New challenges in adaptive real-time systems with parametric WCET

Clément Ballabriga, Julien Forget, Sandro Grebant, Giuseppe Lipari

New challenges in adaptive real-time systems with parametric WCET

Clément Ballabriga, Julien Forget, Sandro Grebant, Giuseppe Lipari
Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL, F-59000, Lille, France
firstname.lastname@univ-lille.fr

The execution time of a real-time task can exhibit large variability due to software parameters (e.g. program inputs) or hardware parameters (e.g. cache state). Traditionally, static Worst-Case Execution Time (WCET) analysis provides a numeric upper-bound to the execution time of a task for any possible combination of the software and hardware parameters. Instead, *parametric WCET analysis* produces a *formula* that represents the WCET as a function of the parameters. The formula can later be *instantiated* with concrete parameter values to provide an upper-bound to the execution time for those parameter values.

We recently proposed a novel approach to parametric WCET analysis [1], which analyzes the binary code of a procedure to produce a WCET formula that represents the WCET of the procedure as a function of its arguments. Compared to previous works (see [1] for an in depth comparison and references), this approach is the first to be simultaneously *adaptive*, *automated*, and *embeddable*. Regarding adaptivity, experiments on TACLeBench [2] demonstrate that in many cases the instantiated WCET varies significantly depending on procedure argument values. Regarding embeddability, the size of the WCET formula and the instantiation time are very small compared to the program size, which enables on-line instantiation. Regarding automation, our approach takes the binary code of a procedure as input and produces a WCET formula dependent on the procedure arguments as output, without requiring assistance from the programmer. The approach is implemented in a publicly available toolset so as to foster collaboration[1]. We believe that this work paves the way for the analysis and implementation of a new range of adaptive real-time systems, as we will illustrate in the rest of this paper.

I. Automated parametric WCET analysis

First, we provide a brief overview of our parametric WCET approach, by illustrating it on the program of Figure 1a. Starting from the binary code of function f, the analysis proceeds as follows:

a) Tree-based program representation: the binary code is translated into a *Control-Flow Tree* (CFT, akin to a Control-Flow Graph) that represents the program structure, where nodes are basic blocks. It consists of a sequence (the root node *Seq*) of basic blocks (A, D) and of an alternative (Alt) between two subtrees (B or C). Output edges of alternative nodes are annotated with conditions on the procedure inputs (`r0`), inferred by abstract interpretation of the binary code. A CFT can also contain loop nodes, not shown in this example.

b) WCET formula: The CFT is translated into a WCET formula. Essentially: the WCET of a Seq node is the sum of the WCETs of its subtrees (denoted $\oplus$); the WCET of an Alt node is the maximum among the WCETs of its subtrees (denoted $\uplus$); the WCET of an alternatives' subtree is multiplied by its condition. Let us assume that the hardware analysis infers that the WCET for A is 10, for C is 5, and that the WCET of B and D are *symbolic*, i.e. unknown statically (denoted $\omega(B)$, $\omega(D)$). We obtain:

$$10 \oplus (((\texttt{r0} \geq 11) \circledast \omega(B)) \uplus ((\texttt{r0} \leq 10) \circledast 5)) \oplus \omega(D)$$

It is important to underline that, for the sake of clarity, in this example we show a simplified version of the formula. In [1], in order to correctly model the impact of caches, each WCET is represented by a list of values. Furthermore, in the general case special simplification rules are applied to reduce the size of the formula when possible.

c) Formula instantiation: The formula is instantiated when symbolic values become known. For instance, assuming $n = 0$ (i.e. $\texttt{r0} = 0$), $\omega(B) = \omega(D) = 8$, we obtain a WCET of 23. The formula is finally compiled into C code, which can be embedded in the program to enable the implementation of an adaptive system.

II. Opening new problems for adaptive real-time systems

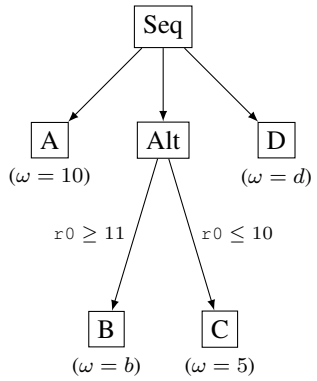
Real-time models where the execution time of a task is not represented by a single value have been considered before, for instance in the mixed-criticality model [3] or in the Generalized Multiframe model [4]. However, in these approaches the WCET cannot be related to the inputs of the task. Our approach enables to consider new scheduling problems where scheduling

[1]Artifact and tools:

- Artifact for [1]: https://gitlab.cristal.univ-lille.fr/sgrebant/rtns_2023_artifact
- Polymalys (abstract interpretation): https://gitlab.cristal.univ-lille.fr/otawa-plugins/polymalys;
- WSymb (symbolic WCET computation): https://gitlab.cristal.univ-lille.fr/otawa-plugins/WSymb.

```
1  f:                              @  int f(int n) {
2          @ ...                   @    /* A */
3          str r0, [fp, #-32]      @    /* A */
4          @ ...                   @    /* A */
5          ldr r3, [fp, #-32]      @    /* A */
6          cmp r3, #10             @    if (n <= 10) /* A */
7          bgt .L2                 @    {   /* still A */
8          @ ...                   @        /* C */
9          b       .L3             @    }   /* C */
10 .L2:                            @    else {
11         @ ...                   @        /* B */
12 .L3:                            @    }
13         @ ...                   @    /* D */
14         bx lr                   @    return; /* still D */
15 .global main                    @  }
16 main:                           @  int main() {
17         @...                    @    /* ... */
18         ldr r0, [fp, #-8]       @    /* Setting parameters */
19         bl f                    @    f(i); /* function call */
20         @ ...                   @  }
```

(a) Program code



(b) Control-Flow Tree of f

Fig. 1: A program to analyze

decisions and schedulability depend *on the program inputs*. Our model implies significant differences compared to previous works that consider multiple possible execution times:

1) In our model, the dependence between the system WCETs and the system environment (perceived through the program input) is explicit, while it is only implicitly modeled through WCET values in the literature;
2) In our model, a single program input can impact simultaneously the WCET of several tasks, while the WCETs of different tasks are independent in the literature;
3) In the literature, the WCET of a given task is modeled by a set of values. In our model, a WCET formula can correspond to an a priori unknown number of different instantiated WCET values.

Considering these peculiarities, we envision several related open scheduling problems. We detail two of them in the remainder of this section. Let us stress that WCET formulae can be instantiated either off-line or on-line. First, a formula can be instantiated repeatedly off-line to quickly explore the parameters space with low execution cost. We propose to consider the following problem, related to sensitivity analysis [5]:

Problem 1. *Considering a task set where the WCET of a task is represented by a WCET formula, determine which valuations of the system inputs make the task set schedulable (difficult due to (2)).*

Second, a formula can be instantiated on-line to efficiently implement an *adaptive* real-time system. As many parameter values become known only at run-time (e.g. program inputs), on-line instantiation can produce a lower WCET than a single WCET computed off-line. This can benefit many adaptive scheduling techniques. Here, semi-clairvoyant scheduling for mixed-criticality systems [6] comes to mind. As proposed in [6], using our approach we can determine at job release whether the WCET of the job is below the LO-criticality WCET.

Problem 2. *Semi-clairvoyance for scheduling a task set where the WCET of a task is represented by a WCET formula (difficult due to (3)).*

REFERENCES

[1] S. Grebant, C. Ballabriga, J. Forget, and G. Lipari, "WCET analysis with procedure arguments as parameters," in *The 31st International Conference on Real-Time Networks and Systems (RTNS 2023)*, ser. RTNS 2023. Dortmund, Germany: ACM, New York, USA, Jun. 2023.

[2] H. Falk, S. Altmeyer, P. Hellinckx, B. Lisper, W. Puffitsch, C. Rochange, M. Schoeberl, R. B. Sørensen, P. Wägemann, and S. Wegener, "TACLeBench: A Benchmark Collection to Support Worst-Case Execution Time Research," in *16th International Workshop on Worst-Case Execution Time Analysis*. Toulouse, France: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, pp. 2:1–2:10. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02610690

[3] S. Vestal, "Preemptive Scheduling of Multi-criticality Systems with Varying Degrees of Execution Time Assurance," in *28th IEEE International Real-Time Systems Symposium (RTSS 2007)*, Dec. 2007, pp. 239–243, iSSN: 1052-8725.

[4] S. Baruah, D. Chen, S. Gorinsky, and A. Mok, "Generalized multiframe tasks," *Real-Time Systems*, vol. 17, pp. 5–22, 1999.

[5] E. Bini, M. Di Natale, and G. Buttazzo, "Sensitivity analysis for fixed-priority real-time systems," *Real-Time Syst*, vol. 39, no. 1, pp. 5–30, Aug. 2008. [Online]. Available: https://doi.org/10.1007/s11241-006-9010-1

[6] K. Agrawal, S. Baruah, and A. Burns, "Semi-Clairvoyance in Mixed-Criticality Scheduling," in *2019 IEEE Real-Time Systems Symposium (RTSS)*. Hong Kong, China: IEEE, Dec. 2019, pp. 458–468, iSSN: 2576-3172.