# Beyond the Traditional Analyses and Resource Management in Real-Time Systems


**Federico Reghenzani**

**Abstract** The difficulties in estimating the Worst-Case Execution Time (WCET) of applications make the use of modern computing architectures limited in real-time systems. Critical embedded systems require the tasks of hard real-time applications to meet their deadlines, and formal proofs on the validity of this condition are usually required by certification authorities. In the last decade, researchers proposed the use of probabilistic measurement-based methods to estimate the WCET instead of traditional static methods. In this chapter, we summarize recent theoretical and quantitative results on the use of probabilistic approaches to estimate the WCET presented in the PhD thesis of the author, including possible exploitation scenarios, open challenges, and future directions.

## 1 Real-Time Systems and the WCET Problem

*Real-time systems* are computing systems in which the correctness of the computation does not depend only on the logic correctness—i.e., that the output is correctly produced—but also on the timing correctness—i.e., that the output is delivered within given time constraints. When such constraints must be satisfied at any time, and even a single violation is considered as a failure of the whole system, we call this system *hard* real-time. Vice versa, a *soft* real-time system allows the violation of timing constraints, provided that the violations do not occur *too often*.[1] Hard real-time systems are often *embedded systems*, and many of them are *mission-* or *safety-critical systems*. To mention a few examples: fly-by-wire computers of aircraft, the airbag control unit in a car, a pacemaker.

F. Reghenzani (✉)
Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Via Ponzio 34/5, 20133 Milano, Italy
e-mail: federico.reghenzani@polimi.it

[1] The term is voluntarily fuzzy. The exact definition of *too often* depends on the particular context.



L. Piroddi (ed.), *Special Topics in Information Technology*,
PoliMI SpringerBriefs, https://doi.org/10.1007/978-3-030-85918-3_6

## 1.1 Scheduling Analysis

The system software plays a critical role in guaranteeing that the applications can satisfy timing requirements. In particular, the scheduler must be designed to correctly prioritize the tasks so that all of them meet their timing deadlines. This design is usually in contrast with general-purpose systems, where the focus is more on the total throughput of the system rather than the response-time of the single task.

During the design phase of a real-time system, the *scheduling analysis* verifies if a given scheduling algorithm is able to schedule all the tasks, correctly satisfying the timing constraints. Each task $\tau_i$ is represented, in the simplest task model, by a set of parameters $\tau_i = (T_i, C_i, D_i)$, where $T_i$ is the period or inter-arrival time, $C_i$ is the Worst-Case Execution Time (WCET), and $D_i$ is the relative deadline. A task is an abstract entity periodically (with period $T_i$) or aperiodically (with minimum inter-arrival time $T_i$) activated. When a task is activated, it starts a new job. The job is the single unit of computation that performs the function the task is developed to. The job has a duration of maximum $C_i$ and must complete its execution by the deadline $D_i$ relative to the activation time.

Computing a correct Worst-Case Execution Time (WCET) is then essential to perform a legitimate scheduling analysis and the consequent claims of satisfying hard real-time constraints. The traditional way to estimate the WCET is to use the information on the hardware architecture combined with the software description (usually in the form of a control-flow graph) and derive the worst-case conditions leading to the WCET. It is not usually possible to compute the exact WCET, but only an approximation and, in particular, an over-estimation of it. A pessimistic over-estimation guarantees, in any case, a correct scheduling analysis.

## 1.2 The WCET Problem in Modern Architectures

Unfortunately, the evolution of hardware, particularly the processor, towards more complex computing architectures makes the computation of the WCET extremely difficult, or the estimated WCET is so pessimistic that it becomes unusable in practice. This is due to the advanced features added to respond to the increasing computational power demand of modern applications, such as machine learning, image vision, etc. To provide a trivial example, let us consider a multi-level cache hierarchy, very common in modern processors: forecasting a cache miss/hit of the single memory access of a program is non-trivial, and assuming all the memory accesses as miss makes the WCET extremely pessimistic (and the presence of cache substantially useless for the scheduling analysis standpoint).

The pervasive use of Commercial-off-the-Shelf (COTS) components in real-time applications is challenging because it adds another layer of complexity in WCET estimation, due to the numerous sources of unpredictability affecting these platforms [6]. In fact, COTS platforms are built with average performance in mind and are not intended to provide a timing model able to compute the WCET easily.

## 2 Probabilistic Real-Time Computing

A possible solution to the WCET estimation problem is the so-called *probabilistic real-time computing*: instead of a scalar value for the WCET, a statistical distribution is provided. This idea originates in the early 2000s from the papers by Edgar et al. [8] and Bernat et al. [1]. The statistical distribution can be estimated with two methods: a *Static Probabilistic Timing Analysis (SPTA)* and a *Measurement-Based Probabilistic Timing Analysis (MBPTA)*. The former estimates the distribution by looking at the same information available to the traditional static (but deterministic) analysis. However, it suffers the same problems of the deterministic analysis and, consequently, it did not spark too much interest in the scientific community. Vice versa, MBPTA is very attractive, thanks to its simplicity, and therefore it is the subject of this work. Two recent comprehensive surveys [5, 7] provide a general overview of probabilistic real-time WCET analyses research of the last years.

### *2.1 The Probabilistic-WCET*

MBPTA approaches apply a statistical procedure to a finite sequence of random variables $X_1, X_2, \ldots, X_n$, representing the execution time of our task under analysis. The output of such procedures is a statistical distribution called *probabilistic-WCET (pWCET)*, and it is usually expressed with its Complementary Cumulative Distribution Function (CCDF):

$$p = P(X \geq \bar{C}) = 1 - F_X(\bar{C})$$

The probability $p$, called *violation probability*, represents the probability of observing an execution time larger than $\bar{C}$. The random variable $X$ represents a generic random variable of the process by assuming that the random variables are identically distributed. The experimenter can select either $\bar{C}$ or $p$ and accordingly compute the other value: it is possible to estimate the violation probability $p$ given a WCET $\bar{C}$ or, vice versa, estimate the WCET $\bar{C}$ given a target violation probability $p$. The latter option is computed by using the Inverse Cumulative Distribution Function (ICDF).

Provided that the pWCET distribution is correctly computed and it represents the real distribution of execution times, it is reasonable to claim that we are compliant with safety-critical processes: a probability of violation, corresponding to the real one, would be just another term in the failure analysis of safety-critical systems. However, guaranteeing that the probability of violation is correctly computed is non-trivial and represents the major obstacle to probabilistic real-time use in the current industrial system. We will discuss this issue in Sect. 3.

### 2.2 Extreme Value Theory

The statistical theory called *Extreme Value Theory (EVT)* provides a reliable way to estimate the pWCET from the observations of the execution time. The mathematical details on this statistical theory are omitted here due to space limitations, but it can be found in the thesis [10] or in specialized books [4]. The main EVT result is the *Fisher-Tippett-Gnedenko theorem* which states that the distribution tail of an observed phenomenon—i.e., in our case, the maxima of execution times—converges to the Gumbel, the Weibull, or the Fréchet distribution, independently from the original distribution of the measured values. This is a key property because we can estimate the pWCET without knowledge of the original distribution of the execution times. Moreover, the three distributions are actually particular cases of a more general distribution: the *Generalized Extreme Value Distribution (GEVD)*, which is characterized by the following Cumulative Distribution Function (CDF):

$$G(x) = \begin{cases} e^{-e^{\frac{x-\mu}{\sigma}}} & \xi = 0 \\ e^{-[1+\xi(\frac{x-\mu}{\sigma})]^{-1/\xi}} & \xi \neq 0 \end{cases} \tag{1}$$

The GEVD distribution $\mathcal{G}(\mu, \sigma, \xi)$ is parameterized by the *location* parameter $\mu$, the *scale* parameter $\sigma$, and the *shape* parameter $\xi$. The latter determines which subclass the distribution is ($\xi = 0$: Gumbel, $\xi < 0$: Weibull, or $\xi > 0$: Fréchet). Equivalently, it is possible to use the *Generalized Pareto Distribution (GPD)*. This theory is applicable provided that three conditions are satisfied:

- The input measurements are *independent and identical distributed (i.i.d.)*;
- The real distribution is in the *Maximum Domain of Attraction (MDA)* of an EVT distribution;
- The measurements used in analysis are *representative* of the real execution.

The next section focuses on explaining how to verify that these conditions are valid.

## 3 Uncertainty Estimation

The estimation of the pWCET is performed via a proper set of algorithmic steps. The overall process leading to estimate the pWCET distribution is, in fact, more sophisticated than just running a distribution estimator. In particular:

1. The sequence of execution time measurements $X_1, X_2, \ldots, X_n$ is tested to verify the validity of the i.i.d. hypothesis;
2. A filtering technique is applied to the time measurements to capture only the "tail part" of th distribution;
3. The remaining samples are used to feed a distribution estimator (such as Maximum Likelihood Estimator or Probabilistic Weighted Moment);

4. A Goodness-of-Fit (GoF) test is performed to verify the correspondence between the estimate distribution and the original set of samples (implicitly verifying also the MDA hypothesis).

### 3.1 The Importance of Statistical Testing

In the context of the aforementioned estimation process, we can distinguish two types of statistical tests we need: (1) a test to verify the i.i.d. hypothesis and (2) the GoF test for the final check of the distribution. However, statistical testing is subject to errors due to the obvious finiteness of the input measurements, and this may impact the final reliability of the obtained pWCET [15]. For this reason, we created two mathematical tools that help an experimenter to assess the quality and reliability of the obtained pWCET: The Probabilistic Predictability Index to check the i.i.d. hypothesis and the Region of Acceptance to verify the Goodness-of-Fit test.

### 3.2 The Probabilistic Predictability Index

The requirement of the i.i.d. hypothesis is actually stricter than needed, and it is possible to split it into three sub-hypothesis [17]: stationarity, short-range independence, and long-range independence. For each of these three categories, we selected statistical tests capable of identifying a violation in these properties [12, 14]: KPSS, BDS, and R/S tests. These tests can be used to evaluate the ability of hardware and software to comply with the three sub-hypotheses of the i.i.d. hypothesis. However, comparing different solutions using separate tests is non-trivial, mainly due to the effect on the significance level $\alpha$. For this reason, we developed an index called *Probabilistic Predictability Index (PPI)* [12], which maintains the statistical properties of the original tests while providing a convenient way to compare hardware/software solutions.

The PPI is a number in the range (0, 1) calculated with the following equation:

$$PPI := \begin{cases} \min_{\forall i} f_i(D_i) \cdot \prod_{i \in v^*} [1 - (CV_{PPI} - f_i(D_i))] & v \neq \varnothing \\ \frac{1}{3} \sum_{\forall i} f_i(D_i) & v = \varnothing \end{cases} \tag{2}$$

where

- $f_i$ are the following functions:

– $f_{KPSS}(x) = e^{-K_{KPSS} \cdot x}$;
– $f_{BDS}(x) = e^{-K_{BDS} \cdot |x|}$;
– $f_{R/S}(x) = e^{-K_{R/S} \cdot x}$
– with $k_{KPSS} = \frac{1}{4}$, $k_{BDS} = -\frac{\log f_{KPSS}(CV_{KPSS})}{|CV_{BDS}|}$, $k_{R/S} = -\frac{\log f_{KPSS}(CV_{KPSS})}{CV_{R/S}}$

- $D_i$ is the value of the *statistic* of each test computed via the original formulas for the KPSS, BDS, and R/S tests;

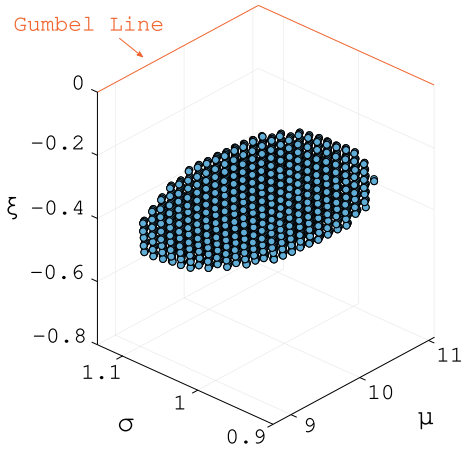


**Fig. 1** An example of region of acceptance depicted with the three axes matching the three GEV parameters

- $CV_{PPI}$ is the critical value for the PPI test, computed as $CV_{PPI} = f_{KPSS}(CV_{KPSS})$
- $v$ is the violation set, i.e. $v = \{i | f_i(D_i) < CV_{PPI}\}$, and $v^*$ is the violation set without the minimum, i.e. $v^* = \{v \setminus \arg\min_{\forall i} f_i(D_i)\}$.

A full description of the steps to derive this formulation is available in the thesis [10]. When this number approaches 1, then the input time series appears to be compliant with the i.i.d. hypothesis, while when approaches 0 the series is non-compliant. The decision value is set at $CV_{PPI}$.

### 3.3 Region of Acceptance

The output of the estimator routine of step 3 of the EVT process is the set of parameters of our distribution. Let us write $(\overline{\mu}, \overline{\sigma}, \overline{\xi})$ the tuple of estimated parameters. The tuple $(\mu^{\circledast}, \sigma^{\circledast}, \xi^{\circledast})$ is the exact, but unknown, distribution. The goal of the GoF test of step 4 of the EVT process is to identify whether the distance between these two tuples is *too large* to compromise the safety of the final pWCET. The output of the GoF test is a region in the parameters space identifying the limits of the acceptable distribution parameters. We call this cloud of point, depicted in Fig. 1, *Region of Acceptance (RoA)*. If the estimated parameter tuple $(\overline{\mu}, \overline{\sigma}, \overline{\xi})$ is inside this region, then the pWCET distribution can be safely accepted, according to the confidence provided by the test.

We know [16, Theorem 3.6] that the exact distribution $(\mu^{\circledast}, \sigma^{\circledast}, \xi^{\circledast})$ is inside the RoA or at its border. For this reason, even considering the limitation of the estimator and the statistical test, we can find a distribution that over-estimates all the others by looking at the boundaries of this multi-dimensional space. Therefore, given a region $R$, a violation probability $\overline{p}$, and a point $\hat{P} \in R$ such that $\hat{P}$ is the point that maximizes the CCDF of the region, then either $\hat{P} = (\mu^{\circledast}, \sigma^{\circledast}, \xi^{\circledast})$ or the pWCET

associated to $\hat{P}$ overestimates the real pWCET given by $(\mu^{\circledast}, \sigma^{\circledast}, \xi^{\circledast})$ at violation probability $\overline{p}$.

The described result is only a first step in the analysis of the RoA. Then, it is possible to derive several mathematical procedures to explore the RoA (described in [16]), dealing with the trade-off pessimism and reliability of the final pWCET result.

## 4 Exploiting Probabilistic Real-Time

Probabilistic real-time is still affected by numerous open challenges and problems (subsequently explained in Sect. 5) to consider it ready to use for WCET estimation of tasks in safety-critical systems. However, we can already trust it in other cases, such as Mixed-Criticality, High-Performance Computing, or to estimate the worst-case energy. The following paragraphs briefly describe these three scenarios and how probabilistic real-time plays a role in them.

### *4.1 Mixed-Criticality*

*Mixed-Criticality Systems* are systems providing a mix of critical functions not all at the same criticality level. In the context of real-time computing, the traditional task model is the one proposed by Vestal in 2007 [18]. Multiple values of WCET are assigned to each task depending on its criticality: higher criticality tasks have several WCETs, depending on the level of assurance used to estimate it. The WCET with the highest level of assurance is usually the one computed with the traditional static method, which is safe but very pessimistic. We can exploit probabilistic information in mixed-criticality systems to: (1) estimate the non-highest level of assurance WCETs with reasonable accuracy, and (2) improve non-functional requirements (such as energy consumption) while still guaranteeing the hard deadline with the statically computed WCET.

In the first case, the idea is to exploit the pWCET to estimate the values of $C_i^j$, which is the execution time of the task $\tau_i$ at the assurance level $j$. At the highest assurance level ($j = \texttt{HI}$), $C_i^{\texttt{HI}}$ is computed with safe static methods. Then, for lower assurance levels, we can use the pWCET by setting different values for the violation probability $p$. Even in the case of the pWCET being incorrect, at least the correctness of `HI`-criticality level tasks is guaranteed. For further details on mixed-criticality scheduling, refer to the Burns et al. survey [3].

In the second case, the timing properties are guaranteed with the WCET estimated with static analysis, while secondary properties, such as energy consumption, temperature, etc., are optimized based on probabilistic information. In this way, an incorrect estimation of the distribution does not invalidate the safety properties. This is an active area of research and subject of a recent work [2].

## 4.2 High-Performance Computing

High-Performance Computing (HPC) clusters are composed of hundreds or thousands of general-purpose servers. What is the relation with embedded safety-critical systems we are talking about in this chapter? Because time-critical applications are also emerging on these systems [13], such as medical imaging, natural disaster prediction, or structures monitoring. All of these application categories require a large amount of computational power (consequently the need to run on HPC clusters) and timing guarantees on the results. Clearly, the deadlines, in this case, are orders of magnitude larger than the deadline in the embedded system case, but the problem of scheduling real-time workload remains the same.

The static computation of WCET on HPC hardware and software is practically impossible. This is due to the complex general-purpose architecture of the single machine and of the network. Exploiting MBPTA and the resulting pWCET can solve the problem because it does not require perfect modeling of hardware and software. However, the EVT hypotheses must be satisfied. First preliminary results [13] showed that the EVT hypotheses could be satisfied with due safety technical shrewdness. The presence of heterogeneous hardware (such as GPGPU computing), which is exploding in HPC in the last years, exhibited an improvement in compliance of EVT hypotheses compared to a full-homogeneous scenario.

## 4.3 Energy Estimations

Even more difficult than the WCET problem, it is the *Worst-Case Energy Consumption (WCEC)* problem. The WCEC is necessary for some critical systems having an energy budget to satisfy as a *functional* requirement. Typical scenarios include systems powered by energy-harvesting devices (e.g., solar panels), such as embedded systems located in remote regions not having access to the power grid or satellites harvesting energy only when exposed to the sun. In these situations, the WCEC is needed to formally verify that the system can survive the period without a stable energy source. However, estimating the WCEC requires not only the WCET estimation as input but also the perfect model of the hardware in terms of power consumption.

To overcome the WCEC estimation problem, we proposed [11] to use the same theory used for the pWCET but directly applied to energy (or power) measurements of our system. In this way, similarly to MBPTA, we can hide the complexity of the power/energy model of the system and exploits EVT to obtain a pWCEC estimation. The same pWCET limitations and hypotheses, which must be satisfied, exist. However, differently from the WCET case, in the WCEC case, the choice of pWCEC is almost mandatory due to the difficulties in estimating even a pessimistic static WCEC.

## 5 Current Open Challenges and Future Directions

In the previous section, we discussed the i.i.d. and MDA hypotheses and how to verify them with statistical testing. We have not yet discussed about the third EVT hypothesis: representativity. The *representativity* informally means that we observed a *sufficient amount* of application behaviors to be sure we minimize the epistemic uncertainty of the phenomenon under statistical analysis. For example, suppose the control-flow graph of a program under analysis has a branch that is never taken during the measurement campaign. In that case, no statistical technique can infer something it cannot see. Representativity is the major obstacle in certifying safety-critical software by exploiting probabilistic real-time. The presence of the probability itself is not a risk for the safety, provided that the probability is perfectly computed. In such a scenario, the probability is added as another term of the failure analysis (like a hardware failure). A more in-depth discussion on representativity is available in the thesis [10].

Besides representativity, several other challenges related to probabilistic real-time are open [9], especially on uncertainty estimation and how to build hardware architectures able to comply with the EVT hypotheses. Therefore, the research is still very active and presents numerous challenges to address in the next years.

In particular, we identified three research fronts we believe to be promising subjects of future research on probabilistic real-time for the next years:

- Continuing the study of the theory behind the pWCET and its safety, with a particular focus on the representativity problem and uncertainty estimation;
- How to exploit, in a different manner, the current probabilistic real-time theory. This includes a more in-depth analysis of pWCET in HPC clusters, the optimization of non-functional metrics (such as energy, power, reliability, temperature, etc.), and monitoring application behaviors via statistical techniques;
- Dealing with fault-tolerance requirements, for example, by allowing tasks to re-execute if a failure occurs. In this scenario, the pWCET information can be exploited to verify the probability of transient faults to happen and perform probabilistic scheduling on the task re-execution.

The scientific community is divided over the future of probabilistic real-time: the barrier of representativity is seen as insuperable for many. However, static WCET analyses are also stuck for years. Our opinion is that it is worth continuing to investigate the probabilistic theory, in particular to quantify how much we can rely on its output—i.e., the pWCET—and to discover other use-cases of the probabilistic information.

## References


1. G. Bernat, A. Colin, S.M. Petters, WCET analysis of probabilistic hard real-time systems, in *23rd IEEE Real-Time Systems Symposium, 2002. RTSS 2002* (IEEE, 2002), pp. 279–288
2. A. Bhuiyan, F. Reghenzani, W. Fornaciari, Z. Guo, Optimizing energy in non-preemptive mixed-criticality scheduling by exploiting probabilistic information. IEEE Trans. Comput.-Aided Des. Integrated Circ. Syst. **39**(11), 3906–3917 (2020)
3. A. Burns, R. Davis. Mixed criticality systems-a review. *Department of Computer Science, University of York, Tech. Rep* (2013), pp. 1–69
4. E. Castillo, Extreme Value Theory in Engineering (Elsevier, Statistical Modeling and Decision Science, 2012)
5. F.J. Cazorla, L. Kosmidis, E. Mezzetti, C. Hernandez, J. Abella, T. Vardanega, Probabilistic worst-case timing analysis: taxonomy and comprehensive survey. *ACM Comput. Surv.*, 52(1):14:1–14:35, February 2019
6. D. Dasari, B. Akesson, V. Nélis, M.A. Awan, S.M. Petters. Identifying the sources of unpredictability in COTS-based multicore, in *International Symposium on Industrial Embedded Systems* (IEEE, 2013), pp. 39–48
7. R. Davis, L. Cucu-Grosjean, A survey of probabilistic timing analysis techniques for real-time systems. Leibniz Trans. Embedded Syst. **6**(1), 03–1–03:60 (2019)
8. S. Edgar, A. Burns, Statistical analysis of WCET for scheduling, in *Proceedings 22nd IEEE Real-Time Systems Symposium (RTSS 2001) (Cat. No.01PR1420)* (2001), pp. 215–224
9. S. Jiménez Gil, I. Bate, G. Lima, L. Santinelli, A. Gogonel, L. Cucu-Grosjean, Open challenges for probabilistic measurement-based worst-case execution time. IEEE Embedded Syst. Lett. **9**(3), 69–72 (2017)
10. F. Reghenzani, *Beyond the Traditional Analyses and Resource Management in Real-Time Systems*. PhD thesis, Politecnico di Milano, Jan 2021. Advisor: Prof. William Fornaciari
11. F. Reghenzani, G. Massari, W. Fornaciari, A probabilistic approach to energy-constrained mixed-criticality systems, in *2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)* (2019), pp. 1–6
12. F. Reghenzani, G. Massari, W. Fornaciari, Probabilistic-WCET reliability: statistical testing of EVT hypotheses. Microproc. Microsyst. 77, 103135 (2020)
13. F. Reghenzani, G. Massari, W. Fornaciari, Timing predictability in high-performance computing with probabilistic real-time. IEEE Access 8, 208566–208582 (2020)
14. F. Reghenzani, G. Massari, W. Fornaciari, A. Galimberti, Probabilistic-WCET reliability: on the experimental validation of EVT hypotheses, in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, COINS '19, New York, NY, USA (2019), pp. 229–234. Association for Computing Machinery
15. F. Reghenzani, L. Santinelli, W. Fornaciari, Why statistical power matters for probabilistic real-time: work-in-progress, in *Proceedings of the International Conference on Embedded Software Companion*, EMSOFT '19, New York, NY, USA (2019). Association for Computing Machinery
16. F. Reghenzani, L. Santinelli, W. Fornaciari, Dealing with uncertainty in pWCET estimations. ACM Trans. Embed. Comput. Syst. **19**(5) (2020)
17. L. Santinelli, J. Morio, G. Dufour, D. Jacquemart, On the sustainability of the extreme value theory for WCET estimation, in *14th International Workshop on Worst-Case Execution Time Analysis*, volume 39 of *OpenAccess Series in Informatics (OASIcs)*, pages 21–30, Germany, 2014. Schloss Dagstuhl–Leibniz
18. 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)* (2007), pp. 239–243