fails due to non-determinism
 - New problems because of real parallelism

- Consequences
 - Very pessimistic estimators
 - No estimation at all



21.03.2018 | TU Darmstadt | Computer Systems Group | Philip Getterming

Runtime Debugging

- Hybrid approach
 - Static control flow graph analysis
 - Dynamic program flow trace

- Execution time analysis
- Runtime verification







Runtime Debugging - Insights



- Trace macrocell inside ARM Processor (Coresight)
- Information provided
 - Synchronization
 - Indirect branches
 - Branch taken or not taken
 - Time difference to last branch
 - Context change
- Postprocessing inevitable





Trace Reconstruction



- FPGA based online reconstruction
 - 4Gb RLDRAM lookup-tables
 - Speculative and lookahead execution
 - Event (re)ordering





Trace Reconstruction - Hardware









Execution Time Analysis - WCET







Execution Time Analysis







21.03.2018 | TU Darmstadt | Computer Systems Group | Philip Gottschling | 13

Execution Time Analysis – Workflow





21.03.2018 | TU Darmstadt | Computer Systems Group | Philip Gottschling | 14





Execution Time Analysis – Examples

• Debie

- Software for a satellite instrument
- Well understood real world example
- 1923 Basic blocks, 172 Routines, 68 Loops
- TACLeBench



Runtime Verification – Generation





Time consuming synthesis!





Runtime Verification – Details







TECHNISCHE Runtime Verification – Observation UNIVERSITÄT DARMSTADT Β B # # +# ==



Runtime Verification - Workflow





21.03.2018 | TU Darmstadt | Computer Systems Group | Philip Gottschling | 19



Runtime Verification - Benefits



- **TeSSLa** Temporal Stream-based Specification Language:
 - Arithmetics, aggregations
 - Real time conditions
 - State machines
- High resource utilization
- Turnaround times less than 1s



Conclusion



- Online trace analysis
 - Hybrid WCET
 - Runtime Verification

- Arbitrary long observation times
- Short turnaround times
- Compact form factor for field applications

