Towards Transparent Worst-Case Execution Time Optimization

Jens Knoop

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

"Make the common case fast!" Following David A. Patterson this is one of eight great ideas that contributed much to the success and progress of computer science and computing technology over the past six decades. Originally coined for the field of computer architecture, this idea is often interpreted as "make the common case fast, and the rare case correct" in program optimization. In fact, so-called aggressive optimizations even accept to improve the common case at the expense of imparing the rare one. This is in contrast to the requirements of optimizing real-time systems, where the above idea is re-phrased as "make the worst case fast." Dedicated optimizations for worst-case execution time (WCET) optimization have been developed respecting this latter idea. A common short-coming of these techniques is that they are blind and not focused because of the lack of information on which program path(s) actually take longest to execute. Consequently, these techniques can not quantify their effect. Their impact is not transparent.

In this talk I will present an approach towards *transparent WCET optimization*. This approach evolves as a combination of *WCET squeezing* and *property-oriented expansion* together with *program optimization*. In this approach, first, WCET squeezing is applied to identify the program paths(s) that take longest to execute. Second, property-oriented expansion is used to segregate these paths, which, third, are subject to thorough program optimization. Being an any-time algorithm, the whole 3-step procedure is thereafter applied repeatedly until no further improvement is possible or the guaranteed worst-case execution time of the optimized program satisfies an externally imposed demand.

In contrast to previous approaches to WCET optimization, the new approach is unique (1) to improve exactly those program paths which take longest to execute, and (2) to precisely quantify the achieved reduction of the WCET of the optimized program over the one of the original program.

This work is supported by the *EU FP7 COST Action IC 1202 Timing Analysis on Code-Level (TACLe)*. It builds on joint work with Laura Kovács and Jakob Zwirchmayr.