WCET Squeezing: A Practical Approach for Proven Precise WCET Bounds

Jens Knoop

Institute of Computer Languages
Vienna University of Technology
knoop@complang.tuwien.ac.at

The time bound computed by a WCET analyzer is usually not tight, leaving a gap between the actual and the computed *worst-case execution time* (*WCET*) of a program. In this talk we present a novel practical on-demand WCET feasibility refinement technique, called *WCET Squeezing*, for minimizing or even closing this gap and proving the WCET bound computed precise. Conceptually, WCET Squeezing works by checking the feasibility of the WCET path delivered by a WCET analyzer. If it is proven infeasible, it will be excluded from a re-run of the WCET analysis. If it is proven feasible, the WCET bound is proven precise. WCET Squeezing is an *anytime* algorithm, i.e., it can be stopped at any time without violating the soundness of its results. This provides the user complete control on the computational effort of WCET Squeezing, which makes WCET Squeezing a conceptually very versatile means.

WCET Squeezing can be used for addressing (1) the classical WCET problem of computing a WCET bound that comes as close as possible to the actual one or even matches it. It can be used for (2) a pragmatic variant of this problem when a program needs only be proven to be fast enough, i.e., to not exceed a given deadline. If the initially computed WCET of the program is above this deadline, WCET Squeezing can be stopped as soon as the squeezed WCET of the program is below it (proving that the program does not exceed its deadline), or if the squeezed WCET is precise but above the deadline (proving that the program will exceed the deadline). Last but not least, WCET Squeezing can be used (3) until a given time budget is exhausted to compute a tight(er) WCET bound for a program if there is need to. It is worth noting that all these three application modes of WCET Squeezing are out of the scope of traditional WCET analyzers.

Technically, WCET Squeezing combines *symbolic program execution* with the *implicit path enumeration technique* (*IPET*) to iteratively improve some initially computed WCET bound. It can be added as a post-process to any WCET analyzer that encodes the IPET problem as an *integer linear program* (*ILP*). We implemented WCET Squeezing as an extension of our r-TuBound WCET analysis tool chain and evaluated it on a set of the Mälardalen WCET benchmarks. These experiments indicate that WCET Squeezing can significantly tighten the WCET estimates of programs computed by traditional WCET analyzers, and often succeeds to compute a proven precise bound – all this at moderate costs.

This is joint work with Laura Kovács and Jakob Zwirchmayr. The research underlying it is partly supported by the FP7-ICT Project 288008 Time-predictable Multi-Core Architecture for Embedded Systems (T-CREST), the FWF National Research Network RiSE (S11410-N23), the FWF Hertha Firnberg Research grant (T425-N23), the WWTF PROSEED grant (ICT C-050), and the CeTAT project of the TU Vienna.