CISTER
Research Center in
Real-Time & Embedded
Computing Systems

Technical Report

Another look at the pWCET estimation problem

Vincent Nélis
Patrick Meumeu Yomsi
Luis Miguel Pinho
Guillem Bernat

CISTER-TR-141105
12-02-2014

Another look at the pWCET estimation problem

Vincent Nélis, Patrick Meumeu Yomsi, Luís Miguel Pinho
CISTER/INESC TEC, ISEP
Polytechnic Institute of Porto
Porto, Portugal
{nelis, pamyo, lmp}@isep.ipp.pt

Guillem Bernat
Rapita Systems, Ltd.
York, United Kingdom
bernat@cs.york.ac.uk

I. Motivation of this work

In many application domains such as automotive and aerospace, it is commonplace that only a subset of the application system is subject to strict requirements, as only few applications actually control or interact with critical components. These requirements often include real-time requirements that compel the applications to output the results of their computations within specified time frames. Such real-time applications are commonly categorized as safety-critical, mission-critical, and non-critical depending on the amplitude of the consequences in case of a failure of the application. Typically, a failure or malfunction of a safety-critical application may result in death or serious injury to people, loss or severe damage to equipment or environmental harm, whereas a failure of a mission-critical application may result in a failure of the entire system, but without damaging it nor its embedding environment, and a failure of a non-critical application has no severe consequences.

A common misconception from academicians is to believe that violating a timing requirement (i.e. missing a task deadline) means that one or several components of the system have failed and thus it is often assumed as a shortcut that underestimating the worst-case execution time (WCET) of a critical application engenders death and irreversible and catastrophic consequences. This is wrong. In fact, if that statement was true one could (safely) conclude on an extremely poor design of the system. In real life, all systems have built-in mechanisms in case the behavior of a task goes off-script and as a rule of thumb, the more critical the system, the more checks and mechanisms to deal with undesired situations, e.g. duplicate/triplicate activities, online monitoring, watchdogs, etc. The objective of WCET analysis is not to compute "safe" bounds on the execution time, as the concept of safety is a property of the system and is not related to the timing behavior of a task. Instead, WCET analysis tools must provide reliable estimations of the tasks WCETs so that the system designers can design an execution environment that is safe for the application.

There are different approaches to WCET analysis, commonly referred to as static, measurement-based, and probabilistic. The authors of [1] have remarkably summarized them as follows. Static timing analysis are based on an accurate system description model. The actual trends to enhance the accuracy of the produced WCET estimates is to increasingly refine the system models, accounting for memory hierarchy, buses and other architectural components while analytically estimating WCET. Measurement-based solutions demand for simple models but are confronted with the problem of guaranteeing the coverage of all the possible execution conditions in order to obtain reliable WCET estimates. Probabilistic approaches focus mainly on (a) deriving the distribution of the execution time (rather than computing the maximum value only) and (b) computing the probability of exceeding a given execution time (a lot of work has been done in the PROARTIS and PROXIMA projects [2], [3]). Note that (a) is more challenging than (b) as one need to manipulate distributions and (b) can be obtained without going through (a). Recently, the current trend in probabilistic approaches is to apply results from the extreme value theory (EVT) framework to the WCET estimation problem [1], [4] but these techniques are still controversial today because the main assumption required for the application of EVT, i.e., the observations are i.i.d., is hardly verified for realistic platforms and applications.

In our opinion, one of the weaknesses inherent to all solutions that rely on the EVT is to base the analysis solely on the measured execution time (MET) of the target application. In a nutshell these EVT-based solutions splits the input set of METs into groups, analyzes the distribution of local maxima within these groups and then estimate how far the execution time may deviate from the average of that "distribution of the extremes". In practice though, recent monitoring techniques enable to collect very detailed information on the runtime. These monitoring tools can develop a complete profile of the execution of an application, including information such as the number of iterations in each loop, a complete function-call tree, the number of calls from and to every functions, the time spent in the body of every function, etc. In short, an execution trace may literally provide a wealth of information on each run of the application. From this viewpoint, solutions based on the EVT are barely scratching the surface of the immense potential knowledge hidden within the runtime traces, as they exploit only the information of the observed execution times of the analyzed function and brush all other information aside. Our research will mainly build on the method proposed in [5]. We will investigate how to explore and use this unexploited knowledge to estimate the probability of occurrence of a given execution time. We will investigate how all these information can be modeled as statistical variables and how copulas and state-of-the-art statistical techniques based on copula such as [6], [7] can be used to infer a probabilistic estimate of the WCET.

II. Our envisioned approach

There are many hindrances to the adoption of probabilistic concepts in WCET analysis methods and tools and our first objective will be to do an extensive review of the state-of-the-art probabilistic timing analysis techniques. We will set up clearly the objectives of pWCET estimation and the impacts and potential exploitation opportunities in industrial applications. Although all the details and procedures of our envisioned approach are not yet rigorously defined, we will roughly follow the following steps.

(1) Extraction of the data. In this step we will monitor and extract from the runtime execution traces the time spent, in number of clock cycles, within the body of each function that has been called from the function under analysis. For each run, we will thus have the value of the input parameters and the cumulative "self" execution time of every function called. Figure 1 shows an example of task code and the corresponding information obtained by running this code on the Kalray MPPA-256 (using only one VLIW core)[1]. Each row of the table characterizes one run of the main function and gives, from the left to the right column, the value of the input integer given to the main function, the number of cycles spent in the main, the "sqrt", and the "pow" functions[2] and finally, the sum of these last three values gives the total execution time of the main function.

```
Int main (int input) {
    long root;
    if (input%2==0) {
        root = sqrt(input);
    } else {
        root = pow(input, 3);
    }
    return (int) root;
}
```

Input	Main	Sqrt	Pow	ExecTime
10	19	84	-	113
11	19	-	51	70
16	19	83	-	112
19	19	-	51	70

Fig. 1. Our example main function and 4 execution traces.

(2) Model the execution times as random variables. After we compute the values of this table, we will model the execution time of each function by a random variable X_i for which we will derive the empirical cumulative distribution function $\text{ECDF}_i()$. That is, in our example we will define X_i ($i = 1, 2, 3$) as the execution time of the "main", the "sqrt", and the "pow" functions and we will compute the function $\text{ECDF}_i(x) = P[X_i \leq x]$ of each variable X_i, also called the *marginal* distribution of X_i. Note that we may also dedicate an extra column (and thus an extra random variable) for each input parameter of every function.

(3) Capture the dependency between the variables. The dependence structure of all the random variables X_i will be captured by fitting a copula to the data. In a nutshell, for a set of d dependent random variables, a copula C is defined as a function that captures the dependence between the d variables such that the joint probability $\text{ECDF}(x_1, x_2, \ldots, x_d)$ of observing simultaneously $X_i \leq x_i$, $\forall i$, can be directly inferred from the copula C and the marginal distributions $\text{ECDF}_i(x)$ of all X_i, $i \in [1, d]$. Although in [5] copulas have already been applied to capture the dependence between the execution time of the basic blocks of a program, we want to further explore how they can be used for the pWCET estimation.

[1]The function "sqrt" is a home-made function that compute the rounded down square root of an integer and "pow(x,3)" simply return x^3.

[2]These numbers represent the cumulative *self* execution time of each function.

(4) The not-so-clear step to capture rare cases. After evaluating the goodness-of-fit of the computed copula, we may want to expand the data to increase the chances of covering corner cases. For example, the extreme value theory could be applied to each variable X_i (or to their inputs) so that we capture extreme execution times for each individual function. Then, regression techniques based on copula such as [7] can be applied to approximate the value of the other variables for each of these extreme execution times and the newly approximated sets of execution times can then be added to the table as if they were observed runs. Another option is to use extreme-value copulas [8] to provide appropriate models for the dependence structure only between rare events.

(5) Derive the distribution of the WCET of the main function. The authors of [6] have proposed a fast algorithm that takes as input the marginal distribution $\text{ECDF}_i()$ of a set of variables X_i and a copula C that characterizes their dependence, and outputs a distribution of the sum of the X_i's. In our context this sum represents the desired distribution of the total execution time of the function F under analysis. Thus we intend to use this technique to finalize our analysis.

Acknowledgement

This work was partially supported by National Funds through FCT (Portuguese Foundation for Science and Technology) and by ERDF (European Regional Development Fund) through COMPETE (Operational Programme 'Thematic Factors of Competitiveness'), within project(s) FCOMP-01-0124-FEDER-037281 (CISTER); by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by National Funds through FCT (Portuguese Foundation for Science and Technology), within project ref. NORTE-07-0124-FEDER-000063 (BEST-CASE, New Frontiers); by National Funds through FCT and the EU ARTEMIS JU funding, within project(s) ref. ARTEMIS/0001/2013, JU grant nr. 621429 (EMC2); by the European Union, under the Seventh Framework Programme (FP7/2007-2013), grant agreement n 611016 (P-SOCRATES).

References

[1] L. Carnevali, A. Melani, L. Santinelli, and G. Lipari, "Probabilistic deadline miss analysis of real-time systems using regenerative transient analysis," in *proceedings of RTNS*, 2014.

[2] "http://www.proartis-project.eu/."

[3] "http://www.proxima-project.eu/."

[4] L. Santinelli, J. Morio, G. Dufour, and D. Jacquemart, "On the Sustainability of the Extreme Value Theory for WCET Estimation," in *14th International Workshop on WCET Analysis*, vol. 39, 2014, pp. 21–30.

[5] G. Bernat, A. Burns, and M. Newby, "Probabilistic timing analysis: An approach using copulas," *J. Embedded Comput.*, vol. 1, no. 2, pp. 179–194, April 2005.

[6] P. Arbenz, P. Embrechts, and G. Puccetti, "The GAEP algorithm for the fast computation of the distribution of a function of dependent random variables," *In Stochastics: An International Journal of Probability and Stochastic Processes*, pp. 1–29, October 2011.

[7] R. A. Parsa and S. A. Klugman, "Copula regression," *Variance*, vol. 5, no. 1, pp. 45–54, 2011.

[8] G. Gudendorf and J. Segers, "Extreme-value copulas," in *Copula Theory and Its Applications*, ser. Lecture Notes in Statistics. Springer Berlin Heidelberg, 2010, pp. 127–145.

Another look at the pWCET estimation problem

Vincent Nélis, Patrick Meumeu Yomsi, Luis Miguel Pinho, Guillem Bernat

CISTER Research Center

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8340509

E-mail: nelis@isep.ipp.pt, pamyo@isep.ipp.pt, lmp@isep.ipp.pt,

http://www.cister.isep.ipp.pt

Abstract