Work-in-Progress: Run-time pWCET Estimation and Quality Monitoring

Federico Reghenzani
DEIB - Politecnico di Milano
Milano, Italy
federico.reghenzani@polimi.it

Filippo Sciamanna
DEIB - Politecnico di Milano
Milano, Italy
filippo.sciamanna@polimi.it

William Fornaciari
DEIB - Politecnico di Milano
Milano, Italy
william.fornaciari@polimi.it

***Abstract*—Measurement-based WCET methods are reliable as long as we know all the application behaviors at design-time. In particular, we need to observe all the possible execution time behaviors before the actual system run-time to get a reliable estimation. This is, unfortunately, difficult to achieve. In this work, we propose an online monitoring framework with the goal to detect, at run-time, unexpected application behaviors and trigger the probabilistic-WCET re-estimation when needed. The online monitoring and estimation framework allows dealing with unexpected situations, such as never-seen applications, faults, or incorrect offline estimations. The proposed approach is tested with simulated and real data.**

***Index Terms*—probabilistic real-time, pWCET, runtime monitoring**

I. Introduction

Estimating the Worst-Case Execution Time (WCET) has become a critical problem for real-time systems because of the increasing complexity of modern computing architectures [1]. In fact, hardware manufacturers added complex features (such as multi-level caches and parallel cores) to overcome the performance barrier caused by the difficulties in increasing the clock frequency.

As a possible solution to the WCET problem, *probabilistic real-time* has been investigated in recent years [2]. In particular, the Measurement-Based Probabilistic Timing Analysis (MBPTA) technique is very attractive because it allows the estimation of a statistical distribution of the WCET directly from the observed execution time samples, without the need for complex platform and application models.

A. Background

Unlike the traditional distribution estimators – which are based on the Central Limit Theorem – MBPTA uses the Extreme Value Theory (EVT) to output the probabilistic-WCET (pWCET). This statistical theory generates, from a sequence of input samples, a distribution which is a good estimate of the tail of the distribution, i.e., where the real WCET is. The main result of EVT can be summarized in the following statement: independently of the original distribution of the input samples, the tail distribution is always a Generalized Extreme Value Distribution (GEVD) or a Generalized Pareto Distribution (GPD), which are asymptotically equivalent. Therefore, we can exploit EVT in MBPTA to estimate the pWCET just by observing the execution time samples of our system.

However, EVT estimates a correct distribution only if three hypotheses are satisfied. The first hypothesis is that the input samples must be independent and identically distributed (i.i.d.); the second hypothesis requires the original distribution to be in the Maximum Domain of Attraction (MDA) of the GEVD or GPD; the last hypothesis is the representativity hypothesis, i.e., that the input samples are representative of the phenomenon we are observing. The first and second hypotheses can be verified through specific statistical tests, while the last must be guaranteed by the experimenter. Further details on the hypotheses are available in the relevant literature, e.g., [3], [4]. The EVT process filters the input data (i.e., the observed execution times) via the Block-Maxima (BM) or Peak-over-Threshold (PoT) filter. The first splits the input samples into blocks of fixed size B and takes the maximum of each block. The second applies a threshold u to remove all the samples lower than the threshold. In this way, we save only the largest execution time samples, representing the behaviors near the WCET. The surviving samples are then the input of an estimator, which is usually the Maximum Likelihood Estimator (MLE) or the Probabilistic Weighted Moment (PWM). The estimator outputs the parameters of the GEVD or GPD (depending on if the selected filtering approach is, respectively, BM or PoT).

The pWCET distribution can be directly used by a probabilistic scheduler or can be transformed into a scalar WCET value and used by any traditional real-time scheduler. The transformation is possible by computing, via the Inverse-Cumulative Distribution Function (ICDF) of the estimated distribution, the value $\overline{WCET}$ at a given violation probability $\bar{p}$. Therefore, the arbitrarily low value $\bar{p}$, for instance, 10^{-9}, is the probability of observing an execution time larger than $\overline{WCET}$. The selection of $\bar{p}$ determines the trade-off between safety and pessimism of the WCET estimation.

B. Motivation and Contributions

Traditionally, the pWCET is estimated offline, and the resulting distribution (or the extracted WCET at a given violation probability level) is used online by the scheduler. The safety of the approach depends only on how the offline measurement campaign has been performed. In this short paper, we propose

This article has received funding from the European Union's Horizon 2020 Research and Innovation programme (RECIPE project, grant no. 801137)

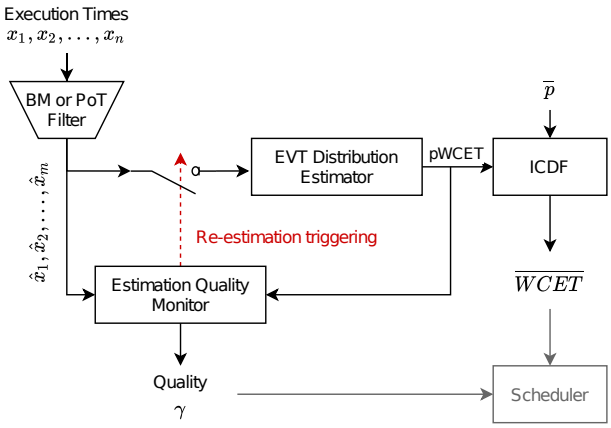

Fig. 1. The logical flow of the proposed approach.

a novel use of the probabilistic theory, i.e., to monitor the pWCET quality online and trigger the re-estimation of the pWCET distribution if the quality of the previous distribution is insufficient. This approach provides the WCET information even in these three cases usually not covered by traditional analyses: 1) the arrival of an unknown application for which we do not know the WCET[1], 2) the incorrect offline estimation of the pWCET, and 3) the reaction to unpredictable events (such as faults) which increase the execution time. All of these three cases are atypical of hard real-time systems because it is difficult to provide strong guarantees under these conditions in all the job executions. Hence, the target of this work is soft real-time or weakly hard real-time systems [6].

Our online framework monitors the quality of the pWCET estimation and triggers, if necessary, a new pWCET estimation. The output of this process is the new pWCET distribution (or the WCET value) and a quality parameter called γ, which represents the amount of trust on the current pWCET estimation. The scheduler can then use these parameters to make decisions on the task schedule. Recent approaches leveraged uncertain WCET values to schedule tasks [7]. We do not discuss how the scheduler can exploit γ, leaving any possibility open for future works.

II. Methodology

The overall flow of the proposed approach is depicted in Fig. 1. The observed execution times are the input of the BM or PoT approach described in Section I-A. The filtered dataset is fed to the EVT estimator and to the quality monitor. When a sufficient number of samples have been acquired so that the EVT estimator is able to produce a pWCET (eventually translated to a scalar $\overline{WCET}$), the estimator stops. The quality monitor continues to run and check the current distribution and, if needed, trigger a re-estimation of the pWCET. The output values of our approach are the pWCET/$\overline{WCET}$ and the quality γ. Formally, we write $\chi(t) \in \{\text{EST}, \text{MON}\}$ to identify, respectively, the estimation and monitoring phases. The system always starts in estimation phase, thus $\chi(0) = \text{EST}$.

[1]This is especially critical in high-performance computing real-time systems, where the applications are seldom known *a priori* [5].

A. pWCET estimation phase

Let us assume we initially measure a sequence of n execution time values $x_1, x_2, ..., x_n$. This sequence is fed to the BM or PoT filter and then to a pWCET estimator, which provides a distribution $\mathcal{G}$. The form of this distribution (GEVD or GPD) is irrelevant in this paper, and this approach works in both cases. However, for simplicity, in the following paragraphs, we refer to the PoT/GPD approach. Determining the precise number of samples n to obtain a good distribution is non-trivial. For this reason, we applied the following rationale: we continuously increase the number of samples until MLE is able to find a solution, the EVT hypotheses are satisfied according to the Probabilistic Predictability Index test [4], and the monitoring block returns a positive answer (see next Section II-B). Experimental data showed that the required number of samples is approximately $n \approx 500$. The initial estimation ($\chi(0) = \text{EST}$) can be performed online or offline. Instead, the subsequent online estimations start when $\chi(t-1) = \text{MON}$ and $\chi(t) = \text{EST}$. Once a new distribution $\mathcal{G}'$ is found, the system switches back to monitoring mode: $\chi(t+n) = \text{MON}$. After the estimation, the $\overline{WCET}$ is computed using the ICDF of the estimated distribution at a given probability level $\bar{p}$.

B. pWCET monitoring phase

When the system is in monitoring mode – i.e., $\chi(t) = \text{MON}$ – the sequence of filtered execution times is provided to the quality monitor, and the distribution estimator does not run. This monitor performs a Goodness-of-Fit (GoF) test to check whether the estimated distribution $\mathcal{G}$ matches the observed samples. The test analyzes a window of W execution time values, i.e., $\hat{x}_{m-W+1}, ..., \hat{x}_m$, where $\{\hat{x}_i\}$ is the sequence post-filtering of the input execution times $\{x_i\}$. In statistics, the output of an hypothesis test is the *statistic* $0 \leq S \leq 1$ and the *critical value* $0 < CV < 1$. If $S > CV$, then the test rejects the initial hypothesis (i.e., the data does not match the estimated distribution). To build a unified index, we define the quality γ at a given time t as follows:

$$\gamma(t) = \begin{cases} -\infty & \text{if } \chi(t) = \text{EST} \\ 1 - \frac{S(t)}{CV(t)} & \text{else} \end{cases} \quad (1)$$

A value $\gamma(t)$ near 1 means the best confidence on the execution time distribution, while a value near 0 a weak confidence. Instead, values $\gamma(t) < 0$ identify the non-confidence region, where the test is able to reject the pWCET distribution according to the execution time data. When the calculated quality is $\gamma(t) < 0$, then the re-estimation is triggered to learn a new pWCET distribution, i.e., $\chi(t+1) = \text{EST}$. During the re-estimation phase, the previous distribution must be considered unreliable (and, therefore, the quality parameter $\gamma(t) = -\infty$). Because the test does not run at every input sample of x_i (but every W filtered samples of $\{\hat{x}_i\}$), the quality value maintains the last value, i.e., $\gamma(\bar{t}) = \gamma(\bar{t}-1)$ if the test does not run at time $t = \bar{t}$.

The parameter W acts on the trade-off between test specificity and detection speed. Large values of W make the test

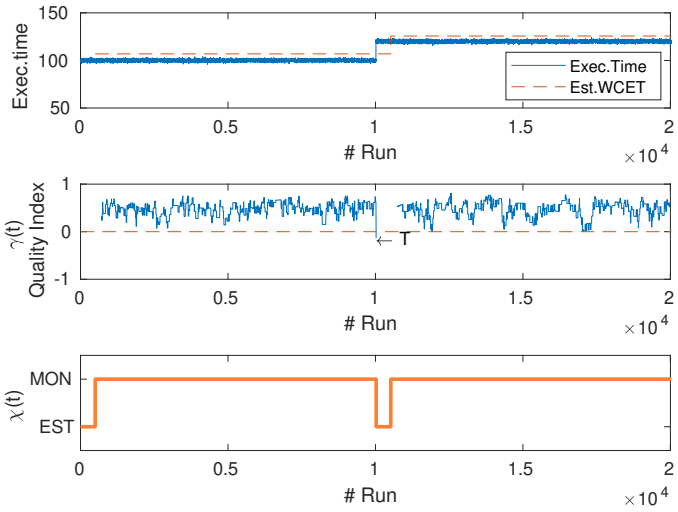

Fig. 2. Simulation 1: $N(100, 1)$ followed by $N(120, 1)$. The T symbol represents the triggering of a new pWCET re-estimation. The Estimated WCET threshold is set with violation probability $p = 10^{-9}$.

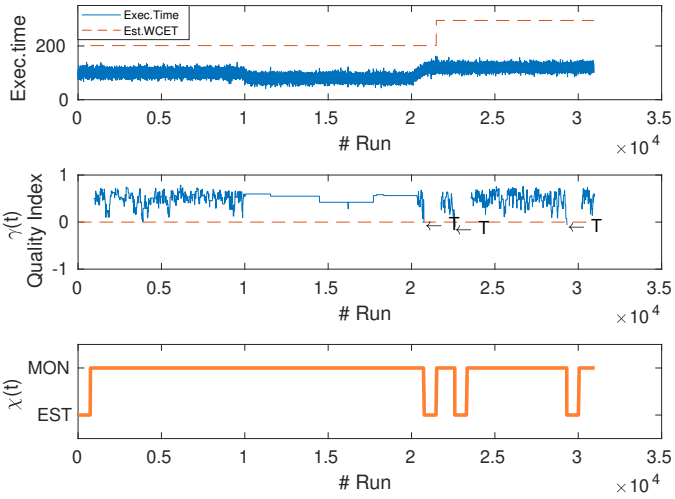

Fig. 3. Simulation 1: $\mathcal{N}(100, 1)$ followed by $\mathcal{N}(80, 1)$ and $\mathcal{N}(120, 1)$. The T symbol represents the triggering of a new pWCET re-estimation. The Estimated WCET threshold is set with violation probability $p = 10^{-9}$.

more stable and reduces the false-detection, but it increases the detection time. A small value W makes the test window smaller, thus fast, but it may experience many false-detections. Experimentally, a window size of $W = 20$ showed good results, but the value should be adjusted according to the operational requirements.

Many GoF tests exist. The most famous – and used in pWCET literature – are the Kolmogorov-Smirnov (KS) and the Anderson-Darling (AD). Due to performance issues of AD (see next Section II-C), in this work we used the KS test. The KS test is performed by computing its statistic S as follows [8]:

$$S = \sup_z \left| \frac{1}{W} \sum_{i=1}^{W} \mathbf{1}_{\hat{x}_i \leq z} - F_{\mathcal{G}}(z) \right| \tag{2}$$

where W is the window size, $\{\hat{x}_i\}$ is the filtered sequence in the window, $F_{\mathcal{G}}$ is the Cumulative Distribution Function (CDF) obtained from the parameters of the estimated distribution $\mathcal{G}$, and $\mathbf{1}_A$ is the indicator function: $\mathbf{1}_A = 1$ if A is true, 0 otherwise. The critical value depends on W only and can be calculated or taken from alread available tabular data.

C. Run-time sustainability

Since both the estimation and monitoring phases are executed at run-time, an analysis of their computational complexity is mandatory to assess the feasibility. It should be noted that the first pWCET estimation can also be performed offline and with a large dataset, thus reducing the probability of triggering the re-estimation. We will focus on the analysis of the subsequent run-time estimations.

The pWCET estimation is dominated by the complexity to run the estimator. MLE has a complexity of $O(n^2)$, while PWM has a complexity of $O(n)$. Since the number of samples is low (see Section II-A), this complexity is certainly affordable. For the experiments, we selected MLE. Regarding the monitoring phase, KS has a linear complexity $O(n)$, while AD has also $O(n)$ to compute S, but it requires $O(nm)$ to compute CV, with m a constant but large factor (on the order of 10 000). Therefore, we selected KS as the GoF run-time test because of its lower complexity than AD. In conclusion, all the algorithms have polynomial complexity, making our approach affordable at run-time. The open source `chronovise` tool [9] has been used as a reference for the algorithms' complexity.

III. Preliminary Results

To evaluate the ability to detect changes and re-estimate the pWCET distribution, we tested our approach in simulation (with a synthetic dataset) and with real execution times measured on a real platform. To simulate the three scenarios described in Section I-B, we introduced *application modes*, i.e., changes in the execution time behavior of the application. In all the analyses, we considered the PoT approach with the threshold u set as the 90^{th} percentile. The value u is updated every time the triggering of a new estimation occurs. The violation probability to compute the WCET is selected as $\bar{p} = 10^{-9}$. The choice of $\bar{p}$ affects only the plot and, in particular, the estimated WCET value. All the scripts and the datasets are available online[2].

A. Simulations

We considered two scenarios generated by the following synthetic datasets:

1) Data sampled from a Gaussian distribution $\mathcal{N}(100, 1)$ (10 000 samples) followed by a distribution Gaussian distribution with higher mean value: $\mathcal{N}(120, 1)$ (10 000 samples).
2) Data sampled from a Gaussian distribution $\mathcal{N}(100, 10)$, followed by a sharp transition to a Gaussian distribution $\mathcal{N}(80, 10)$, and followed by a slow transition to a Gaussian distribution $\mathcal{N}(120, 10)$.

[2]https://doi.org/10.5281/zenodo.5528004

The results are depicted, respectively, in Fig. 2 and 3.

In the first test, the distribution is estimated after $n = 496$ samples. From $t = 497$ the system switches to monitoring mode ($\chi(497) = \text{MON}$), until the distribution switch. Because the distribution switch is sharp, the test immediately detects a violation of the pWCET distribution ($\gamma < 0$), and the system switches back to the estimation mode. Then, the system goes back to monitoring mode ($\chi(10\,503) = \text{MON}$) until the end.

The second case is more dynamic than the first. The first distribution is found at[3] $t = 746$. After 10 000 samples, the distribution is replaced with the lower-mean distribution. It is possible to notice a lower variability in the quality index because a lower number of samples survive the PoT filter with threshold u, thus the KS window fills slower. The samples of the lower-mean distribution never triggered the pWCET re-estimation, as excepted. Finally, at $t = 20\,000$, the distribution with a higher mean is slowly introduced, and the KS test triggers the re-estimation at $t = 20\,777$, followed by another re-estimation at $t = 22\,607$ when the samples are at the new distribution. Then, at $t = 29\,328$, a false positive result of the test triggers a new distribution estimation, which outputs a similar distribution (visible from the estimated WCET).

B. Real-board experiments

Once we evaluated the performance of our approach on synthetic data, we tested it using the data collected from a real platform. We set up a NUCLEO-F746ZG, a development board from ST equipped with an STM32F746ZG MCU, and run a micro-benchmark derived from the popular WCET Mälardalen suite [10], that performs the insertion sort algorithm on a random array of size 1000. At the end of each run, we annotated the execution time thanks to the board's hardware timers. To simulate different execution modes, we run, in sequence, three input scenarios: 1) a completely random array; 2) a partially ordered array; 3) a reversed partially ordered array (i.e., near the worst-case).

The results are depicted in Fig. 4. In the first scenario $[0 - 5\,000)$ samples, the re-estimation is triggered after few samples ($t = 1179$) because of a drop of the quality index below zero. Then, the same estimation remains until the change to the third scenario, and the change is detected at $t = 10\,007$. Another re-estimation is triggered at $t = 11\,479$.

IV. Discussion and Conclusions

The preliminary results showed how the use of the test to detect a mode change in the software is effective. Clearly, if the mode change is sharp, as in Fig. 2 and 4, it cannot avoid the WCET overruns in the first samples before detection. However, if the change is progressive, the re-estimation is triggered before a WCET overrun occurs, like Fig. 3. The quality index γ can be used by the scheduler to determine the current level of confidence of the estimated WCET. This value appears to be noisy and sometimes triggers a re-estimation even if there is no application mode change. This is due to the value $W = 20$: increasing the window size would make the index more stable but also slower to detect mode changes. The W trade-off must be decided case by case and depending on the application requirements.

[3]To simplify the discussion all the time references are expressed per-run.

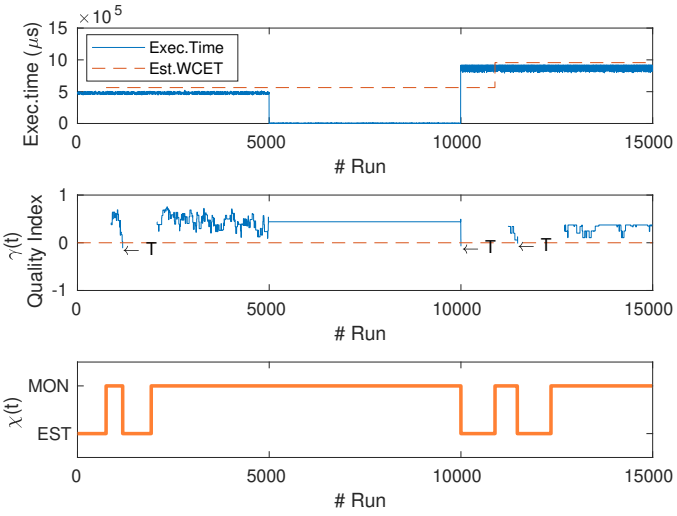

Fig. 4. Real world experiments. The T symbol represents the triggering of a new pWCET re-estimation. The Estimated WCET threshold is set with violation probability $p = 10^{-9}$.

The preliminary results are encouraging and show the good detection capability of the proposed monitor. Having a pWCET estimation monitor is essential to address the three scenarios described in Section I-B. Future works include the refinement of the estimation/monitor procedure and the exploitation for scheduling purposes of the index γ.

References

[1] P. Axer, R. Ernst, H. Falk, A. Girault, D. Grund, N. Guan, B. Jonsson, P. Marwedel, J. Reineke, C. Rochange, M. Sebastian, R. V. Hanxleden, R. Wilhelm, and W. Yi, "Building timing predictable embedded systems," *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 4, Mar. 2014.

[2] R. I. Davis and L. Cucu-Grosjean, "A survey of probabilistic timing analysis techniques for real-time systems," *LITES: Leibniz Transactions on Embedded Systems*, pp. 1–60, May 2019.

[3] L. Santinelli, F. Guet, and J. Morio, "Revising measurement-based probabilistic timing analysis," in *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2017, pp. 199–208.

[4] F. Reghenzani, G. Massari, and W. Fornaciari, "Probabilistic-WCET reliability: Statistical testing of EVT hypotheses," *Microprocessors and Microsystems*, vol. 77, p. 103135, 2020.

[5] F. Reghenzani, G. Massari, and W. Fornaciari, "Timing Predictability in High-Performance Computing With Probabilistic Real-Time," *IEEE Access*, vol. 8, pp. 208 566–208 582, 2020.

[6] G. Bernat, A. Burns, and A. Liamosi, "Weakly hard real-time systems," *IEEE Transactions on Computers*, vol. 50, no. 4, pp. 308–321, 2001.

[7] G. von der Brüggen, N. Piatkowski, K.-H. Chen, J.-J. Chen, and K. Morik, "Efficiently Approximating the Probability of Deadline Misses in Real-Time Systems," in *30th Euromicro Conference on Real-Time Systems (ECRTS 2018)*, vol. 106, Germany, 2018, pp. 6:1–6:22.

[8] F. J. J. Massey, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[9] F. Reghenzani, G. Massari, and W. Fornaciari, "chronovise: Measurement-based probabilistic timing analysis framework," *Journal of Open Source Software*, vol. 3, no. 28, p. 711, 2018.

[10] J. Gustafsson, A. Betts, A. Ermedahl, and B. Lisper, "The Mälardalen WCET Benchmarks: Past, Present And Future," in *10th International Workshop on Worst-Case Execution Time Analysis (WCET 2010)*, vol. 15, Germany, 2010, pp. 136–146.