Measurement-Based Worst-Case Execution Time Estimation Using the Coefficient of Variation

JAUME ABELLA, Barcelona Supercomputing Center (BSC)
MARIA PADILLA, Universitat Autònoma de Barcelona, Departament de Matemàtiques
JOAN DEL CASTILLO, Universitat Autònoma de Barcelona, Departament de Matemàtiques
FRANCISCO J. CAZORLA, IIIA-CSIC and BSC

Extreme Value Theory (EVT) has been historically used in domains such as finance and hydrology to model worst-case events (e.g. major stock market incidences). EVT takes as input a sample of the distribution of the variable to model and fits the tail of that sample to either the Generalised Extreme Value (GEV) or the Generalised Pareto Distribution (GPD). Recently, EVT has become popular in real-time systems to derive worst-case execution time (WCET) estimates of programs. However, the application of EVT is not straightforward and requires a detailed analysis of, and customisation for, the particular problem at hand. In this paper we tailor the application of EVT to timing analysis. To that end (1) we analyse the response time of different hardware resources (e.g. cache memories) and identify those that may lead to radically different types of execution time distributions. (2) We show that one of these distributions, known as mixture distribution, causes problems in the use of EVT. In particular, mixture distributions challenge not only properly selecting GEV/GPD parameters (i.e., location, scale and shape), but also determining the size of the sample to ensure that enough tail values are passed to EVT and that only tail values are used by EVT to fit GEV/GPD. Failing to select these parameters has a negative impact on the quality of the derived WCET estimates. We tackle these problems, by (3) proposing *MBPTA-CV*, a new mixture-distribution aware, WCET-suited MBPTA method that builds upon recent EVT developments in other fields (e.g. finance) to automatically select the distribution parameters that best fit the maxima of the observed execution times. Our results on a simulation environment and a real board show that MBPTA-CV produces high-quality WCET estimates.

CCS Concepts: • **Computer systems organization** → *Embedded software; Real-time system specification;*

Additional Key Words and Phrases: Worst-case execution time, extreme value theory, probabilistic analysis, randomisation

ACM Reference Format:
Jaume Abella, Maria Padilla, Joan del Castillo, Francisco J. Cazorla, 2016. Measurement-Based Worst-Case Execution Time Estimation Using the Coefficient of Variation. *xxx* 0, 0, Article 0 (0), 25 pages.
DOI: xxx

1. INTRODUCTION

Timing is one of the most important non-functional properties for critical real-time systems. Timing analysis focuses on deriving tight and reliable Worst-Case Execution Time (WCET) estimates. These are needed for verification and validation – mandatory steps in the design of critical systems. Further, WCET estimates are required to be as tight as possible in order to optimise the system's size, weight, power requirements and cost. The relevance of timing analysis is confirmed by the recent funding of several projects, involving leading industries from the aerospace, railway and automotive domains (among others), with focus on increasingly complex hardware including parMERASA (http://www.parmerasa.eu/), PROXIMA (http://proxima-project.eu/), ARGO (http://www.argo-project.eu/), PRET (https://chess.eecs.berkeley.edu/pret/), DREAMS (http://www.uni-siegen.de/dreams/home/).

Until recently, simple 8-bit and 16-bit microcontrollers have been the main choice for critical real-time systems. However, the increasing performance requirements across segments like automotive, space and aerospace [Buttle 2012; Edelin 2009; Owens 2015; Patte and Lefftz 2011] can only be realistically achieved using higher performance processors. Such processors include fea-

The research leading to these results has received funding from the European Community's FP7 [FP7/2007-2013] under the PROXIMA Project (www.proxima-project.eu), grant agreement no 611085. This work has also been partially supported by the Spanish Ministry of Science and Innovation under grant TIN2015-65316-P and the HiPEAC Network of Excellence. Jaume Abella has been partially supported by the Ministry of Economy and Competitiveness under Ramon y Cajal postdoctoral fellowship number RYC-2013-14717. We thank Paul Caheny and Enrico Mezzetti for proofreading this manuscript.
.
DOI: xxx

tures that have been used for decades in the high-performance domain, like caches and pipelined cores. Those features, however, pose new challenges to analysis techniques [Wilhelm et al. 2008]. For instance, deriving accurate timing models of the hardware, as needed for static timing analysis, is challenged by the increasing complexity of modern systems' internal state: while each hardware component may have deterministic behaviour, their complex relation is generally hard to track and model [Mezzetti and Vardanega 2011]. Further, the lack of details on processor internals, due to IP restrictions or incomplete specifications, limits the information available for analytical timing models. Those models, therefore, resort to worst-case assumptions to account for the unknown, leading to pessimistic predictions [Mezzetti and Vardanega 2011]. Likewise, measurement-based approaches find difficulties in deriving evidence that worst-case system behaviour is captured in the measurement runs, negatively impacting the confidence that users have on the derived timing bounds [Paulitsch et al. 2015].

Probabilistic timing analysis (PTA) [Bernat et al. 2002; Hansen et al. 2009; Cucu-Grosjean et al. 2012; Lu et al. 2012; Cucu et al. 2013; Lu et al. 2011; Cazorla et al. 2013a] is a family of techniques suitable for processors including high-performance hardware features. PTA provides probabilistic WCET (pWCET) estimates that upper bound the residual risk of exceeding an execution time bound with an arbitrarily low probability. PTA approaches do not account for the absolute WCET when its associated residual risk (also referred to as exceedance probability) is below an acceptable threshold. That threshold relates to the failure rates dictated by the corresponding integrity levels in the domain-specific standards, such as ARP4761 [SAE International 2001] in avionics and ISO26262 [International Organization for Standardization 2009] in automotive. We focus on the measurement-based variant of PTA, MBPTA [Cucu-Grosjean et al. 2012; Cucu et al. 2013], in reason of its affinity with industrial practice where the dominant type of analysis builds upon measurements [Wilhelm et al. 2008].

MBPTA uses Extreme Value Theory [Feller 1996; Kotz and Nadarajah 2000] (EVT) as a building block. When applied to timing analysis, EVT relies on a *sample* of execution time observations captured in the *analysis-time* tests. This sample – whose size is maintained in the range of thousands to keep the analysis cost affordable – is used to predict the timing behaviour of tasks *during operation* for small exceedance probabilities, e.g. $[10^{-6}, 10^{-15}]$ [Wartel et al. 2015]. Noticeably, EVT does not deal with the *representativeness* [Abella et al. 2014b; Reineke 2014; Milutinovic et al. 2016] of the execution time observations passed to it and collected during the analysis stage. Representativeness ensures that analysis-time observations capture the impact of those hardware/software events (e.g. cache misses) affecting the execution time of the application during system operation. Hence, EVT considers the system from which data is captured – the computing platform in our case – as a black box and does not deal with the representativeness of the *execution conditions* under which its input data are collected. However, representativeness is vital for correctly applying EVT, which requires ensuring that analysis-time observations can be used to derive pWCET estimates that hold during operation.

Once representativeness is achieved, EVT can be used to derive pWCET estimates that hold during operation. However, EVT is a generic statistical tool whose use must be tailored to the particular problem at hand, as has been done in other fields like finance [Gilli and Këllezi 2006] and hydrology [Clarke 2002]. Failing to do so may result in low-quality EVT predictions, i.e. not solving the problem addressed. In this respect, some MBPTA techniques [Edgar and Burns 2001; Hansen et al. 2009; Bernat and Newby 2006; Santinelli et al. 2014] do not provide means to relate analysis conditions to those during operation, making it more difficult to assess whether the derived pWCET estimates upper bound execution time distributions at operation [Cazorla et al. 2013b]. Other techniques [Cucu-Grosjean et al. 2012; Cucu et al. 2013] allow relating WCET estimates with operation conditions, but they apply EVT without considering the particular characteristics of the execution time distributions modelled. That, too, challenges the quality of the WCET estimates obtained since – as shown in this paper – some distributions need EVT to be applied only under specific conditions.

Contribution. We contribute to MBPTA's landscape with a technique – and its implementation – that captures the particular shape of the observed execution times and works in conjunction with existing representativeness methods. In particular, our contributions cover three elements. (1) We perform an analysis of the execution time distributions of real-time programs that are passed as input to EVT. We show how EVT application is challenged when the execution time distribution under analysis is not *well-behaved*, meaning that it presents a step-like shape. This type of distribution is referred to in probability and statistics as *mixture distribution*. In particular, the task of fitting EVT parameters becomes more complex as it heavily depends on the characteristics of the execution time distribution. Building on the above analysis, (2) we show how to tailor the parameter-selection process of EVT to the problem of WCET estimation. As a final step, (3) we describe a method that successfully implements the proposed parameter selection, effectively helping to provide high-quality pWCET estimates. In more detail our contributions are as follows:

(1) *Distribution Analysis*. Our work covers two parts. First, we analyse the influence of different hardware resources on the programs' execution time distributions. In particular, the response time of some resources may lead to either *well-behaved* or *mixture* distributions. While well-behaved distributions can be processed by EVT in an straightforward manner, mixture ones challenge EVT application. And second, we provide a detailed analysis of the constraints imposed on real-time programs – e.g. upper-bounded loop bounds and finite recursion level – and how they affect the programs' probabilistic execution time distribution. This is fundamental to understanding which constraints of real-time programs' shape their timing behaviour. Such specific timing behaviour is the basis for some of the assumptions made in this paper to tailor EVT application. This is also fundamental to demonstrating the validity of the method for certification.
(2) *EVT tailoring*. Based on the analysis in (1), we tailor EVT to pWCET estimation. In particular, we find the GEV/GPD parameters (i.e. location, scale and shape) that best fit the tail of the input distribution. We further define the scenarios in which EVT parameters must be modified in a pWCET-centric manner and those scenarios in which it is required to increase the sample size until enough tail values are captured.
(3) *The MBPTA-CV method*. Building on (1) and (2), we propose MBPTA-CV, a new automatic mixture-distribution aware, WCET-suited MBPTA method for WCET estimation. MBPTA-CV builds upon the understanding of when EVT parameters can be regarded as valid for timing analysis, i.e. they fit the boundary conditions of the timing analysis problem. MBPTA-CV also specifically fits only values from the distribution of the tail in the data sample. To that end, MBPTA-CV builds on a recent tail-classification method for EVT projections – the CV-plot [Del Castillo et al. 2014]. Overall, MBPTA-CV complies with all constraints related to the timing-analysis problem and applies EVT appropriately to deliver high-quality pWCET estimates.

We compare *MBPTA-CV* against an existing MBPTA approach, *MBPTA-orig* [Cucu-Grosjean et al. 2012; Cucu et al. 2013] on both a hardware-randomised simulated platform and a real FPGA COTS platform deploying software randomisation [Kosmidis et al. 2013]. In both cases we target a pipelined single-core architecture comprising a multi-level cache hierarchy. Our results show that MBPTA-CV improves the quality of the pWCET estimates of *MBPTA-orig* while requiring shorter computation time. In particular, MBPTA-CV requests larger samples whenever not enough tail observations are collected with the default sample size and is able to deliver reliable pWCET estimates for a wider range of tail shapes than *MBPTA-orig*. Interestingly, for the Automotive EEMBC benchmarks, the number of runs to carry out is, on average, lower for MBPTA-CV than for MBPTA-orig.

The rest of this paper is structured as follows: Section 2 provides background on MBPTA and EVT. Section 3 introduces our first contribution: an *execution time distribution analysis* in the context of WCET estimation and the implications on EVT. Section 4 presents our proposed *tailoring of EVT to WCET estimation* considering the characteristics of the execution time distributions passed to EVT as input, and the boundary conditions of WCET estimation. Section 5 presents MBPTA-

CV an implementation of our method that is evaluated in Section 6. Related work is reviewed in Section 7 and conclusions provided in Section 8.

2. BACKGROUND AND PROBLEM STATEMENT

2.1. MBPTA, EVT and Representativeness

EVT is a branch of statistics used to predict the probability of events more extreme than those that can be usually observed in a sample. EVT uses as input a data sample (i.e. a set of measurements), and produces as output a curve describing the tail (either left or right) of the distribution to which the input sample belongs. EVT is agnostic to the particular variable being measured and how it is measured (as long as some statistical properties hold for the sample). Therefore, when applied to execution time measurements, EVT makes no assumption on the conditions of the system (a computing platform in our case) from which the measurements are collected. In this respect, EVT has to be understood as a technique to predict the probabilities of the combined impact of timing events observed in the analysis-time measurements, i.e. the sample. In general, EVT cannot predict the appearance of unobserved events since their impact on execution time can be arbitrarily large [Abella et al. 2014b]. In order to ensure that those events affecting execution time during operation are triggered during analysis tests, MBPTA builds a representativeness argument by imposing some requirements on the timing behaviour of the platform. Platforms fulfilling those requirements are called MBPTA-compliant platforms [Kosmidis et al. 2014]. Hence, unlike EVT, MBPTA considers some system internals, so following a grey-box approach. In particular, MBPTA requires identifying those platform components (both hardware and software) with jittery timing behaviour that can affect the execution time measurements [Cazorla et al. 2013b]. Those sources of jitter (*SoJ*), are conveniently controlled *so that their impact on the measurements collected during the analysis phase either matches or upper bounds their impact during operation* [Cazorla et al. 2013b].

SoJ are upper-bounded at analysis time either deterministically or probabilistically [Kosmidis et al. 2014]. Detterministic upper-bounding is performed by enforcing that, at analysis time, the *SoJ* experience their highest latency. In this line, hardware support has been proposed to force jittery resources, i.e. on which operations take different latencies depending on the input values (such as floating-point units), to work on their worst latency [Paolieri et al. 2009b]. Probabilistic upper-bounding is typically enforced by deploying time-randomised resources. For instance, time-randomised caches [Kosmidis et al. 2013; Kosmidis et al. 2013; Anwar et al. 2015] make hits and misses have a probabilistic behaviour, i.e. accesses have a probability of hit (miss) in the cache.

MBPTA-compliance properties can be achieved with hardware or software means. At hardware level, for deterministic upper-bounding, the proposal in [Paolieri et al. 2009b] forces jittery resources to work on their worst latency. For probabilistic upper-bounding, also at hardware level, buses arbitrate requests using a random policy [Jalle et al. 2014] rather than a deterministic policy like round-robin. Another example of the latter corresponds to random placement and replacement policies [Kosmidis et al. 2013; Anwar et al. 2015] used to master cache jitter. With the same goal, software randomisation solutions [Kosmidis et al. 2013] have been proposed to master the cache jitter – randomly allocating program objects in memory – for COTS cache designs (e.g. deploying modulo placement and LRU replacement).

The benefits of MBPTA compliance include reducing the pressure on the end user to control the *SoJ* affecting program's execution time. In particular the user does not have to architect experiments so that the worst-case behaviour of multiple *SoJ* are triggered in the same experiment.

— For instance, let us assume a single-path program[1] comprising floating-point operations that have a variable latency depending on the particular operands (which is commonly the case in many real processors). By forcing floating-point operations to work at their worst latency, MBPTA frees the user from controlling the values operated on in the analysis tests.

[1] How to apply MBPTA in the context of multiple paths is detailed in Section 7.

— It is also well-known that memory mapping, i.e. the placement of objects (code, data) in memory, determines how objects are mapped to different cache sets. This occurs because the objects' addresses determine the set they are mapped to, ultimately impacting which accesses will hit/miss in cache[2]. Cache randomisation makes *cache placement random across runs* so that in every new run, objects are randomly assigned different sets in cache. As a result, the user is relieved from controlling memory mapping in such a way that the potential bad cache layouts that can appear during operation are captured at analysis. As applications become more complex and incremental software integration more ubiquitous [Mezzetti and Vardanega 2013] exercising such control becomes increasingly hard, so relieving the end user from this burden becomes critical for keeping high confidence in the timing analysis process.

2.2. EVT Methods: Block Maxima and Peak Over Threshold

Accurately approximating the tail of a distribution requires EVT to select only those observations that truly correspond to the tail, out of the observations in the sample. Two main methods exist for that purpose: Block Maxima (BM) and Peak Over Threshold (PoT) [Feller 1996]. This section introduces BM and PoT and their parameters. We subsequently relate those parameters to the particular characteristics of the sample data used, which in our case are the execution time observations taken at analysis time.

Block Maxima. BM takes as an input parameter the block size (bs), which is used to split the sample into smaller equally-sized blocks. For instance, given a sample of $R = 1,000$ observations and a block size of $bs = 20$ elements, BM splits the data into $nb = 50$ blocks with 20 observations each. Given that observations in the sample are independent[3], any arbitrary way to perform such distribution among blocks is valid. Typically blocks are created with consecutive elements in the sample: $[1, bs]$, $[bs{+}1, 2{\cdot}bs]$, $[2{\cdot}bs{+}1, 3{\cdot}bs]$, and so on and so forth. The highest value of each block is selected to create a small sample (of maxima values) with as many observations as blocks (so $nb = \lceil R/bs \rceil$ values). Those nb values are then used to fit the appropriate Generalised Extreme Value (GEV) distribution. In the context of BM there are three continuous probability distribution families: Weibull, Gumbel and Fréchet. They are jointly described by the GEV parametric distribution family. The Cumulative Distribution Function (CDF) of GEV is defined as [Coles 2001]:

$$G(x;\mu,\sigma,\xi) = exp\left[-\left(1+\xi\frac{x-\mu}{\sigma}\right)^{-1/\xi}\right] \tag{1}$$

where the parameters μ, σ and ξ are known as the *location*, *scale* and *shape* respectively. Note that in Equation 1 it holds that $1+\xi\left(\frac{x-\mu}{\sigma}\right) > 0$.

The shape parameter, ξ, determines whether the distribution corresponds to a Weibull ($\xi \leq 0$), Gumbel ($\xi = 0$) or Fréchet ($\xi \geq 0$) distribution, which are also known as light, exponential and heavy tail distribution respectively. The importance of this parameter is justified by its impact on the tail of the distribution, which is the relevant part for MBPTA.

The typical shape of the tail for each GEV distribution is illustrated in Figure 1 with the Complementary CDF, (CCDF or exceedance distribution), in logarithmic scale of a synthetic example. As shown, the Weibull distribution (light tail) with $\xi = -0.5$ has a sharp slope and a maximum value (200 in the example). The Gumbel distribution (exponential tail) with $\xi = 0$ has also a relatively sharp slope but it has no maximum. Finally, the exceedance probability for the Fréchet distribution (heavy tail) with $\xi = 0.5$ decreases only polynomially.

Peak-over-Threshold. PoT relies on a threshold (th) value to keep only those values in the sample higher than th. This way a smaller sample corresponding to the observations with information

[2]In COTS architectures the mapping between the addresses accessed and the cache set they are mapped to is determined by the placement function – e.g. modulo.

[3]Independence across observations must hold, in general, to apply EVT. Under some circumstances independence is not strictly needed as discussed in Section 7.

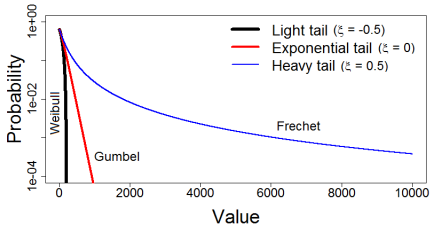


Fig. 1. Example of CCDF for light, exponential and heavy tail GEV distributions with $\xi = -0.5$, $\xi = 0$ and $\xi = 0.5$ respectively. ($\mu = 0$ and $\sigma = 100$).

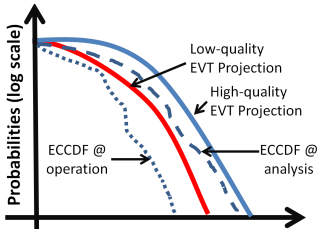


Fig. 2. ECCDF at analysis and during operation, and two EVT projections.

about the tail distribution is kept. We use N_{+th} to refer to the number of values (exceedance values) not rejected from the sample, i.e. those values higher than th. For instance, $N_{+th} = 100$ means that only the 100 highest values are kept. The probability distributions obtained for the tail are described by the Generalised Pareto Distribution (GPD) family, whose CDF is as follows:

$$H(x; \mu, \sigma, \xi) = 1 - \left[1 + \xi \left(\frac{x - \mu}{\sigma}\right)\right]^{-1/\xi} \tag{2}$$

where the parameters μ, σ and ξ have the same interpretation as for GEV (in fact ξ is identical), and $x > \mu$. GPD can also be described using a two-parameter form instead of a three-parameter form, but this is irrelevant for our discussion here. GPD delivers exactly the same distributions as GEV for the same ξ values, so light, exponential and heavy tail distributions[4] as shown in Figure 1, with similar values for μ and σ. We refer the interested reader to [Coles 2001] for details on how μ and σ relate between GEV and GPD.

2.3. Quality of the WCET estimates and evidence for certification

The timing part of the verification and validation process focuses on covering time-related hazards in accordance with the safety standard(s) in place. To that end, it is required to assess the quality of the provided WCET estimates, or in other words, the confidence that can be had on their correctness.

In this process formal proofs are always welcome, but in general they are hard – if at all possible – to build upon real processors: while typically formal proofs cover the soundness of the timing analysis method, they provide no assessment on the method's inputs. And it is clear that a tool provided with inaccurate or incomplete timing data may produce low-quality WCET estimates. For instance, in the context of increasingly complex processors, formal proofs cannot be provided on the correctness of the timing (cost) models for static timing analysis. Instead, industry resorts to measurements to derive those figures. As an illustrative example, Airbus and AbsInt had to resort to measurements to obtain some access latencies for the Freescale P4080 processor [Nowotsch et al. 2014]. These, as well as other aspects related to the reliability of the WCET estimates, are further elaborated in [Paulitsch et al. 2015].

In practice, measurement-based timing analysis techniques are the most common approach in use in industry [Wilhelm et al. 2008]. An engineering margin is added to the highest execution time captured during the analysis, referred to as high watermark. The engineering margin is based on an expert's experience and knowledge about the underlying hardware/software platform. This unscientific approach makes it difficult to assess the quality of the resulting WCET estimates.

For MBPTA, which aims to decrease the user's burden to provide guarantees on the quality of the WCET estimates, evidence for certification builds on two elements:

[4]The particular methods used to obtain μ, σ and ξ given a sample (i.e. set of observations) are omitted in this discussion. Details can be found in [Coles 2001].

— Provide evidence that the *SoJ* are upper-bounded, either deterministically or probabilistically, so that their impact is higher at analysis time than during operation. If this is achieved, observations at analysis can be used to upper-bound execution time during operation. This is better illustrated in Figure 2 in which the dotted line represents the empirical CCDF (ECCDF) of the execution times of the program during operation and the dashed line corresponds to the ECCDF of the execution times observed at analysis. The former cannot be obtained in the general case during the analysis phase [Cazorla et al. 2013b].
— Provide evidence that the EVT projection tightly fits the sample it is provided as input. That is precisely the goal of correctly tailoring EVT for WCET estimation so that high-quality EVT projections upper-bounding the analysis time distribution are obtained (see the blue solid line). Low-quality EVT projections are those that may not upper-bound the analysis time distribution (see the red solid line). Depending on how much the analysis time distribution upper-bounds the operation-time distribution, low-quality EVT projections may also fail to upper-bound the operation time distribution.

In this paper, we provide a thorough analysis of the conditions/constraints affecting the execution time behaviour of real-time programs. From this analysis we further elaborate on the appropriate parameter selection for GEV/GPD. This analysis helps gathering evidence for (and is fundamental to) building appropriate arguments on the quality of the provided pWCET estimates. Then, we provide an implementation of the method called MBPTA-CV and provide evidence on how it provides high-quality pWCET projections that upper-bound the analysis time distribution and hence, the operation-time one also.

2.4. Problem statement

Given a sample of an execution time distribution of a given real-time program on a MBPTA-compliant platform, our goal is to propose a method that (1) selects only tail values from that sample and (2) selects GPD parameters to tightly fit the sample values. This is done under a set of constraints that apply to real-time programs, e.g. finite execution time. Besides a correct EVT projection, the quality of the obtained pWCET estimate relies on the sample capturing execution times being representative of the operation conditions. With respect to the latter, we verified that all execution time samples contain at least one observation of all relevant timing events, so representativeness is always achieved. In particular, as indicated in [Abella et al. 2014b; Benedicte et al. 2016; Milutinovic et al. 2016], these events correspond to cache placements that can occur with non-negligible probability (e.g. above 10^{-15} per run for the highest integrity levels) and where addresses producing a highest miss increase when placed together are, effectively, placed in the same set. We used the technique in [Milutinovic et al. 2016] to identify those placements and verified that they occurred at least once in the execution time measurements collected.

3. CAUSES AND IMPLICATIONS OF WELL-BEHAVED AND MIXTURE DISTRIBUTIONS

In this section we analyse whether hardware resources may produce mixture execution time distributions (see Table I). This type of distribution is the focus of this paper as it poses difficulties in the application of EVT to accurately approximate its tail. In particular, selecting only those observations that truly correspond to the tail of the distribution, out of all those observations in the sample, becomes much more challenging.

3.1. Taxonomy

Jitterless resources such as integer adders and those enforced to work on their worst latency (e.g. floating-point unit) have a fixed impact and shift to the right the execution time distribution.

Time-randomised resources may lead to either well-behaved or mixture distributions. This depends on several factors. On the one hand, whether the jitter caused by the resource is low or high in relation to other SoJ; on the other hand, whether the resource is used frequently: the higher the number of requests the lower the probability of not capturing the distribution of their response times.

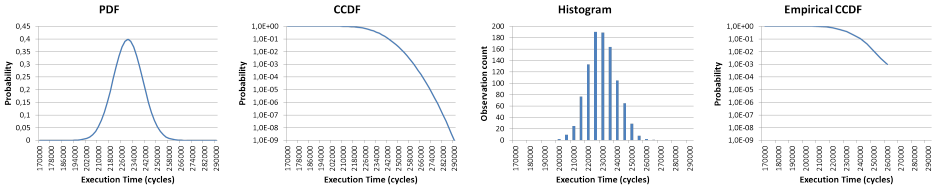


Fig. 3. Example of well-behaved distribution: PDF, CCDF, histogram and ECCDF.

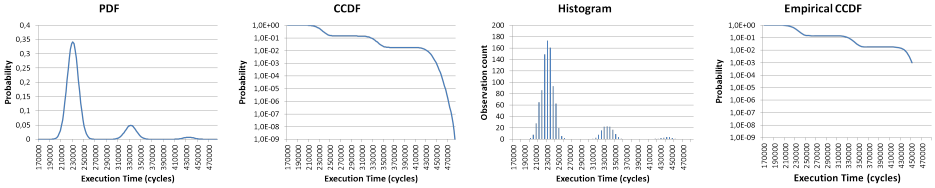


Fig. 4. Example of mixture distribution: PDF, CCDF, histogram and ECCDF.

(1) Recent work has shown that random arbitration in resources such as a shared bus or memory controller [Jalle et al. 2014], create increasing and smooth variability for decreasing probabilities in the execution time distribution [Abella et al. 2014b]. This occurs because each individual event has limited impact on execution time and they occur frequently during program's execution. This is, for instance, the case of bus arbitration that takes few cycles. Furthermore, events such as shared resource arbitration are highly independent – though not completely. That is, with randomised bus arbitration, the time required to serve requests is largely insensitive to other events occurring in the pipeline or in the caches. As a result, these resources lead to *well-behaved* execution time distributions like the one in Figure 3 that shows, from left to right, the PDF, the CCDF in logarithmic scale, the histogram for a sample with 1,000 observations and the ECCDF for that sample. The PDF resembles quite well the shape of a Gaussian distribution due to the high degree of independence across random events and their relatively high number.
(2) Cache replacements under random cache replacement are mostly independent events with limited individual impact in execution time, and they occur often during a program execution.
(3) Random placement in caches – implemented with either hardware [Kosmidis et al. 2013; Anwar et al. 2015] or software [Kosmidis et al. 2013] means – may produce *mixture* (step-like) execution time distributions like the one shown in Figure 4. Those steps occur because some particular addresses randomly compete for the same cache set. If they are mapped to the same set and exceed cache set space, a large number of misses may be experienced. Conversely, if they are randomly mapped to different sets, few conflict misses occur. As a result, the execution time variation across those different scenarios is enough to create large steps in the CCDF. For instance, the example in Figure 4 shows a case where most placements create few conflicts (leftmost peak in the PDF), some other placements create a large number of conflicts (middle peak in the PDF) and few placements create an even larger number of conflicts (rightmost peak in the PDF). The variability around each peak corresponds to other random effects such as shared resource arbitration. In practice, the number of peaks and their probabilities can be arbitrary. Further, unlike random replacement or random bus arbitration, which occur very frequently during the execution of the program, placement is randomised only once, right before executing the program – followed by a cache flush for cache consistency.

Corollary: Overall, random placement can produce mixture distributions, while random replacement and arbitration, jitterless resources, and those made to work in the worst latency cannot. The latter two shift the distribution to the right, but their impact is captured in analysis-time measure-

Table I. Taxonomy of resources

Resource	Impact
Fixed latency resources (e.g. ALU) Resources forced to work on their worst latency (e.g. some FPU operations)	Shift distribution to the right
Random bus arbitration for on-chip buses Random memory-controller arbitration Random replacement	Lead to well-behaved distributions
Random placement	May lead to mixture distributions

ments. Meanwhile, cache replacement and random arbitration result in well-behaved distributions, and require few runs to ensure that random effects are properly captured [Abella et al. 2014b; Benedicte et al. 2016; Milutinovic et al. 2016]. Hence, only the inclusion of random cache placement requires taking extra care in the application of EVT to derive high-quality WCET estimates.

3.2. Mixture Distributions, representativeness and EVT application

The impact of mixture distributions on representativeness has been well studied in the literature. For MBPTA-compliant platforms, research has been conducted on how to provide evidence that those *SoJ* creating steps in the execution time distribution are captured in the execution time sample fed to EVT as part of MBPTA [Abella et al. 2014b; Benedicte et al. 2016; Milutinovic et al. 2016]. Those works focus either on the memory objects accessed for both hardware time-randomised caches [Abella et al. 2014b] and software time-randomised caches [Benedicte et al. 2016], or on cache access patterns for hardware time-randomised caches [Milutinovic et al. 2016]. Enforcing representativeness, in turn, ensures that EVT is provided with a sample that captures all the peaks in the execution time distribution. For instance, coming back to the example in Figure 4, the representativeness step of MBPTA guarantees that EVT is provided with a sample with a similar histogram to the one in the figure, which contains measurements for all peaks in the PDF. In this work we used those methods to set up configurations where representativeness holds for all benchmarks with the minimum number of runs used in any experiment.

The impact of mixture distributions on EVT application is actually the matter of this paper. Ideally, we would want EVT to provide distributions closely upper-bounding those shown in the CCDF in Figures 3 and 4. The difficult of achieving show depends on the shape of the distribution.

— *Remark 1:* For well-behaved distributions, all values above the mode already provide information about the tail, so virtually any bs for BM and th for PoT capture the shape of the tail.
— *Remark 2:* For mixture distributions, measurements corresponding only to the rightmost peak of the PDF are truly tail measurements. EVT can only deliver high-quality pWCET estimates when it uses sufficient values from the rightmost peak, and only values from the rightmost peak. This depends on bs for BM or th for PoT and requires understanding how pWCET distributions are affected when i) non-tail measurements are considered and/or ii) few tail measurements are used.

In the following section we show the impact of nb and th on BM and PoT respectively. This reveals that selecting those parameters is a complex task and highly depends on the characteristics of the observations in the sample. Thus, this challenges the automation of the parameter selection.

4. CUSTOMISING EVT TO THE TIMING ANALYSIS PROBLEM (MBPTA)

EVT is used in domains ranging from hydrology to finance. In each domain, there are certain features that significantly influence how EVT ought to be better applied. These include the range of values of the variable under consideration, the number of experiments that can be performed – limiting the size of the sample – and the conditions under which experiments are performed. In order to tailor EVT for WCET estimation we identify four constraints or observations (O1-O4):

— (O1) *Upper-bounded value range*. The highest execution time of a program is finite, i.e. it has a maximum. This occurs because, in the critical real-time domain – the target of our work – some code constraints exist. For instance, loops have a known maximum of iterations and infinite loops

are forbidden. In the target domain, programs with real-time constraints have a maximum execution time, which emanates from the fact that they can execute a finite number of instructions with fixed maximum latencies. The maximum execution time, while it exists, is typically unknown, making the estimation of the WCET a challenge.

— (O2) *Discrete nature*. Due to the discrete nature of execution times, execution time distributions are discrete too. Moreover, some execution time values within the range of the minimum and maximum execution times of a program may not be feasible due to the specific latencies of the hardware resources, their complex relation and the specific use made by the program under analysis. Therefore, the sizes of the steps between potential execution times can be arbitrarily large. This, in fact, matches with the analysis in Section 3, where we show that mixture distributions, i.e. distributions with large steps, may be obtained from execution times.

— (O3) *Sample size*. The timing validation process is a costly part of system (or subsystem) validation. This limits the time the user can devote to timing validation, which, therefore, limits the number of program runs that can be performed (i.e. the sample size). While the affordable number of runs is user and application dependent, sample sizes in the order of hundreds to few thousands of measurements have been regarded as affordable, for example, in the avionics industrial domain [Wartel et al. 2015].

— (O4) *Pessimism over Optimism*. The goal in using EVT in critical real-time applications is to derive upper-bounds to program execution time during operation. It stands to reason that choices in the application of EVT must trade pessimism for accuracy whenever higher accuracy increases the likelihood of being optimistic.

4.1. Customising the Type of the tail: ξ

In order to obtain low pWCET values, we are interested in choosing the tail with the sharpest slope, i.e. with the lowest ξ value, as shown in Figure 1, as long as it can be proven to be an upper-bound to the execution time distribution during operation. Based on *(O1)*, we can discard heavy tails since execution time distributions in our target domain are upper-bounded. However, since the maximum value is generally unknown, we cannot discriminate between exponential and light tails. Therefore, in line with *(O4)*, we must use exponential distributions to model the tail of the pWCET curve, since it is an upper-bound for all potential tails of the execution time distribution being analysed. This implies that $\xi = 0$. Some works already show that execution times can be modelled with exponential tails (Gumbel distributions with GEV) [Cucu-Grosjean et al. 2012; Hansen et al. 2009; Edgar and Burns 2001]. In this direction, it is worth noting that in the case of MBPTA-orig, the bs leading to a ξ value closer to 0 (so exponential tail) is selected.

Other authors have reached a different conclusion and argue that execution times may need to be modelled with heavy tails [Lima et al. 2016]. This is observed when varying some input values of programs whose loop bounds (or number of recurrent calls) directly depend on those input values. While heavy tails may be appropriate in some domains, it is not the case in the critical real-time domain where timing analysis is conducted with the help of flow-facts that provide bounds on loops (or recurrent calls). Otherwise, if those bounds did not exist, programs could take unbounded execution time by construction, which clashes with the nature of critical real-time programs (O1). Moreover, if different sets of input values are analysed, the pWCET for each set needs to be handled separately by MBPTA since, otherwise, the identical distribution premise would not hold despite the fact statistical tests may be passed[5]. How to combine pWCET estimates for different paths and how to achieve full path coverage in the context of MBPTA is discussed elsewhere [Ziccardi et al. 2015].

4.2. Fitting the Tail Distribution to the Input Data: σ and μ

When fitting the best GPD or GEV distribution to the data sample, different approaches can be followed. For instance, *MBPTA-orig* fits the best GPD or GEV distribution whose ξ value is sufficiently

[5]Statistical tests provide information about the data sample at hand rather than about the distribution being observed. Therefore, those tests may be passed either by chance or by introducing some form of randomisation in the data collection process.

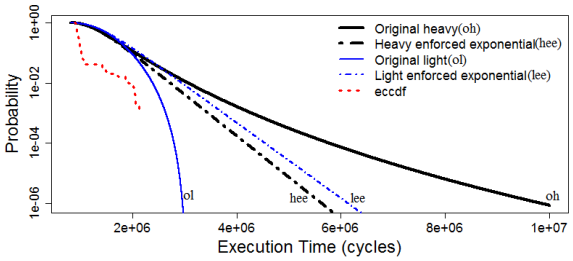


Fig. 5. Enforcing exponential distributions for `cacheb`. The dotted red line shows the (empirical) CDF of the observed values to asses the quality of the derived EVT projection. The empirical CDF is also shown in Figure 6 and Figure 7.

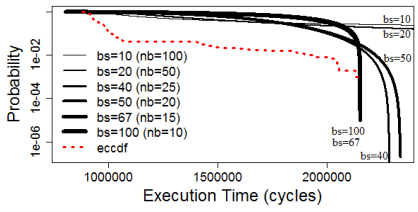
close to 0 so that the exponentiality hypothesis cannot be rejected statistically (the Exponential Test is used for that purpose [Garrido and Diebolt 2000]). Therefore, it obtains the three EVT parameters (ξ, σ and μ) where $\xi \approx 0$ but it is exactly 0 only occasionally. *MBPTA-orig* enforces exponential tails by enforcing $\xi = 0$, so increasing or decreasing ξ, but it preserves the values obtained for σ and μ. We refer to this approach as *Fit&Enforce* (or F&E for short). F&E (1) fits the best GPD or GEV distribution to the data once exponentiality cannot be rejected, thus obtaining $< \xi_{unc}, \sigma_{unc}, \mu_{unc} >$ (unc refers to *unconstrained*). Note that determining the distributions best fitting the data can be done with methods available in most statistical packages (e.g. R). Then, (2) replaces ξ_{unc} by 0 providing as output $< 0, \sigma_{unc}, \mu_{unc} >$. However, this has some implications, since a relation (dependence) exists among all three EVT parameters.

Our proposal for fitting an exponential tail to the data sample – which is the one we use later as part of MBPTA-CV – consists of enforcing $\xi = 0$ and then fitting the other parameters, σ and μ, once an exponential tail cannot be rejected as hypothesis. We refer to this approach as *Enforce&Fit* (or E&F for short). E&F, therefore, only requires determining the exponential distribution that best fit the data (tail values) once exponentiality has been tested, instead of determining the GPD or GEV distribution that best fit tail values. Overall, E&F (1) imposes $\xi = 0$, and then, (2) fits the best exponential distribution to the data, thus obtaining $< 0, \sigma_{\xi=0}, \mu_{\xi=0} >$. The differences between the two approaches (F&E and E&F) are better illustrated with the following example.

Synthetic example. We focus on one EEMBC AutoBench benchmark [Poovey 2007] – `cacheb` – and collect $R = 1,000$ execution time observations on a cycle-accurate simulator modelling a MBPTA-compliant processor architecture. Further details on the evaluation framework are provided later in Section 6. We study the effect of enforcing $\xi = 0$ on the assumption that the tail of the execution time distribution is exponential (or can be upper-bounded with an exponential distribution). With F&E, after enforcing $\xi = 0$, the values of the standard deviation (sd) and the average ($\overline{x}$) of the sample are kept as they were when ξ was not zero. This may lead to relatively low-quality approximations if ξ is different enough from 0 since the fitting of the distribution has been made for all three parameters. Figure 5 shows the impact F&E has for the `cacheb` benchmark when applying BM with $bs = 22$ and $bs = 28$ respectively. In both cases the Exponential Test [Garrido and Diebolt 2000] is passed.

— For $bs = 22$, ξ is 0.089, so slightly in the heavy tail region (*original heavy* in the plot). If $\xi = 0$ is enforced without fitting the other parameters again, the slope of the distribution sharpens as shown in the plot (hee). This could negatively affect the quality of the pWCET curve since there is no control on how much the distribution sharpens (see oh and hee in Figure 5).
— For $bs = 28$, ξ is -0.201, so in the light tail region (*original light* in the plot). Enforcing $\xi = 0$ without fitting the remaining parameters smoothens the slope of the distribution, potentially jeopardising the tightness of EVT results (see ol and lee in Figure 5).

E&F improves the accuracy of the exponential distribution obtained compared to F&E since it does not mix parameters for different tail fittings. Instead, E&F fits the appropriate distribution to the data. The quality of tails obtained with F&E decreases when the variability in the highest values of



(a) EVT projections and eecdf

nb	100	50	25	20	15	10
bs	10	20	40	50	67	100
ξ	1.672	0.744	-0.470	-0.439	-1.000	-1.090

(b) ξ values for `cacheb` with different bs parameters

Fig. 6. Effect of having data not from the tail: the case of `cacheb` with GEV (BM) and $R = 1,000$.

the sample (normalised w.r.t. the median) is high. In this particular example, the *sd* in the highest 41 values is 259441 cycles, whereas the *sd* of the complete sample is 159564 cycles. Thus, variability in the tail increases by more than 60% in absolute terms.

4.3. Selecting Only Tail Values for Tail Fitting

Both BM and PoT aim at filtering out values not belonging to the tail. However, for a given nb for BM or N_{+th} for PoT, the observations considered as part of the tail may actually include non-tail observations. Whether or not this happens depends on the particular sample under consideration.

For instance, for a sample size of 1,000 values, the `cacheb` benchmark sample has 959 observations in the range [879000, 1125000] and the remaining 41 in the range [1402000, 2152000]. Therefore, by observation one can conclude that at most 41 values belong to the tail. For this example, Figure 6(a) shows the effect of using different bs and nb values for BM (similar results are obtained using different N_{+th} values for PoT, which we omit in the plot for the sake of clarity). The value of ξ for all scenarios is shown in Figure 6(b).

When bs is low, e.g. $bs = 10$ or $bs = 20$, BM uses too many values ($nb = 100$ and 50 respectively). This may cause some of the resulting nb observations to not actually belong to the tail. For instance, if $bs = 20$ at most 41 of the $nb = 50$ values used to approximate the tail truly belong to the tail[6]. The effect of selecting non-tail values is that $\xi \gg 0$, thus in the domain of heavy tails and regarded as not appropriate in our context, as explained in Section 4.1. The reason is that *fitting a function to a sample with two groups of observations (tail values and non-tail values) leads to a polynomial (heavy-tailed) function whose slope is smooth enough to remain close to both groups*. This is, for instance, illustrated in Figure 6(a) and Figure 6(b) in which we see the slow decreasing rate of the exceedance probability for those cases where $nb = 50$ ($bs = 20$) and $nb = 100$ ($bs = 10$). Conversely, when bs is large enough, e.g. 40, fewer observations are kept (e.g. $nb \leq 25$ at most) and it is less likely that any block lacks tail values. Thus, either $\xi \approx 0$ or $\xi \leq 0$. Overall, in theory a larger block size reduces the probability of taking non-tail values. However, this also leads to the problem of using few observations to fit the EVT distribution, which decreases the confidence in the obtained results. Analogous observations can be made for PoT where keeping too many values (e.g. $N_{+th} = 100$) forces GPD to use many non-tail values, thus leading to a heavy tail distribution.

Our solution to this problem is explained in Section 5.2 as part of the MBPTA-CV process. The basic idea is that MBPTA-CV ensures that the set of tail values used as input to EVT can be modelled with an exponential tail. When, for a given set of values, the best suited distribution is a heavy-tail distribution, then the user is asked for more runs until exponentiality is achieved.

[6]While a similar problem occurs for PoT, BM presents a problem w.r.t. PoT: whenever nb is smaller but close to the number of values in the sample that truly belong to the tail, it is likely that i) some blocks do not include any tail value and so values from the bulk of the distribution are selected and ii) some blocks contain multiple tail values so that all but one are rejected. Hence, unless nb is significantly smaller than the number of tail values, chances are that i) and/or ii) occur. Overall, BM may be discarding some tail values and including some non-tail values, which will undermine the accuracy of the distribution used.

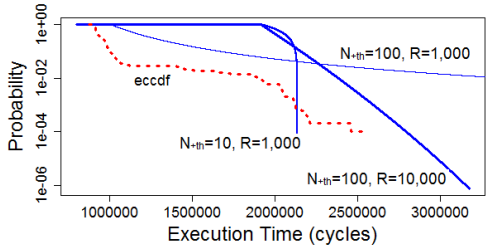


Fig. 7. Effect of missing enough tail values and GPD (PoT) for `cacheb`.

4.4. Selecting Sample Size to Have Enough Tail Values

In general there is not a clear answer on how to choose the minimum size of a sample to fit an exponential distribution. Different works [Cai and Hames 2011; Clarke 2002] – in fields where the use of EVT is well established – point to values between 10 and 50 observations for Gumbel distributions, which have the same nature as exponential ones. In the context of MBPTA, where representativeness is achieved by deploying one of the existing proposals presented in Section 3, the starting sample contains at least one value of the tail. Statistically, the larger the sample, the larger the number of tail values in the sample. Therefore, appropriate means are needed to determine whether the sample size needs to be increased or sufficient tail values have been collected for an accurate tail fitting. This has to be done in the context of (O3) as presented in Section 4.

For the example of `cacheb` we only have 41 tail observations, which may not be enough for fitting an exponential distribution if no less than 50 tail values are wanted. Increasing N_{+th} for PoT (or reducing bs to increase nb for BM) may lead to low-quality WCET estimates. For instance, for $N_{+th} = 100$, many non-tail values are used, so $\xi = 0.416$ indicating that the tail distribution obtained is a heavy tail (as shown in Figure 7), which is unusable in practice. If instead of increasing N_{+th} for $R = 1,000$, we use a larger sample, e.g. $R' = 10,000$, and keep $N_{+th} = 100$, then all values belong to the tail and obtain $\xi = -0.028$, which is very close to the exponential distribution (i.e. the exponentiality hypothesis cannot be rejected).

Figure 7 also shows that if $N_{+th} = 10$ then $\xi < 0$, and we obtain a light tail distribution. Hence, assuming an exponential tail distribution would be safe. Whether 10 values are enough for an accurate fit depends on the degree of accuracy desired, which relates to the possibility of obtaining further measurements and the implications of low accuracy for the problem at hand. We discuss this matter in detail in Section 5.2, where we present our choices for MBPTA-CV.

4.5. Corollary

From the analysis performed in this section, the following important conclusions are obtained for a correct application of EVT. First, enforcing $\xi = 0$ provides always a reliable upper-bound in the problem at hand. Second, the E&F (enforce and fit) approach must be used to fit the distribution to the data to obtain the best fit. Third, using non-tail values for tail fitting may lead to inaccurate tail distributions. And fourth, enough tail values must be used for accurate tail fitting, even if this implies increasing the sample size.

5. MBPTA-CV

In this section we introduce MBPTA-CV, a new method to obtain pWCET estimates that builds upon three pillars: the execution time distribution analysis in Section 3, the customised application of EVT to the pWCET problem, presented in Section 4, and recent findings in the EVT field such as the CV-plot. We start with background Section 5.1 describing the CV-plot technique to classify tails as heavy, exponential, or light, for different exceedance thresholds under PoT. Then we present the parameter choices of MBPTA-CV (Sections 5.2 and 5.3) and its application steps (Section 5.4).

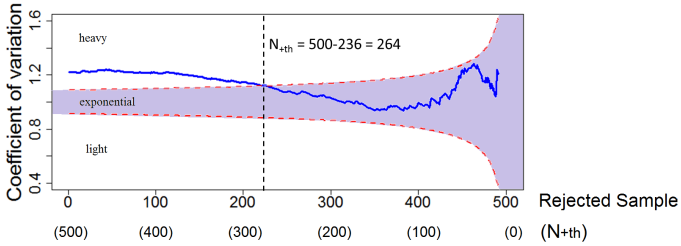


Fig. 8. Example of CV-plot. Dashed lines limit the range in which exponentiality cannot be rejected with a confidence of 0.95. The vertical line shows the point in which there are 236 rejected values, so $N_{+th} = 500 - 236 = 264$.

5.1. Background on CV-plot: a Method for Tail Classification

The coefficient of variation (CV) of a distribution is the ratio between its standard deviation and its mean, which interestingly makes the CV independent of the scale of the distribution. For a given distribution the *residual* CV determines the type of the tail: $CV = 1$ for exponential tails, $CV \geq 1$ for heavy tails and $CV \leq 1$ for light tails [Del Castillo et al. 2014].

5.1.1. Theoretical residual CV. The theoretical residual CV for a given threshold th, $CV(th)$, is defined as follows. Let X be a continuous non-negative random variable with distribution function $F(x)$. Given that $th > 0$, which is the case for execution times, the random variable of the conditional distribution of threshold exceedance values $X - th$ so that $X > th$, denoted $X_{th} = (X - th \,|\, X > th)$, is called the residual distribution of X over th. For instance, given an execution time distribution with values 100, 200, 300 and 400, each having probability 0.25, their residual distribution over 250 would be $300 - 250 = 50$ and $400 - 250 = 150$ each with probability 0.5. For a given th, we compute the mean of the exceedance values, $M(th)$, and their variance, $V(th)$, also known as *residual mean* and *residual variance* respectively. The theoretical residual CV is obtained as follows:

$$CV(th) \equiv CV(X_{th}) = \frac{\sqrt{V(th)}}{M(th)} \tag{3}$$

$CV(th)$ is used in the context of GPD since it considers exactly all exceedance values above a particular threshold th, as GPD does. In fact, $CV(th)$ characterises the distribution and, in the case of GPD, it provides a constant value, where $\xi < 0.5$:

$$CV(th) \equiv CV(X_{th}) = \frac{1}{\sqrt{1 - 2 \cdot \xi}} \tag{4}$$

Since, in the context of GPD, $CV(th)$ depends only on ξ, $CV(th)$ is constant for any th value.

5.1.2. Empirical Residual CV. For a given sample one can obtain the type of the tail for a given data sample and th value estimating $CV(th)$ as $cv(th) = sd/\overline{x}$, where sd and $\overline{x}$ are the standard deviation and mean of the sample for X_{th} respectively. Note that $M(th)$ and $\sqrt{V(th)}$ are the theoretical counterparts of the empirical sd and $\overline{x}$ values obtained for a given sample. Based on the $cv(th)$ estimator, one can iteratively draw the *CV-plot* that comprises a $cv(th)$ value for each th [Del Castillo et al. 2014]. The application of the CV-plot in the scope of EVT has not occurred until very recently. As shown later in the evaluation section, the CV-plot is a very powerful tool when used in the context of MBPTA since (1) it provides a fast way to classify tails (i.e. computing $cv(th)$ instead of having to fit an exponential distribution to the data) and (2) allows analysing jointly the $cv(th)$ for different th values. Note, however, that how to use this information is application-dependent and, in our case, a key contribution of MBPTA-CV.

The CV-plot is illustrated in the example in Figure 8. The dashed lines determine the range in which exponentiality cannot be rejected with a confidence of 0.95. We observe how the range narrows as more values are considered (i.e. fewer are rejected) and vice-versa.

— If X is in the *domain of attraction* of a Gumbel distribution [Kotz and Nadarajah 2000] (i.e. within the dashed lines), then it has an exponential tail, i.e. $cv(th)$ is close enough to 1.
— If the distribution has a heavy tail for a particular number of observations, then its $cv(th)$ estimator is above both dashed lines.
— If it has a light tail, then $cv(th)$ is below both dashed lines. In this case we can assume exponentiality for pWCET estimation purposes since, as explained before, assuming a heavier tail than in reality (i.e. assuming an exponential tail instead of a light one) is a conservative choice.

The blue solid line shows the CV-plot for a particular data sample of $R = 1,000$ observations. First, based on exploratory data analysis all observations that do not correspond to the right tail of the distribution are rejected. This would imply rejecting observations lower than the value with highest probability in the histogram (highest peak in the PDF in Figures 3 and 4). In practice, in our environment, this is achieved by rejecting the lowest half of the sample[7]. Next, in an iterative manner, th is increased so that the number of exceedance observations (N_{+th}) decreases, and for each such th value the $cv(th)$ estimator is computed, resulting in the blue line. This line corresponds to the actual $cv(th)$ as we discard values out of the remaining 500 (after discarding the lowest 500).

Interestingly, the CV-plot provides information about the type of the tail of the distribution when using PoT as we increase th – and so decrease the size of the sample (set of exceedance values) used to obtain the GPD. For instance, in Figure 8 we observe that the tail is classified as heavy for the range $N_{+th} \in [500, 264)$ of observations, since $cv(th)$ is above both dashed lines. Instead, for $N_{+th} \leq 264$ the exponentiality hypothesis cannot be rejected.

5.2. MBPTA-CV: Parameter Selection

Statistics in general build upon some parameters set based on experience. For instance, many hypothesis tests build upon a significance level that is typically set to 0.05 or 0.01, which determines how biased the test result must be to reject the hypothesis under consideration. In our case we use 0.05 since it allows us reject samples which are not likely to meet some statistical properties (e.g. independence and identical distribution) such as those with test values in the range [0.05, 0].

The application of GPD in the context of EVT is also subject to a critical parameter: the number of tail measurements needed to obtain a *sufficiently* accurate approximation to the real distribution of the tail. As for other parameters in statistics, different answers can be found in the literature, all of them proven as sufficiently good in the respective problems considered. In the case of exponential tail fitting, the number of measurements required for obtaining *sufficiently* accurate approximations ranges between 10 and 50 according to the relevant literature [Cai and Hames 2011; Clarke 2002].

Due to the safety relevance of the problem at hand, we impose that MBPTA-CV uses no less than 50 tail measurements to fit the tail distribution. It can be reasonably argued, with respect to safety standards (e.g. ISO26262 in the automotive domain), that the residual risk of this assumption is negligible, being no more relevant than the residual risk emanating from some non-automated processes such as the enumeration of relevant test cases for software testing and the enumeration of the relevant fault models for a fault type for hardware testing.

Building on the CV-plot, next we present the rules to select the number of tail values (N_{+th}) to feed EVT with, as depicted in Figure 9, and the rationale behind.

[7]We have observed empirically that execution time samples in the context of MBPTA-compliant platforms have the value (or range of values) with highest probability well below the median, so rejecting the lowest half of the sample may, at most, reject some useful observations, but does not keep inappropriate ones. Moreover, if inappropriate observations were kept, $cv(th)$ would be far above 1 and thus, based on the criteria explained later, they would not be used for pWCET estimation purposes since only those exceedance values leading to a smaller $abs(cv(th) - 1)$ are considered.

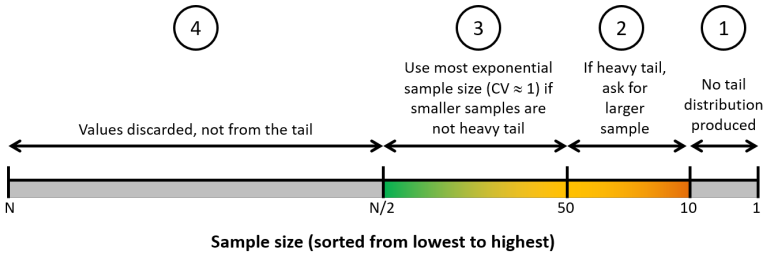


Fig. 9. Sample regions and how tail values are used in MBPTA-CV.

— *Rule 1:* Samples with less than ten values ($N_{+th} < 10$) are disregarded since tail approximations with few values are not used due to their unreliability [Cai and Hames 2011; Clarke 2002].
— *Rule 2:* The $cv(th)$ exponentiality test is performed for all $50 > N_{+th} \geq 10$.
  — If the test fails for any N_{+th} in that range it would mean that some observations for N_{+th} do not belong to the tail. In this situation, we regard the tail approximation as having low-quality since a heavy tail is obtained for N_{+th} in the range [10,50). The end user is instructed to collect more execution time measurements since, eventually, 50 values belonging to the tail – and without abnormally high values[8] – will be obtained.
  — If the test passes (i.e. the tail is classified as exponential or light) for all those N_{+th} values, then a suitable N_{+th} value can be chosen where $N_{+th} \geq 50$.
— *Rule 3:* Once the $cv(th)$ exponentiality test is passed for $50 > N_{+th} \geq 10$, the final N_{+th} value is the one for which it holds the following: (1) $N_{+th} \geq 50$, (2) the $cv(th)$ exponentiality test is passed for any N'_{+th} such that $N_{+th} \geq N'_{+th} \geq 10$, and (3) $cv(th)$ is the closest to 1.
— *Rule 4:* With MBPTA-CV, tail fitting is never carried out with more than half of the number of elements in the sample ($N_{+th} > N/2$). This occurs because the lower half of the sample is assumed not to belong to the tail of the distribution.
— *Rule 5:* Unlike MBPTA-orig, the MBPTA-CV method tests exponentiality for the actual N_{+th} chosen as well as for lower N_{+th}, instead of just testing it for N_{+th}. Further, MBPTA-CV accepts light tails given that they are upper-bounded by exponential tails, rather than requesting more runs as MBPTA-orig does.

5.3. MBPTA-CV: Fitting an Exponential Distribution for the Tail

In this section we propose how to obtain the exponential distribution that best fits the tail modelled. With MBPTA-CV, given a th (and thus N_{+th} exceedance values) so that exponentiality cannot be rejected, the exponential tail distribution fitting best those N_{+th} exceedance values needs to be determined. The CDF of an Exponential distribution is as shown in Equation 5, where only λ needs to be determined[9].

$$F(x, \lambda) = \lambda \cdot exp(-\lambda x) \quad (\forall x \geq 0) \tag{5}$$

With MBPTA-CV λ is obtained as the inverse of $\overline{x}$ for X_{th} ($X_{th} = X - th$ as described before). We subtract th from all N_{+th} ($X_{th} = N_{+th} - th$) values and compute the mean $\overline{x}$ of the resulting values. We obtain λ as $1/\overline{x}$. Once the exponential distribution is obtained with this λ, we add back the value of th to move the exponential distribution to its correct location. This is an important contribution of MBPTA-CV, given that the exponential distribution best fitting the tail modelled can be easily obtained as described above (i.e. applying E&F).

[8] Sporadically, abnormally high values occurring with very low probability may be part of the sample, thus making true tail values be too far to that other value. By increasing the sample size, other values sufficiently close to the abnormally high one will be observed so that an exponential tail will not be rejected.
[9] Note that $\forall x \geq 0$ holds always in our case since execution times are always higher than 0.

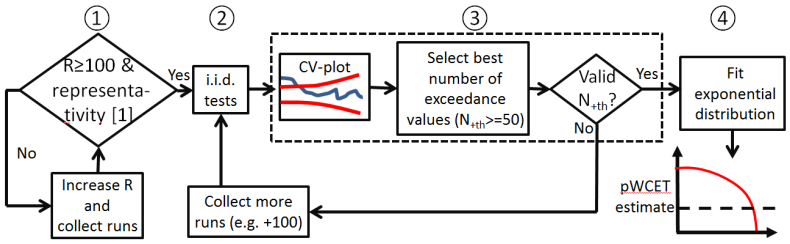


Fig. 10. Steps in the application of MBPTA-CV.

5.4. MBPTA-CV: Application Steps

MBPTA-CV application steps are shown in Figure 10 and described next.

STEP ①. The initial sample size for MBPTA-CV, as well as for MBPTA-orig, needs to be large enough according to the representativeness criteria presented in Section 2. With MBPTA-CV, as explained before, at least 50 observations are needed to approximate the tail distribution. Given that half of the sample is rejected, the sample size must have at least 100 observations. Since high confidence is needed for pWCET estimation, and execution time distributions may have a step-like (mixture) behaviour that may lead to heavy tails for some sample sizes, we impose the following constraints, whose rationale is given in Section 5.2 and formalised here:

— *Const1)* The exceedance threshold selected th must pass the $cv(th)$ exponentiality test, being $N_{+th} \geq 50$ so that the tail distribution may be fitted accurately.
— *Const2)* For a given th (and the corresponding N_{+th}), for all number of exceedances (NE) in the range $N_{+th} > NE \geq 10$, the $cv(th)$ exponentiality test must also be passed.

STEP ②. The next step consists in applying independence and identical distribution (i.i.d.) tests (see Section 6.1.2). Given that the sample is obtained on a MBPTA-compliant platform (with an appropriate measurement collection process), i.i.d. is obtained by construction. However, i.i.d. tests, as any other statistical test, can fail. In that case, the sample needs to be increased until the i.i.d. tests are passed, which is guaranteed to occur eventually since i.i.d. is obtained by construction.

STEP ③. Given a sample of size R we take the upper half and, according to *Const2*, $N_{+th} \geq 50$. Hence, up to $R/2 - 49$ exceedance values are tested. Neither the exceedance value selected (N_{+th}) nor any smaller exceedance NE with $N_{+th} > NE \geq 10$ can be in the heavy tail region. Among those fulfilling these criteria, we select as N_{+th} the value closer to the exponentiality, i.e. the N_{+th} value for which $cv(th)$ is closer to 1. If no N_{+th} value fulfils those criteria, the sample size is increased and the process repeated (passing the i.i.d. tests) until these criteria are met for N_{+th}. Since execution time distributions cannot exhibit heavy tails (see (O1) in Section 4), it is guaranteed that for a sufficiently large sample an N_{+th} value will be found fulfilling the above criteria.

STEP ④. The tail of the distribution is fitted with an exponential distribution (Section 5.3). The value at the target exceedance probability is selected as the pWCET estimate (e.g. 10^{-15} per run).

Overall, MBPTA-CV is aware of the nature of execution time distributions for critical real-time applications. MBPTA-CV narrows down the application of EVT considering the boundary conditions of WCET estimation, and it performs an informed parameter selection and appropriate tail fitting for obtaining high-quality pWCET estimates.

6. EVALUATION

MBPTA-CV can be applied to obtain pWCET estimates of programs running on MBPTA-compliant platforms, regardless of whether such compliance is achieved with customised hardware [Kosmidis et al. 2014] or software-only solutions on COTS hardware [Kosmidis et al. 2013]. MBPTA-CV is also agnostic on how the execution time impact of complex hardware features has been managed (i.e.

by means of upper-bounding or randomisation). A detailed discussion on this matter can be found in [Kosmidis et al. 2014]. While a number of hardware features found in mainstream consumer electronics have not been fully embraced in critical real-time systems, some of them, such as cache hierarchies [Wartel et al. 2015] have already been successfully analysed in the context of MBPTA. The same applies to the contention in the access to multicore resources that needs to be managed analogously as for on-core jitter, i.e. upper-bounding or randomising its timing impact [Jalle et al. 2014; Kosmidis et al. 2014].

In this section we focus on two different setups: a simulated environment where MBPTA-compliance emanates from hardware (Section 6.1), and a real COTS FPGA board in which MBPTA-compliance is obtained with software randomisation (in an external report [BSC 2017]). In both setups we perform several runs of every application in isolation. Yet, the former the platform is configured in a multicore-aware manner in which every request to the shared hardware resources of the EEMBC Autobench under analysis factors in the potential delay it can suffer due to contention with any other tasks. As a result the pWCET estimates upper bound the impact of multicore contention.

We use benchmarks from EEMBC AutoBench [Poovey 2007], a well-known suite for real-time systems mimicking some functions used in automotive embedded systems. Benchmarks in this suite average more than 5,000 lines of code.

6.1. Hardware Randomised Platform

6.1.1. Sample Generation Framework. We make runs in isolation in a cycle-accurate simulator based on SoCLib [SoCLib 2012] that models a 4-core processor architecture resembling that of the Cobham Gaisler NGMP [Cobham Gaisler] – acknowledged as one of the multicore processors currently assessed by the European Space Agency for its future missions. We introduce randomisation support for MBPTA-compliance in multicore processors. In particular, execution time measurements on the hardware randomised platform are collected under the *analysis mode* for shared resource arbitration (i.e. bus and memory controller). This mode imposes that, despite the task under analysis may be running in isolation, all its requests are arbitrated assuming that the other cores always have requests ready to be sent. This way, the contention accounted for the task under analysis upper-bounds any real contention it can suffer during operation – when contenders may or may not have requests ready to compete for shared resources.

Our design comprises local instruction (IL1) and data (DL1) caches in all cores and a partitioned L2 cache. IL1 and DL1 caches are 8KB, 4-way, with 32 bytes per line. The L2 is 128KB, 8-way (2 ways per core), with 32 bytes per line. DL1 is write-through, so each store operation accesses L2 regardless of whether it hits in DL1. All caches implement random placement and replacement [Kosmidis et al. 2013]. The requests generated by the data and instruction caches are handled by means of a shared bus implementing random permutations – one of the most efficient MBPTA-compliant arbitration policies [Jalle et al. 2014]. Analogously, a random permutation policy is also used to arbitrate requests in the memory controller. IL1 and DL1 latencies are 1 cycle, bus latency (once granted access) is 1 cycle and L2 cache latency is 2 cycles, thus leading to L2 total turnaround latencies between 5 and 17 cycles. We assume a CPU frequency of 800MHz, and a JEDEC-compliant [JEDEC 2008] 256Mb x 16 DDR2-800E SDRAM device composed of a single DIMM, single rank and 4 banks. This leads to a CPU-SDRAM clock ratio of 2. The latency of the memory controller is, therefore, 16 processor cycles and inter-access latency is 27 processor cycles since up to 11 cycles are required to close a page once an access has been served, assuming the time-predictable constant-latency memory controller design described in [Paolieri et al. 2009a].

6.1.2. Independence and Identical Distribution. We use the Ljung-Box independence test [Box and Pierce 1970] to test autocorrelation for 20 different lags simultaneously, in line with the proposal in [Abella et al. 2015], where this is shown to be a very strong independence test. We have applied the Ljung-Box independence test and the Kolmogorov-Smirnov two-sample i.d. test [Feller 1996] to all our samples, which have passed both tests for a $\alpha = 0.05$ significance level.

Table II. Normalised quantiles and standard dev. for the execution time of all EEMBC Automotive benchmarks in the HW-randomised platform.

Benchmark	a2time	aifftr	aifirf	aiifft	basefp	bitmnp	cacheb	canrdr
MIN	0.933	0.950	0.965	0.953	0.976	0.961	0.948	0.968
Q1	0.979	0.987	0.990	0.986	0.995	0.987	0.986	0.992
MEDIAN	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Q3	1.027	1.016	1.012	1.019	1.005	1.016	1.049	1.009
MAX	1.166	1.303	1.423	1.226	1.023	1.108	2.319	1.158
sd	0.038	0.037	0.039	0.033	0.007	0.024	0.172	0.016

Benchmark	idctrn	iirflt	matrix	pntrch	puwmod	rspeed	tblook	ttsprk
MIN	0.983	0.967	0.986	0.968	0.973	0.953	0.940	0.952
Q1	0.994	0.992	0.997	0.993	0.994	0.988	0.988	0.988
MEDIAN	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Q3	1.007	1.009	1.004	1.008	1.007	1.013	1.016	1.014
MAX	1.071	1.053	1.242	1.194	1.041	1.121	1.102	1.116
sd	0.011	0.014	0.026	0.018	0.010	0.022	0.024	0.022

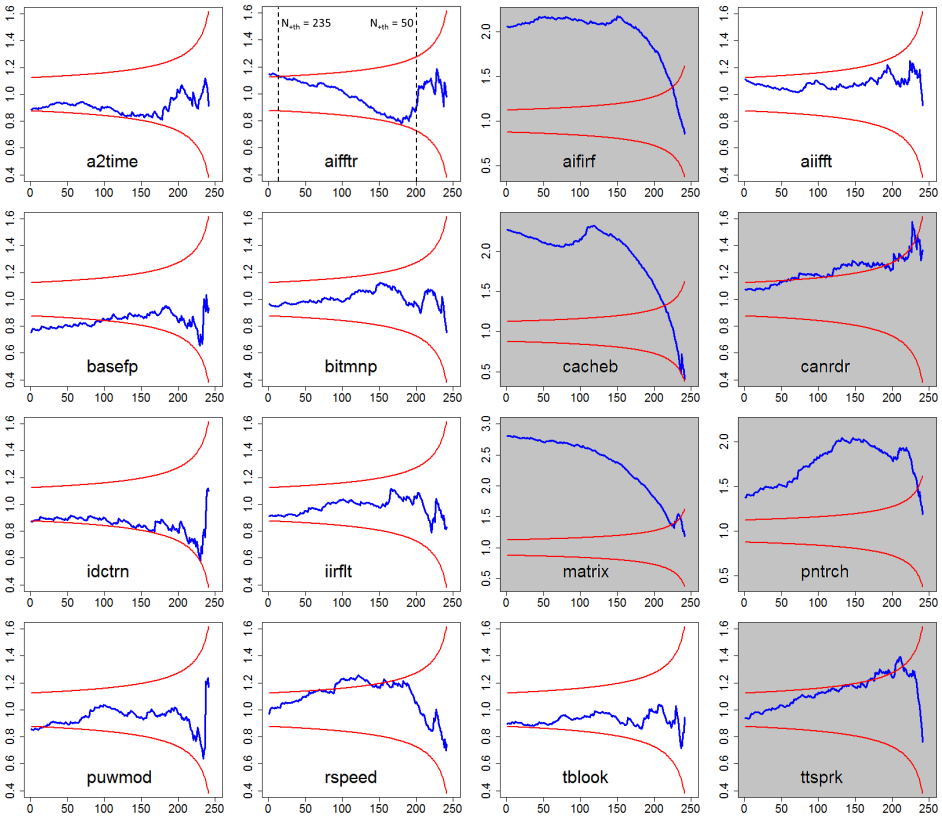


Fig. 11. CV-plots for all EEMBC with 500 samples. Recall that we take only the half with highest values (250 in this case). The x-axis shows the number of rejected values.

6.1.3. MBPTA-CV results. For a better understanding of the results, we start by showing in Table II the minimum (MIN), Quantile 1, median, Quantile 3, maximum (MAX) and standard deviation values for the execution times collected for each EEMBC benchmark. Values have been normalised to their respective median to facilitate the interpretation of the data.

We applied MBPTA-CV for samples of $R = 500$ runs. The CV-plots are shown in Figure 11. For instance in the case of `aifftr` when $N_{+th} < 235$ exponentiality cannot be rejected. Further, N_{+th} values smaller than 50 are not used for pWCET estimation purposes as explained before. For all number of exceedance values in the region $NE \in [50, 235]$, $N_{+th} = 147$ has been chosen as it provides the $cv(th)$ value closest to 1 (0.999 in particular). MBPTA-CV derived a N_{+th} value meeting *Const1* and *Const2* for all benchmarks but: `aifirf`, `cacheb`, `canrdr`, `matrix`, `pntrch` and `ttsprk`, whose CV-plot is highlighted in light-grey to easily identify them. Benchmarks `aifirf`, `cacheb`, `matrix` and `pntrch` fall within the exponentiality region for $N_{+th} < 50$, while the $cv(th)$ of `canrdr` and `ttsprk` enter and leave the exponentiality region several times. In all these cases, MBPTA-CV requests more execution time observations. To check the impact of this, we increased the size of the sample iteratively to 1,000, 2,000 and 3,000 until the criteria were fulfilled. The new CV-plots for those six benchmarks that needed larger sample sizes are depicted in Figure 12. As shown, MBPTA-CV finds a solution for all those six benchmarks with 1,000, 2,000 or 3,000 observations. The final sample size, $cv(th)$ and N_{+th} values for each benchmark are detailed in Table III. On average the number of runs is relatively small, around 1,063. Instead MBPTA-orig, which builds upon GEV and considers all potential values for bs, selects the one whose ξ is closest to 0 (and hence to the Gumbel distribution) as long as it is close enough not to reject the hypothesis of having a Gumbel distribution. MBPTA-orig starts with a sample of 100 runs and increases it iteratively by 50 until 5 consecutive sample sizes lead to close enough pWCET estimates – determined using the CRPS metric [Cucu-Grosjean et al. 2012] – for which it is considered that longer sample sizes would not provide further information on the shape of the execution time distribution. However, no consideration is given to the fact that using larger values for bs (so fewer blocks and hence values closer to the tail) may lead to Fréchet distributions. If larger values for bs (so fewer high values) led to a Fréchet distribution (so heavy tail), it would mean that some of those few values do not belong to the tail and thus, using more values (thus a lower bs value) includes non-tail values, which may diminish the quality of the pWCET estimate.

While both MBPTA-CV and MBPTA-orig select their respective parameters automatically, with the objective of finding accurate distributions resembling the tail, only MBPTA-CV takes into account the fact that data may include non-tail observations to reject some th values even if those th values passed the exponentiality test.

6.1.4. Comparison against observed distributions. Showing that a WCET estimation method is reliable in all accounts is hard: either due to the uncertainty brought about by the data used to feed static timing models (e.g. processor specifications, flow facts) or due to the quality of the data used to predict in measurement-based models (e.g. coverage and control of the experiments) [Paulitsch et al. 2015]. In order to provide additional evidence on the confidence of MBPTA-CV, we collected huge amounts of execution times for two EEMBC benchmarks and compared their distribution to the obtained pWCET curves. We collected 10^7 runs for each benchmark, each taking around 20 seconds per run. In a powerful cluster with 500 jobs running in parallel it took around 10 days to collect the specified number of runs. While performing such an experiment is not needed for MBPTA-CV (and it is infeasible in the general case), it provides evidence on how tight MBPTA-CV is w.r.t. observed values. Figure 13 shows that MBPTA-CV estimates tightly upper-bound the observed distributions with a difference at 10^{-7} of 1.1% and 25.8% respectively. We repeated the same approach with the other benchmarks for fewer runs (10^5) and observe the same trend.

6.1.5. MBPTA-CV vs MBPTA-orig. We used two metrics to compare both approaches.

Required number of runs: While MBPTA-CV requires on average 1,063 runs per benchmark, MBPTA-orig required 1,644 runs on average. We conclude that both approaches require a relatively low number of runs, which facilitates the applicability of MBPTA.

Tightness and Reliability: In order to compare MBPTA-CV and MBPTA-orig, we have collected the pWCET estimates for an exceedance threshold of 10^{-15} per run. In the case of MBPTA-CV we use the sample size reported in Table III, whereas for MBPTA-orig we use the sample size

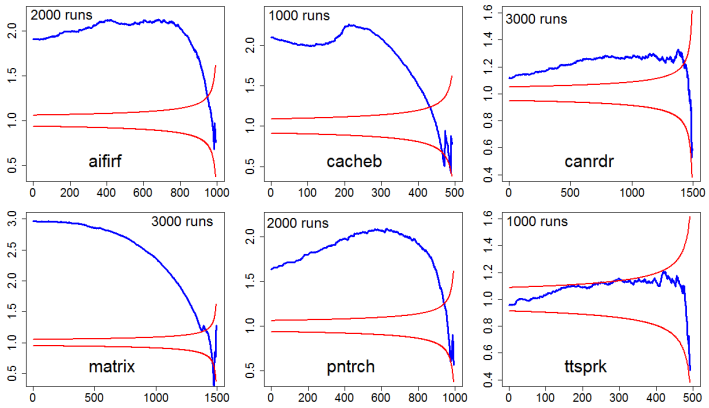


Fig. 12. CV-plot for `aifirf`, `cacheb`, `canrdr`, `matrix`, `pntrch` and `ttsprk`.

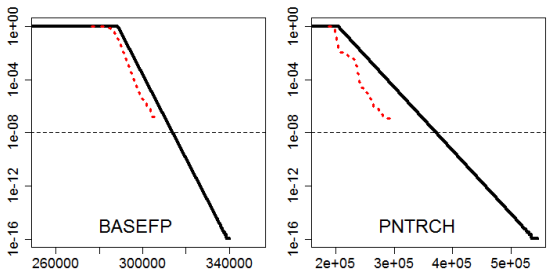


Fig. 13. pWCET distributions (black line) and empirical CCDF (dashed red line) for `basefp` and `pntrch`.

Table III. Results with MBPTA-CV for all EEMBC benchmarks.

Parameter	a2time	aifftr	aifirf	aiifft	basefp	bitmnp	cacheb	canrdr
Sample size	500	500	2,000	500	500	500	1,000	3,000
$cv(th)$	0.982	0.999	1.225	1.014	0.951	1.000	1.004	1.109
N_{+th}	53	147	51	179	69	170	52	57

Parameter	idctrn	iirflt	matrix	pntrch	puwmod	rspeed	tblook	ttsprk
Sample size	500	500	3,000	2,000	500	500	500	1,000
$cv(th)$	0.918	1.000	1.003	1.153	1.000	1.047	1.001	1.097
N_{+th}	196	129	60	50	135	52	58	96

automatically selected by the method itself. Table IV shows the pWCET estimates obtained with MBPTA-orig normalised w.r.t. those obtained with MBPTA-CV. We identify two scenarios.

Scenario A. For all benchmarks but `cacheb` and `matrix` both methods provide relatively similar distributions of the tails. However, discrepancies are relatively high sometimes. For instance, MBPTA-orig delivers a pWCET estimate 11% higher than MBPTA-CV for `a2time` and 5% lower for `ttsprk`. This results from the fact that, as explained in Section 4.2, MBPTA-orig uses the F&E approach to estimate σ and μ (based on sd and $\overline{x}$ respectively) for the Gumbel distribution, thus not fitting σ and μ when ξ is enforced to be 0.

— If $\xi < 0$ the tail is light to some extent. By enforcing $\xi = 0$ while keeping sd and $\overline{x}$ unmodified, the slope of the distribution smoothens and hence pWCET estimates obtained for a given exceedance threshold (e.g. 10^{-15}) are unnecessarily pessimistic. This is the case of `a2time` whose $\xi = -0.05$.

Table IV. MBPTA-orig pWCET estimates vs MBPTA-CV for EEMBC benchmarks.

a2time	aifftr	aifirf	aiifft	basefp	bitmnp	cacheb	canrdr
1.11	1.02	1.27	1.03	0.99	1.06	0.32	1.19

idctrn	iirflt	matrix	pntrch	puwmod	rspeed	tblook	ttsprk
0.99	1.03	0.35	1.02	0.97	0.93	1.01	0.95

— Conversely, if $\xi > 0$ the tail is somewhat heavy. Therefore, by enforcing $\xi = 0$ while keeping sd and $\overline{x}$ unmodified, the slope of the distribution sharpens and hence pWCET estimates become lower. This diminishes the quality of the pWCET estimate since evidence on being an actual upper-bound decreases. This is the case of `ttsprk` whose $\xi = 0.06$. Note that, as shown in Section 5.3, MBPTA-CV does not suffer this anomaly since λ is conveniently computed for the exponential distribution fitting the tail (E&F approach).

Scenario B. pWCET estimates for `cacheb` and `matrix` are much lower for MBPTA-orig than for MBPTA-CV. This occurs mainly due to two reasons: (1) MBPTA-orig converges with very large blocks so that very few values are obtained to fit the tail distribution. For both benchmarks the tail is approximated with only 7 blocks. Note that MBPTA-CV uses no less than 50 exceedance values. (2) Using so few values may lead to anomalous low variation because other (lower) tail values are discarded. Hence, sd (and the pWCET estimate) is much lower for MBPTA-orig than for MBPTA-CV. Overall, these two elements may lead to low-quality pWCET estimates. This can be addressed by either restricting the minimum number of blocks for MBPTA-orig or using MBPTA-orig with a larger sample size. For instance, for these two benchmarks we collected 10,000 runs and applied MBPTA-orig. Results show much higher pWCET estimates: 1.00 and 0.95 of those for MBPTA-CV for `cacheb` and `matrix` respectively, thus corroborating our two hypotheses. Remaining (small) differences can be explained with the same arguments as for *scenario A*.

Execution Time Requirements: Regarding computation time requirements, our implementation in the *R package* of both methods is very fast. Excluding the collection of the execution times, MBPTA-orig may take between less than 1 second and around 1 minute in a conventional laptop depending on the size of the sample needed to converge. This occurs because Gumbel distributions need to be fitted for each sample size and compared. Conversely, MBPTA-CV always took less than 1 second since comparing different sample sizes is basically done by comparing $cv(th)$ values that do not need fitting any distribution.

7. RELATED WORK

Timing Analysis techniques. Existing timing analysis techniques can be categorized into deterministic ones (DTA) [Wilhelm et al. 2008] that estimate a single WCET value, and probabilistic ones (PTA) that estimate a probabilistic WCET distribution. Each family of techniques comprises a static (SDTA, SPTA) and a measurement-based (MBDTA, MBPTA) variant. DTA advocates for time-deterministic platforms while PTA advocates for time-randomised ones. DTA and PTA have already been compared [Abella et al. 2014a], showing that there is not a dominant technique. It is also worth noting that this study was done under the assumption that each technique has all input data needed for an accurate analysis.

While in this paper we focus on MBPTA, SPTA has also been studied [Cazorla et al. 2013a; Altmeyer and Davis 2014] in the literature. However, so far SPTA focuses only on simplified architectures and requires similar information as SDTA (i.e. knowing cache placement of memory accesses). Thus, SPTA is not competitive against SDTA yet.

In the area of MBDTA, different approaches on execution time measurement collection are discussed and compared in [Petters 2003]. Some authors specifically focus on collecting measurements for program segments and compose them conveniently [Wenzel et al. 2008; Betts et al. 2010]. Some other works aim at producing test data automatically to reduce the burden on the user [Kirner et al. 2004; Wenzel et al. 2008; Wenzel et al. 2005].

Representativeness. EVT has been used to estimate the WCET of programs running on top of non-MBPTA-compliant architectures [Edgar and Burns 2001; Hansen et al. 2009; Bernat and Newby 2006; Santinelli et al. 2014]. The main challenge of those architectures is providing evidence of the representativeness of the execution time observations passed to EVT, which is a critical step in MBPTA (see Section 2). To the best of our knowledge, the representativeness challenge has not been studied on non-MBPTA platforms yet [Cazorla et al. 2013b]. Hence, results lack evidence on whether the analysis-time measurements capture those events that can arise during operation, thus impacting the quality of the pWCET estimates.

MBPTA-orig [Cucu-Grosjean et al. 2012; Cucu et al. 2013] has been positively assessed in industrial case studies [Wartel et al. 2015]. In this work, we identify some potential weaknesses of MBPTA-orig when some events are under-represented in the execution time sample passed as input to EVT. In the setup considered in this paper, this situation arises when using a large L2 cache together with the EEMBC benchmarks, whose working set is very small. This made several programs have high execution times only in few runs due to L2 cache conflicts. Instead, previous works with benchmarks with smaller L2 caches (or no L2 cache at all) and with larger industrial case studies are not necessarily subject to this effect since conflicts occurred with higher probabilities (or could not occur at all). However, the scenarios described in this paper show that state-of-the-art MBPTA techniques may face scenarios resulting in low-quality pWCET estimates. We also identify those scenarios as well as their cause. We further propose a new MBPTA technique particularly suited to tackle those issues, thus resulting in high quality pWCET estimates.

Recently, it has been shown that, when MBPTA is applied to MBPTA-compliant platforms, low-quality WCET estimates can be obtained if events with high impact on execution time and low probability are not captured [Reineke 2014]. This has been properly addressed in [Abella et al. 2014b; Benedicte et al. 2016; Milutinovic et al. 2016]. Authors in [Abella et al. 2014b; Benedicte et al. 2016; Milutinovic et al. 2016] propose methods to capture those timing events so that they are conveniently represented in the execution time measurements used by MBPTA. This has been done for both hardware and software time-randomised caches, either considering only the objects accessed by the program or considering also their access patterns.

EVT assumptions and application domain. In this work input data used by MBPTA (and so by EVT) is independent, although some works show that this is not strictly necessary if maxima are independent or if the dependence is weak [Coles 2001; Santinelli et al. 2014]. Incorporating this technology could broaden the application of MBPTA in general and MBPTA-CV in particular.

As we detailed in Section 4.1, other authors have shown that heavy tails may be needed for execution time modelling [Lima et al. 2016]. However, they build their work upon premises different to those imposed by MBPTA and by usual WCET analysis [Wilhelm et al. 2008].

Path coverage. pWCET estimation is orthogonal to the problem of path coverage. In general, one cannot rely on path frequency observed at analysis time to derive probabilities of occurrence of each path during operation. To obtain full path coverage with MBPTA, three methods exist: (1) applying MBPTA to each path regarded as relevant by the end user and deploy as pWCET the curve upperbounding the curves of all those paths. This choice is referred to as probabilistic envelope. (2) Collecting measurements on an extended version of the target program whose timing upper-bounds all possible paths of the original program [Kosmidis et al. 2014], and applying MBPTA-CV on those measurements. And (3) using Extended Path Coverage (EPC) [Ziccardi et al. 2015], which increases execution time measurements of each basic block to account for their worst initial state. Then, those measurements can be combined to produce end-to-end measurements of *all* paths so that MBPTA-CV can be applied on them, as in the probabilistic envelope.

8. CONCLUSIONS

In this paper we investigate the applicability of Extreme Value Theory (EVT) to the estimation of the WCET of real-time programs. We tailor EVT application to the particular problem at hand with the following three contributions. First, we analyse the (execution time) distributions that are passed to EVT. In particular, we analyse the resources that can produce mixture distributions, which

challenge EVT application. We also analyse the boundary conditions of execution time distributions (e.g. existence of maxima) affecting the applicability of EVT. Second, building on those analyses, we tailor EVT (i.e. GEV/GPD) parameters to estimate the WCET of real-time programs during operation based on a sample of execution times captured during system analysis. And third, we propose and evaluate MBPTA-CV, a new (automatic) mixture-distribution aware, MBPTA method for the WCET estimation of real-time programs.

REFERENCES

J. Abella, J. del Castillo, F.J. Cazorla, and M. Padilla. 2015. Extreme value theory in computer sciences: The case of embedded safety-critical systems. In *International Conference on Risk Analysis (ICRA)*.

J. Abella, D. Hardy, I. Puaut, E. Quinones, and F.J. Cazorla. 2014a. On the Comparison of Deterministic and Probabilistic WCET Estimation Techniques. In *ECRTS*.

J. Abella, E. Quinones, F. Wartel, T. Vardanega, and F.J. Cazorla. 2014b. Heart of Gold: Making the Improbable Happen to Extend Coverage in Probabilistic Timing Analysis. In *ECRTS*.

S. Altmeyer and R. I. Davis. 2014. On the Correctness, Optimality and Precision of Static Probabilistic Timing Analysis. In *DATE*.

H. Anwar, C. Chen, and G. Beltrame. 2015. A probabilistically analysable cache implementation on FPGA. In *NEWCAS*.

P. Benedicte, L. Kosmidis, E. Quinones, J. Abella, and F.J. Cazorla. 2016. A Confidence Assessment of WCET Estimates for Software Time Randomized Caches. In *INDIN*.

G. Bernat, A. Colin, and S.M. Petters. 2002. WCET analysis of probabilistic hard real-time systems. In *RTSS*.

G. Bernat and M. Newby. 2006. Probabilistic WCET analysis, an approach using copulas. *Journal of Embedded Computing* (2006).

A. Betts, N. Merriam, and G. Bernat. 2010. Hybrid measurement-based WCET analysis at the source level using object-level traces. In *WCET Analysis Workshop*.

G.E.P. Box and D.A. Pierce. 1970. Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *J. Amer. Statist. Assoc.* (1970).

BSC. 2017. Technical Report UPC-DAC-RR-CAP-2017-3. (2017). http://www.ac.upc.edu/RR/2017/3.pdf.

D. Buttle. 2012. ETAS GmbH, Germany, Real-Time in the Prime-Time. Keynote talk.. In *ECRTS*.

Y. Cai and D. Hames. 2011. Minimum sample size determination for generalized extreme value distribution. *Communications in Statistics Simulation and Computation* 40 (2011), 99 – 110.

F.J. Cazorla, E. Quinones, T. Vardanega, L. Cucu, B. Triquet, G. Bernat, E. Berger, J. Abella, F. Wartel, M. Houston, L. Santinelli, L. Kosmidis, C. Lo, and D. Maxim. 2013a. PROARTIS: Probabilistically Analysable Real-Time Systems. *ACM Trans. on Embedded Computing Systems* (Dec. 2013).

F.J. Cazorla, T. Vardanega, E. Quinones, and J. Abella. 2013b. Upper-bounding Program Execution Time with Extreme Value Theory. In *WCET Analysis Workshop*.

R. T. Clarke. 2002. Fitting and testing the significance of linear trends in Gumbel-distributed data. *Hydrology and Earth System Sciences* 6, 1 (2002), 17–24. DOI:http://dx.doi.org/10.5194/hess-6-17-2002

Cobham Gaisler. *NGMP Preliminary Datasheet Version 2.1, May 2013*.

S. Coles. 2001. *An Introduction to Statistical Modeling of Extreme Values*. Springer.

L. Cucu, A. Gogonel, and C. Lo. 2013. PROARTIS. D3.6 Probabilistic and Statistical Techniques for Timing Analysis in Multicore. (2013).

L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Quinones, and F.J. Cazorla. 2012. Measurement-Based Probabilistic Timing Analysis for Multi-path Programs. In *ECRTS*.

J. Del Castillo, J. Daoudi, and R. Lockhart. 2014. Methods to Distinguish Between Polynomial and Exponential Tails. *Scandinavian Journal of Statistics* 41, 2 (2014), 382–393. DOI:http://dx.doi.org/10.1111/sjos.12037

G. Edelin. 2009. Embedded Systems at THALES: the Artemis challenges for an industrial group. In *ARTIST*.

S. Edgar and A. Burns. 2001. Statistical Analysis of WCET for Scheduling. In *RTSS*.

W. Feller. 1996. *An introduction to Probability Theory and Its Applications*.

M. Garrido and J. Diebolt. 2000. The ET test, a goodness-of-fit test for the distribution tail. In *conference on mathematical methods in reliability. Methodology, Practice and Inference*.

M. Gilli and E. Këllezi. 2006. An Application of Extreme Value Theory for Measuring Financial Risk. *Computational Economics* 27, 2 (2006).

J. Hansen, S. Hissam, and G. A. Moreno. 2009. Statistical-Based WCET Estimation and Validation. In *WCET Analysis Workshop*.

International Organization for Standardization. 2009. *ISO/DIS 26262. Road Vehicles – Functional Safety*.

J. Jalle, L. Kosmidis, J. Abella, E. Quinones, and F.J. Cazorla. 2014. Bus Designs for Time-Probabilistic Multicore Processors. In *DATE*.

JEDEC. 2008. *DDR2 SDRAM Specification JEDEC Standard No. JESD79-2E*.

R. Kirner, P. Puschner, and I. Wenzel. 2004. Measurement-based worst-case execution time analysis using automatic test-data generation. In *SEUS Workshop*.

L. Kosmidis, J. Abella, E. Quinones, and F.J. Cazorla. 2013. A Cache Design for Probabilistically Analysable Real-time Systems. In *DATE*.

L. Kosmidis, J. Abella, F. Wartel, E. Quinones, A. Colin, and F. J. Cazorla. 2014. PUB: Path Upper-Bounding for Measurement-Based Probabilistic Timing Analysis. In *ECRTS*.

L. Kosmidis, C. Curtsinger, E. Quinones, J. Abella, E. Berger, and F.J. Cazorla. 2013. Probabilistic Timing Analysis on Conventional Cache Designs. In *DATE*.

L. Kosmidis, E. Quinones, J. Abella, T. Vardanega, I. Broster, and F.J. Cazorla. 2014. Measurement-Based Probabilistic Timing Analysis and Its Impact on Processor Architecture. In *Euromicro DSD*.

S. Kotz and S. Nadarajah. 2000. *Extreme value distributions: theory and applications*. World Scientific. 185 pages.

G. Lima, D. Dias, and E. Barros. 2016. Extreme Value Theory for Estimating Task Execution Time Bounds: A Careful Look. In *ECRTS*.

Y. Lu, I. Bate, T. Nolte, and L. Cucu-Grosjean. 2011. A New Way about using Statistical Analysis of Worst-Case Execution Times. *ACM SIGBED Review* (September 2011).

Y. Lu, T. Nolte, I. Bate, and L. Cucu-Grosjean. 2012. A Statistical Response-Time Analysis of Real-Time Embedded Systems. In *RTSS*. 351–362.

E. Mezzetti and T. Vardanega. 2011. On the industrial fitness of WCET Analysis. *WCET Workshop* (2011).

E. Mezzetti and T. Vardanega. 2013. A rapid cache-aware procedure positioning optimization to favor incremental development. In *RTAS*.

S. Milutinovic, J. Abella, and F.J. Cazorla. 2016. Modelling Probabilistic Cache Representativeness in the Presence of Arbitrary Access Patterns. In *ISORC*.

J. Nowotsch, M. Paulitsch, D.B. Uhler, H. Theiling, S. Wegener, and M. Schmidt. 2014. Multi-core Interference-Sensitive WCET Analysis Leveraging Runtime Resource Capacity Enforcement. In *ECRTS*.

J. Owens. 2015. Delphi Automotive, The Design of Innovation That Drives Tomorrow. Keynote talk.. In *DAC*.

M. Paolieri, E. Quinones, F.J. Cazorla, G. Bernat, and M. Valero. 2009b. Hardware Support for WCET Analysis of Hard Real-Time Multicore Systems. In *ISCA*.

M. Paolieri, E. Quinones, F. J. Cazorla, and M. Valero. 2009a. An Analyzable Memory Controller for Hard Real-Time CMPs. *IEEE Embedded Systems Letters* 1, 4 (Dec 2009), 86–90.

M. Patte and V. Lefftz. 2011. *System Impact of Distributed Multi Core Systems.* Technical Report Contract 4200023100. European Space Agency.

M. Paulitsch, O. Medina, H. Karray, K. Mueller, D. Muench, and J. Nowotsch. 2015. Mixed-Criticality Embedded Systems - A Balance Ensuring Partitioning and Performance. In *Euromicro DSD*.

S.M. Petters. 2003. Comparison of Trace Generation Methods for Measurement Based WCET Analysis. In *WCET Analysis Workshop*.

J. Poovey. 2007. *Characterization of the EEMBC Benchmark Suite*. North Carolina State University.

J. Reineke. 2014. Randomized Caches Considered Harmful in Hard Real-Time Systems. *Leibniz Transactions on Embedded Systems* 1, 1 (2014).

SAE International. 2001. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment. *ARP4761* (2001).

L. Santinelli, J. Morio, G. Dufour, and D. Jacquemart. 2014. On the Sustainability of the Extreme Value Theory for WCET Estimation. In *WCET analysis workshop*.

SoCLib. 2003-2012. -. (2003-2012). http://www.soclib.fr/trac/dev.

F. Wartel et al. 2015. Timing Analysis of an Avionics Case Study on Complex Hardware/Software Platforms. In *DATE*.

I. Wenzel, R. Kirner, B. Rieder, and P. Puschner. 2008. Measurement-Based Timing Analysis. In *ISOLA*.

I. Wenzel, B. Rieder, R. Kirner, and P. Puschner. 2005. Automatic Timing Model Generation by CFG Partitioning and Model Checking. In *DATE*.

R. Wilhelm et al. 2008. The worst-case execution time problem: overview of methods and survey of tools. *Trans. on Embedded Computing Systems* 7, 3 (2008), 1–53.

M. Ziccardi, E. Mezzetti, T. Vardanega, J. Abella, and F. J. Cazorla. 2015. EPC: Extended Path Coverage for Measurement-based Probabilistic Timing Analysis. In *RTSS*.