Formal Analysis of Processor Timing Models

Reinhard Wilhelm*

Informatik
Universität des Saarlandes
Saarbrücken

Abstract. Hard real-time systems need methods to determine upper bounds for their execution times, usually called worst-case execution times. This talk gives an introduction into state-of-art Timing-Analysis methods. These use Abstract Interpretation to predict the system's behavior on the underlying processor's components and Integer Linear Programming to determine a worst-case path through the program. The abstract interpretation is based on an abstract processor model that is conservative with respect to the timing behavior of the concrete processor. Ongoing work is reported to analyze abstract processor models for properties that have a strong influence on the expected precision of timing prediction and also on the architecture of the timing-analysis tool. Some of the properties we are interested in can be model checked.

WCET Determination

Hard real-time systems need methods to determine upper bounds for their execution times, usually called worst-case execution times, (WCET). Based on these bounds, a schedulability analysis can check whether the underlying hardware is fast enough to execute the system's task such that they all finish before their deadlines. This problem is nontrivial because performance-enhancing architectural features such as caches, pipelines, and branch prediction introduce "local non-determinism" into the processor behavior; local inspection of the program can not determine what the contribution of an instruction to the program's overall execution time is. The execution history determines whether the instruction's memory accesses hit or miss the cache, whether the pipeline units needed by the instruction are occupied or not, and whether branch prediction is correct or not.

Tool Architecture

State-of-art Timing-Analysis methods split the task into (at least) two subtasks, the prediction of the task's behavior on the processor components such as caches and pipelines, formerly called "micro-architecture modeling" [HBW94], and the determination of a worst-case path. They use Abstract Interpretation to predict

* Work reported herein is supported by the Transregional Collaborative Research Center AVACS of the Deutsche Forschungsgemeinschaft.

the system's behavior on the underlying processor's components and Integer Linear Programming to determine a worst-case path through the program [LMW99]. A typical tool architecture is the one of aiT, the tool developed and marketed by AbsInt Angewandte Informatik in Saarbrücken, cf. Fig. 1.

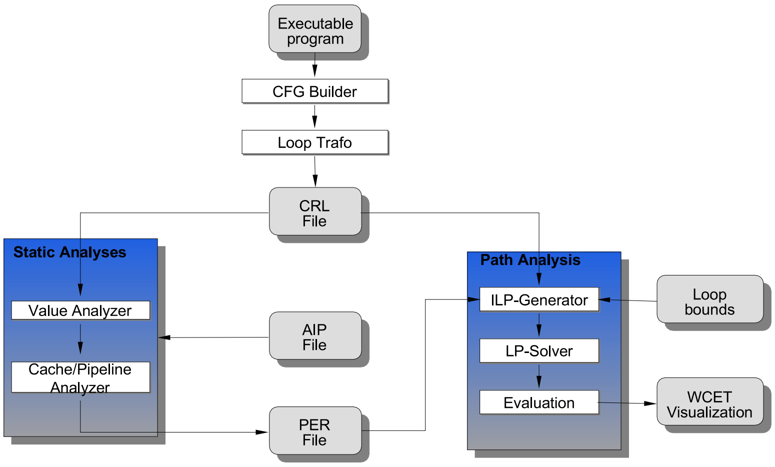

Fig. 1. Architecture of the aiT WCET analysis tool

The articles [FHL+01,LTH02] report on WCET tool developments for complex processor architectures, namely the Motorola ColdFire 5307 and the Motorola PowerPC 755. These were the first fully covered complex processors.

Timing Anomalies

The architecture of WCET–tools and the precision of the results of WCET analyses strongly depend on the architecture of the employed processor [HLTW03]. Out-of-order execution and control speculation introduce interferences between processor components, e.g. caches, pipelines, and branch prediction units. These interferences forbid modular designs of WCET tools, which would execute the subtasks of WCET analysis consecutively. Instead, complex integrated designs are needed resulting in high demand for space and analysis time.

In the following, several such properties of processor architectures are described. They cause the processor to display what is called *Timing Anoma-*

lies [Lun02]. Timing anomalies are contra-intuitive influences of the (local) execution time of one instruction on the (global) execution time of the whole program. The interaction of several processor features can interact in such a way that a locally faster execution of an instruction can lead to a globally longer execution time of the whole program. This is only the first case of a timing anomaly. The general case is the following. Different assumption about the processor's execution state, e.g. the fact that the instruction is or is not in the instruction cache, will result in a difference ΔT_1 of the execution time of the instruction between these two cases. Either assumption may lead to a difference ΔT of the global execution time compared to the other one. We say that a timing anomaly occurs if either

$\Delta T_1 < 0$ i.e., the instruction executes faster, and
- $\Delta T < \Delta T_1$, the overall execution is accelerated by more than the acceleration of the instruction, or
- $\Delta T > 0$, the program runs longer than before.

$\Delta T_1 > 0$ i.e., the instruction takes longer to execute, and
- $\Delta T > \Delta T_1$ i.e., the overall execution is extended by more than the delay of the instruction, or
- $\Delta T < 0$ i.e., the overall execution is the program takes less time to execute than before.

The case $\Delta T_1 < 0 \wedge \Delta T > 0$ is a critical case for WCET analysis. It makes it impossible to use local worst case scenarios for WCET computation. This necessitates a conservative, i.e., upper approximation to the damages potentially caused by all cases or forces the analysis to follow all possible scenarios.

Unfortunately, as [LS99,Lun02] have observed, the worst case penalties imposed by a timing anomaly may not be bounded by an architecture-dependent, but program-independent constant, but may depend on the program size. This is the so-called *Domino Effect*. This domino effect was shown to exist for the Motorola PowerPC 755 in [Sch03].

Formal Analysis of Processor Timing Models

The abstract-interpretation-based timing analysis is based on abstract processor models that are conservative with respect to the timing behavior of the concrete processors. To prove this is a major endeavor to be undertaken in the Transregional Collaborative Research Center AVACS. Another line of research is the derivation of processor timing models from formal specifications in VHDL or Verilog.

We are currently applying formal analysis of timing models to check for relevant properties, e.g., use model checking to detect timing anomalies and domino effects or their absence, resp. Bounded model checking can be used to check for the existence of upper bounds on the damage done by one processor component onto the state of another one, e.g. the damage of a branch misprediction to the instruction cache by loading superfluous instructions. The bound can be computed from architectural parameters, such as the depth of the pipeline and the length of prefetch queues.

Acknowledgements

Thanks go to Stephan Thesing for clarifications about timing anomalies.

References

[FHL+01] C. Ferdinand, R. Heckmann, M. Langenbach, F. Martin, M. Schmidt, H. Theiling, S. Thesing, and R. Wilhelm. WCET Determination for a Real-Life Processor. In T.A. Henzinger and C. Kirsch, editors, *Embedded Software*, volume 2211 of *Lecture Notes in Computer Science*, pages 469 – 485. Springer, 2001.

[HBW94] Marion G. Harmon, T.P. Baker, and David B. Whalley. A Retargetable Technique for Predicting Execution Time of Code Segments. *Real-Time Systems*, 7:159–182, 1994.

[HLTW03] Reinhold Heckmann, Marc Langenbach, Stephan Thesing, and Reinhard Wilhelm. The influence of processor architecture an the design and the results of WCET tools. *IEEE Proceedings on Real-Time Systems*, 91(7):1038–1054, July 2003.

[LMW99] Yau-Tsun Steven Li, Sharad Malik, and Andrew Wolfe. Performance estimation of embedded software with instruction cache modeling. *Design Automation of Electronic Systems*, 4(3):257–279, 1999.

[LS99] Thomas Lundqvist and Per Stenström. Timing anomalies in dynamically scheduled microprocessors. In *Proceedings of the 20th IEEE Real-Time Systems Symposium (RTSS'99)*, pages 12–21, December 1999.

[LTH02] M. Langenbach, S. Thesing, and R. Heckmann. Pipeline Modelling for Timing Analysis. In *Static Analysis Symposium*, volume 2274 of *LNCS*, pages 294–309. Springer Verlag, 2002.

[Lun02] Thomas Lundqvist. *A WCET Analysis Method for Pipelined Microprocessors with Cache Memories*. PhD thesis, Chalmers University of Technology, Göteborg, Sweden, 2002.

[Sch03] Joern Schneider. *Combined Schedulability and WCET Analysis for Real-Time Operating Systems*. PhD thesis, Saarland University, 2003.