Code-Level Timing Analysis of Embedded Software

EMSOFT’12 Invited Talk Session Outline

Heiko Falk
Institute of Embedded Systems / Real-Time Systems
Ulm University
Ulm, Germany
Heiko.Falk@uni-ulm.de

Kevin Hammond
School of Computer Science
University of St. Andrews
St. Andrews, Scotland
kh@cs.st-andrews.ac.uk

Kim G. Larsen
Department of Computer Science
Aalborg University
Aalborg, Denmark
kgl@cs.aau.dk

Björn Lisper
Department of Computer Science
Mälardalen University
Västerås, Sweden
bjorn.lisper@mdh.se

Stefan M. Petters
School of Engineering
Polytechnic Institute of Porto
Porto, Portugal
smp@isep.ipp.pt

ABSTRACT

Embedded systems are often business- or safety-critical, with strict timing requirements that have to be met for the information-processing. Code-level timing analysis (used to analyse software running on some given hardware w.r.t. its timing properties) is an indispensable technique for ascertaining whether or not these requirements are met. However, recent developments in hardware, especially multi-core processors, and in software organisation render analysis increasingly more difficult, thus challenging the evolution of timing analysis techniques. This special session aims to give an overview over the current state of the art and the future challenges w.r.t. code-level timing analysis and introduces TACLe, a recently started EU-funded networking activity targeting these challenges.

Categories and Subject Descriptors
C.3 [Real-time and embedded systems]

General Terms
Algorithms, Performance

Keywords
Timing Analysis, WCET, Multi-Core Processors

Timing Analysis for Multicore/Manycore Architectures

Kevin Hammond, University of St. Andrews
Multicore/manycore architectures present significant new challenges to timing analysis. Since hardware resources are shared between processor cores, activities on one core can affect the timing of activities on other cores. It is no longer possible to consider execution times independently for each core: rather, timing analysis must take all possible simultaneous activities into account. These processor architectures thus lack timing composability, i.e., the ability to infer timing for a full system from the timing of its parts, which has a highly negative effect on the timing predictability of multicore/manycore systems. This talk introduces the research area, outlines the major challenges that must be overcome and highlights promising new approaches to timing analysis for multicore/manycore architectures.

Reconciling Compilation and Timing Analysis

Heiko Falk, Ulm University

The current state of the art in designing software for hard real-time systems is heavily unsafe. On one hand, the actual industrial design practice relies on measurements or simulations so that no guarantees about the worst-case timing of a piece of software can be derived. On the other hand, current compilers usually optimise the average-case execution time (ACET) of a program, instead of the worst-case execution time (WCET). This talk presents a WCET-aware compiler for hard real-time systems. A WCET timing model provides valuable data about the worst-case behaviour of a program to be compiled and optimised. This timing model is then used by specialised optimisations which achieve a fully automatic minimisation of the WCET.

Early-Stage and Portable Timing Analysis

Stefan M. Petters, Polytechnic Institute of Porto

Since redesigning a software system is very costly, designers usually choose to over-specify the hardware initially and then just verify that it is indeed sufficiently powerful. However, as systems’ complexity rises, these initial safety margins can prove to be very expensive. Undertaking lightweight (but less precise) analysis in the early stages of the design process has the potential to drastically reduce total hardware costs.
Analysis of Mixed-Critical Embedded Systems with Multiple Objectives
Kim G. Larsen, Aalborg University
We demonstrate how a combination of model checking and statistical model checking may be applied to analyse on the one hand hard real-time system response time requirements and on the other hand soft real time requirements allowing a trade-off between time and energy consumption.

TACLe – An EU COST Action on Timing Analysis on Code-Level
Björn Lisper, Mälardalen University
TACLe is a new COST Action that will coordinate and support European research in code-level timing analysis through a number of networking activities. The Action gathers all the prominent European groups in the area as well as groups in neighbouring areas. The coordinated research is organised in the following working groups:

WG1: Timing models for multi-cores and timing composability
WG2: Tooling Aspects
WG3: Early-stage timing analysis
WG4: Other resources than time

The Action is motivated particularly by the challenges to code-level timing analysis brought by the rapid transition to multi-core technology, and anticipated future evolution to many-core architectures. These challenges require a coordinated effort to be met. The purpose of TACLe is to provide this coordination.