A Probabilistic Approach to Energy-Constrained Mixed-Criticality Systems

Federico Reghenzani*, Giuseppe Massari*, William Fornaciari*
* *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano*, Milano, Italy
name.surname@polimi.it

***Abstract*—In battery-powered embedded systems, the energy budget management is a critical aspect. For systems using unreliable power sources, e.g. solar panels, the continuous system operation is a challenging requirement. In such scenarios, effective management policies must rely on accurate energy estimations. In this paper we propose a measurement-based probabilistic approach to address the worst-case energy consumption (WCEC) estimation, coupled with a job admission algorithm for energy-constrained task scheduling. The overall goal is to demonstrate how the proposed approach can introduce benefits also in mission-critical systems, where unsafe energy budget estimations cannot be tolerated.**

I. Introduction

The increasing performance demand in modern embedded systems, under thermal and power budget constraints, is introducing growing challenges, often addressed with solutions that increase also the hardware complexity. In critical systems, where *a priori* analyses are required, this complexity has a direct (negative) impact on the accuracy of power and timing models [2]. Moreover, without a proper characterization of the workload, such models can easily lead to unrealistic results [3]. Focusing on *energy-constrained* systems, the accurate estimation of the system energy consumption plays a key role. This is especially true when the main requirement is to guarantee the system operation in presence of unreliable power sources, which may also be the case of critical systems [4].

In this context, battery-powered devices are increasing their pervasiveness, while posing some challenges: what shall we do when the energy budget is not sufficient to execute all the tasks? Asyaban et al. [5], for instance, proposed a scheduler for mixed-criticality tasks, for scenarios in which the energy budget is subject to uncertainty. Even though the *mixed-criticality* concept is frequently intended in a timeliness-sense [6], in this work we enlarge the scope by considering the *energy* as the most critical resource.

Related Works. To the best of our knowledge, the first work on WCEC analysis has been presented in 2006 [7]. The estimation was based on static code analysis and energy models at micro-architectural level. Experimental results showed WCEC overestimations up to 30%. More recently, the *0g* tool [8] presented two approaches to WCEC estimation: a static code analysis and a measurement-based technique. The former is used to characterize the energy consumption of critical tasks. The latter is used to estimate the consumption in case of non-critical tasks, without any guarantees on the reliability of the results. Moreover, the employment of the static code analysis requires the availability of a platform-specific per-instruction characterization of the energy consumption. Recently, static WCEC analyses have been system-wide extended, including peripherals [9]. The recent work of Pallister et al. [10] is similar to our approach, because the estimation of the WCEC, at instruction-level, is performed by fitting a Weibull distribution. However, the selection of this single-class distribution and the focus on instruction-level limit the applicability to the full-system WCEC estimation. Finally, [11] presented an analysis of the limitations of the current WCEC analysis techniques, demonstrating that it is a NP-hard problem and no efficient approximation algorithms exist. Furthermore, the authors suggested the possibility of moving towards statistical methods, like the ones based on Extreme Value Theory (EVT) for WCET analyses [12], that is exactly the cornerstone of our approach.

Contributions. To the best of our knowledge, this is the first paper that actually applies EVT to WCEC estimation. Based on the results of the analysis, we propose also an energy-constrained job admission policy for task scheduling. To assess the theoretical results, we performed an experimental validation on a real multi-core based embedded board, running multi-threaded benchmarks.

II. Task and System Model

This section presents the task and the system model on which the WCEC analysis and the proposed policy are based.

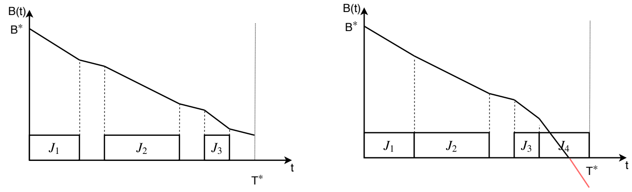
Mixed-Energy Criticality Task Model. A set of n periodic tasks is identified as: $\boldsymbol{\tau} = \{\tau_1, \tau_2, ..., \tau_n\}$. Each task activation is called *job* and it is represented by the notation $J_{i,j}$, to identify the j-th job of the i-th task. In this work, we consider periodic and/or sporadic tasks but not the aperiodic case, assuming that the task periodicity is known *a priori*.

Before defining the considered task model, we recall the simplest task model used in mixed *time*-criticality theory: the tuple $\tau_i = (X_i, D_i, T_i, L_i)$, where X_i is the worst-case execution time[1], D_i is the task relative deadline, T_i

[1]In some mixed-time criticality task models X_i is a set of WCETs, one for each criticality level. However, this discussion is outside the scope of this paper.

This research was funded by EU H2020 project RECIPE [1] (grant 801137)

978-1-7281-2954-9/19/$31.00 ©2019 IEEE



(a) Energy-feasible scenario. (b) Non energy-feasible scenario

Fig. 1: Budget function $B(t)$ examples varying the number of tasks in the survival period T^*.

is the period (or the minimal inter-arrival time in case of sporadic tasks), and L_i is the criticality level. The latter can assume different types of values, such as integer numbers, real numbers, or categorical values like *HI* and *LO*.

In mixed-*time* criticality systems, the criticality is focused on the timeliness. In this paper, since we consider the energy as the critical resource, we deal with mixed-*energy* criticality systems without explicitly taking into account time constraints. This energy-criticality concept is similar to the one proposed by Völp et al. [13]. The time constraints can be considered by a state-of-the-art real-time scheduler *a posteriori* of the results of this work. To this aim, we adapted the traditional task model as follows: $\tau_i = (E_{i,j}, T_i, L_i)$ where $E_{i,j}$ represents the energy required to execute the j-th job, T_i is the activation period, and L_i is the critical level. However, since the exact value of $E_{i,j}$ is seldom predictable, it is a common practice to replace it with the worst-case energy consumption (WCEC). All the jobs of the same task have the same WCEC value identified by the symbol E_i. It is important to specify that the energy consumption value of a task $E_{i,j}$ and its worst-case E_i do not include the energy consumption of the system in idle state. In other words, we consider the task energy consumption as an additional contribution with respect to the energy consumption of the system in idle.

For the criticality level, we use the labels *HI* and *LO*, to respectively identify *high-criticality* tasks, for which jobs execution must guaranteed, and *low-criticality* tasks, for which the jobs execution can be guaranteed only if enough energy budget is actually available. If a job is not executed due to lack of energy, we say that this task has been *dropped*, conversely we say that the task has been *admitted* to the scheduling queue.

Energy-Constrained System Model. Following the same approach proposed by Völp et al. [13], the system model is characterized by two parameters: the energy budget B^* and the survival period T^*. This notation denotes that the system is allowed to consume a maximum B^* of energy in the continuous period of time T^*. In order to make the subsequent theoretical discussion easier, we define the *budget function* $B(t)$ as the total amount of remaining energy budget as a function of time, with $B(0) = B^*$.

A couple of examples of the budget function are depicted in Figure 1. This function decreases at least at the rate of the power consumption in idle state. We use E_{IDLE} to indicate the energy consumption of the system in idle for the whole timespan $(0, T^*)$. Consequently, it holds that $B(0) - B(T^*) > E_{IDLE}$ regardless of any job admission or scheduling decision.

Possible use cases of the presented energy-constrained mixed-criticality models include several battery-powered applications, such as devices powered by unreliable energy harvesting sources, mobile devices and space applications [14].

A. Scheduling Model

In this paper, the *high-level schedule (HLS)* identifies the set $\boldsymbol{S} = \{S_{1,j}, S_{2,j}, ..., S_{n,j}\}$ where $S_{i,j}$ is a boolean value indicating if the job $J_{i,j}$ can be scheduled or not. Therefore, $S_{i,j} = 1$ means that $J_{i,j}$ is allowed to run, while $S_{i,j} = 0$ indicates that the job is dropped. The traditional notion of *schedule*, including the order in which to run the tasks, is out of the scope of this work and it can be performed by any state-of-the-art scheduler having the $\boldsymbol{S}$ set as input.

Definition 1. *A HLS* $\boldsymbol{S}$ *is said to be* energy-feasible *if the energy budget is sufficient to run all the jobs, thus* $B(T*) \geq 0$.

Definition 2. *A HLS* $\boldsymbol{S}$ *is said to be* correct *if and only if all the HI-crit tasks are scheduled:* $S_{k,j} = 1 \quad \forall \tau_k$ *s.t.* $L_k = HI$.

These definitions are the basis for the first schedulability requirement, called *minimal energy-schedulability condition*:

Lemma 1. *An energy-feasible and correct HLS exists only if:*

$$\sum_{k \text{ s.t. } L_k = HI} \sum_j E_{k,j} < B(0) - E_{IDLE} \tag{1}$$

where $E_{k,j}$ *is the energy consumed by the* j*-th job of the* k*-th HI-crit task. This energy does not include the static energy consumed by the system in idle* E_{IDLE}. □

Lemma 1 states that the energy budget must be sufficient to guarantee at least the execution of the HI-crit tasks, otherwise no energy-feasible and correct HLS exists. Since the energy of the single job $E_{i,j}$ is hard to predict, we rely on the upper-bound represented by the WCEC of the task E_i. Unfortunately, as already discussed in Section 1, estimating E_i in modern architectures running complex workload is either not trivial or it leads to very over-approximated results. Moreover, the introduction of the WCEC value in Lemma 1 would mean to consider an extremely pessimistic scenario, in which all the jobs consume the worst-case amount of energy, that seldom occurs on real systems. These reasons motivated our idea of evaluating the adoption of probabilistic approaches for estimating the WCEC (E_i), as already done, similarly, with WCET analyses [12].

III. Probabilistic Energy Estimation

In this section, we show how to determine a reliable not underestimated WCEC value (E_i), by exploiting a probabilistic measurement-based approach, while maintaining system model and requirements unchanged.

Measurement-Based Methodology. The probabilistic model we propose in this paper is based on the *Extreme Value Theory (EVT)*. This statistical theory is commonly used to predict natural disasters, such as earthquakes intensity or river water levels; it has been also considered in real-time computing to estimate the Worst-Case Execution Time (WCET) [12]. The EVT is a measurement-based method used to infer the WCET by directly sampling the task execution times. In this work, we apply this methodology to estimate the WCEC of the jobs of each task.

A measurement-based approach comes with important advantages: (1) the system is considered as a black-box (accurate platform models are not required); (2) no need to perform in-depth static analyses of the task source code; (3) the output is a *worst-case* estimation (and not a *mean-case* estimation) within a given level of confidence, that represents the probability of failure to meet the energy budget requirement B^*. While point (2) enables the possibility to estimate the WCEC for previously unknown tasks, point (3) is a fundamental requirement for mission-critical systems. However, in order to obtain a small probability of failure, e.g. 10^{-9}, traditional measurement-based methods would need a huge amount of samples. This requires to run an extremely high number of jobs in a Monte Carlo fashion. Unfortunately, this is often unfeasible, which justifies the use of EVT briefly described in the next paragraphs.

1) Extreme Value Theory (EVT): This theory has been developed to overcome the limitations of the well-known Central Limit Theorem. Although the theorem is widely used to approximate the mean value of an observed phenomena, it does not fit when we look at the distribution tail, i.e. worst-case events. The cornerstone of the overall EVT is the following theorem formulated in the 1920s-1950s [15]:

Theorem 1 (Fisher-Tippett-Gnedenko theorem)**.** *Given a sequence of i.i.d.*[2] *random variables $X_1, X_2, ..., X_n$, the random variable representing their maxima $M_n = max(X_1, X_2, ..., X_n)$ converges to the Weibull, Gumbel or Fréchet distribution for $n \to \infty$.*

It has been later proved [16] that these three forms can be generalized to a single distribution called *Generalized Extreme Value (GEV)* distribution. This distribution represents the extreme values, i.e. the probability that unseen rare events happen. It is identified by three parameters $GEV(\mu, \sigma, \xi)$ and it has the following *cumulative distribution function (cdf)*:

$$G(x) = \begin{cases} e^{-e^{\frac{x-\mu}{\sigma}}} & \xi = 0 \\ e^{-[1+\xi(\frac{x-\mu}{\sigma})]^{-1/\xi}} & \xi \neq 0 \end{cases} \tag{2}$$

In order to estimate the GEV parameters, two main techniques have been developed: the *Block-Maxima (BM)* and the *Peak-over-Threshold (PoT)*[3]. Both methods apply a filter to the input data $X_1, X_2, ..., X_n$, to obtain a new sequence of random variables $Y_1, Y_2, ..., Y_m$ with $m < n$. In the first case, BM, the set is divided in blocks of constant size B and the maximum of each block is kept, while the other values are discarded. In the latter case, PoT, a threshold u is set, such that all the under-threshold values are discarded. The resulting sequence $Y_1, Y_2, ..., Y_m$ is the input of a well-known estimator, e.g. the Maximum Likelihood Estimator (MLE), from which we obtain an estimation of the parameters of the GEV distribution [17].

2) Probabilistic Worst-Case Energy: Let us assume we measured several times the energy consumed for executing the jobs of a specific task. The energy samples $x_1, x_2, ..., x_n$ represent the realization of the input random variables of Theorem 1. These values can be fed into the previously described EVT process, to obtain a GEV distribution representing the probability of extreme events. Through this distribution, it is possible to exploit its *inverse complementary cumulative distribution function (iccdf)* to obtain an estimation of the WCEC, given a violation probability level p: $\text{iccdf}(p) = F'(1-p) = \{\text{WCEC s.t. } p = P(X > \text{WCEC})\}$ where X is the random variable representing the job energy consumption.

Probabilistic Task Model. By exploiting the EVT method described in Section III, it is possible to compute the two random variables $\mathcal{E}_i$ and $\mathcal{E}_{IDLE}$, respectively representing the worst-case value of E_i and E_{IDLE}. Thanks to this measurement-based approach we can verify the condition of Lemma 1 without having accurate models of system and workload. The task model can then be rewritten as: $\tau_i = (\mathcal{E}_i, T_i, L_i)$. However, the application of the schedulability condition in Lemma 1 is now complicated by the introduction of random variables. To combine random variables, two possible approaches can be considered:

1) Set a value for the probability of failure p and use the iccdf of $\mathcal{E}_i$ to obtain scalar worst-case values for E_i;
2) Perform the actual sum of the random variables.

The first option is the simplest one, but it probably leads to excessive overestimation, since we always considers the worst-case value for all the jobs. The second option instead requires the application of the *convolution* operator between random variables. In this work, we consider the second option, since it allows us to obtain a tight estimation of the real WCEC. Given two random variables $\mathcal{A}$ and $\mathcal{B}$, the convolution of their cdfs is $\text{cdf}_C(x) = \text{cdf}_A(x) \circledast \text{cdf}_B(x)$, where $\circledast$ is the convolution operator. For the convolution computation, the following integral has to be solved: $\text{cdf}_C(x) = \int \text{cdf}_A(x-y)\ d\text{cdf}_B(y)$. This is hard to obtain analytically, especially for GEV distributions, but easy to obtain using one of the several numerical algorithms available in literature. Regarding the $\mathcal{E}_{IDLE}$ value, we simply compute the worst-case value at the predefined probability failure p, i.e. $\bar{E}^p_{IDLE} = \text{iccdf}_{\mathcal{E}_{IDLE}}(p)$.

By using the probabilistic model just defined, the previous Lemma can be rewritten as:

Lemma 2. *Given the set $\{\mathcal{E}^*_k\}$ of random variables defined as:*

$$\mathcal{E}^*_k = \overset{nr\_jobs}{\circledast} \mathcal{E}_k \qquad \forall k \text{ s.t. } \tau_k \in \boldsymbol{\tau} \tag{3}$$

there exists an energy-feasible and correct schedule, with a

[2]independent and identically distributed

[3]To be precise, using the PoT approach the distribution of maxima converges to another distribution called Generalized Pareto Distribution (GPD) that is, however, asymptotically equivalent to the GEV.

violation probability p only if:

$$\sum_{k \text{ s.t. } L_k=HI} \bar{E}_k^p < B(0) - \bar{E}_{IDLE}^p \tag{4}$$

where $\bar{E}_k^p$ is the WCEC value of all the jobs of the k-th task, with violation probability p, computed by using the convoluted random variables: $\bar{E}_k^p := iccdf_{\mathcal{E}_k^*}(p)$. The term $\bar{E}_{IDLE}^p$ represents instead the WCEC estimation in idle mode. □

If the number of tasks is extremely high, the use of another convolution operator replacing the sum operator in Equation 4 should be considered to reduce the over-approximation. For convenience, we indicate the left hand term of the Equation 4, i.e. the WCEC of all the HI-crit tasks, with $\bar{E}_{HI}$.

IV. Energy-aware Task Scheduling

In this section, we focus on the job admission problem, i.e. to whether a job can enter the scheduling queue or not. In mixed-criticality systems, the admission of HI-crit jobs must be guaranteed in any case. Consequently, we need to define a policy for the admission of LO-crit tasks, on the basis of the energy required by the tasks and the system energy budget available. We can derive the energy budget for LO-crit tasks from Lemma 2, as it follows:

$$B^{LO}(t) = B(t) - \bar{E}_{HI}. \tag{5}$$

Accordingly, if there exists an energy-feasible and correct HLS, then it is always $B^{LO}(t) \geq 0$. Following the same probabilistic approach of HI-crit tasks, we can state the *maximum energy consumption condition* for LO-crit tasks as:

Lemma 3. *Given a failure probability p and the survival period T^*, LO-crit tasks energy consumption upper bound is:*

$$\sum_{k \text{ s.t. } L_k=LO} \bar{E}_k(p) < B^{LO}(0) - \bar{E}_{IDLE}^p \tag{6}$$

We refer to the left-hand part of Equation 6 as $\bar{E}_{LO}$, deriving the overall remaining unused energy budget as follows:

$$B(T^*) = B(0) - \bar{E}_{HI} - \bar{E}_{LO} - \bar{E}_{IDLE} \tag{7}$$

The *job admission algorithm* assigns the values to the set $S_{k,j}$, in accordance with the previous conditions. Relying on the previous WCEC estimations, this section aims at proposing a policy that minimizes $B(T^*)$, while maintaining a fair energy distribution among LO-crit tasks. An algorithm carrying out the minimal positive value of $B(T^*)$, among the values carried out by all the algorithms, is said to be *optimal*.

The job admission pseudo-code is shown in Algorithm 1. The policy considers the scheduling of HI-crit tasks first, followed by the LO-crit ones. The algorithm sets $S_{i,j} = 1$ for all the jobs of the HI-crit tasks (lines 10-12), so that it complies with the HLS correctness requirement (Definition 2). The remaining energy budget B is reduced by the WCEC value estimated for the scheduled task (with violation probability p) (line 13). If $B < 0$, no energy-feasible and correct HLS exists (line 14).

Algorithm 1 Mixed-Energy Criticality Job Admission

1: **Input**: τ_i task set, $\bar{E}_k^p$ WCEC of task k, $\bar{E}_{\text{IDLE}}^p$ WCEC of idle system, B^* energy budget and T^* survival period, p violation probability.
2: **Output**: $S_{k,j}$ (the HLS).
3: **procedure** SCHEDULE
4: $\quad B \leftarrow B^* - \bar{E}_{\text{IDLE}}^p$
5: $\quad B^{LO} \leftarrow$ ScheduleHITasks(B)
6: $\quad B^{NULL} \leftarrow$ ScheduleLOTasks(B^{LO})
7: **end procedure**
8: **procedure** SCHEDULEHITASKS(B)
9: $\quad$ **for all** k s.t. $L_k = HI$ **do**
10: $\quad\quad$ **for all** j job of task k **do**
11: $\quad\quad\quad S_{k,j} \leftarrow 1$
12: $\quad\quad$ **end for**
13: $\quad\quad B \leftarrow B - \bar{E}_k(p)$
14: $\quad\quad$ assert($B > 0$)
15: $\quad$ **end for**
16: $\quad$ **return** B
17: **end procedure**
18: **procedure** SCHEDULELOTASKS(B)
19: $\quad \boldsymbol{\tau}^* \leftarrow$ sort $\{\tau_i : L_i = LO\}$ by $\bar{E}_i^p$ ascending order
20: $\quad n \leftarrow$ size($\boldsymbol{\tau}^*$)
21: $\quad$ **for all** $\tau_i \in \boldsymbol{\tau}^*$ **do**
22: $\quad\quad B_i \leftarrow \frac{B}{n}$
23: $\quad\quad$ **if** $\bar{E}_i^p \leq B_i$ **then**
24: $\quad\quad\quad$ **for all** job j in τ_i **do**
25: $\quad\quad\quad\quad S_{i,j} \leftarrow 1$
26: $\quad\quad\quad$ **end for**
27: $\quad\quad$ **else**
28: $\quad\quad\quad$ nr_jobs $\leftarrow$ max_jobs(τ_i, B_i)
29: $\quad\quad\quad \mathcal{E}_i^* \leftarrow \circledast_j^{\text{nr\_jobs}} \mathcal{E}_i$
30: $\quad\quad\quad \bar{E}_i^p \leftarrow \text{iccdf}_{\mathcal{E}_i^*}(p)$
31: $\quad\quad\quad S_{i,\{1,2,\ldots,m\}} \leftarrow$ selection_policy(τ_i, nr_jobs)
32: $\quad\quad$ **end if**
33: $\quad\quad B \leftarrow B - \bar{E}_i^p$
34: $\quad\quad n \leftarrow n - 1$
35: $\quad$ **end for**
36: $\quad$ **return** B
37: **end procedure**

Once all the HI-crit tasks are admitted for scheduling, we can compute the energy budget for LO-crit tasks as in Equation 5. In this version of the algorithm, we apply a fair policy by distributing the same budget over all the tasks (line 22). In case a task τ_i does not use the entire budget assigned (lines 23-27), all the jobs are admitted. Conversely, if the budget is not sufficient to execute all the jobs (lines 27-32), then the maximum number of allowed jobs must be computed (line 28). The convolution operator must then be re-applied to get the new energy consumption estimation (lines 29-30). In both cases, the overall LO-crit budget is decremented by the WCEC estimation of the task, $\bar{E}_i^p$ (line 33), thus preserving the unused energy for other tasks. It is worth mentioning that, as it requires the exploration of a large number of possible convolutions to compute the random variable in Equation 3, the function $max\_jobs(\tau_i, B_i)$ is computationally intensive. In particular, since state-of-the-art convolution algorithms have a complexity of $O(n \log(n))$ and the exploration is performed at most for n times, then the overall worst-case complexity of the Algorithm 1 for n tasks is upper-bounded by $O(n \cdot M \cdot \log(M))$, where M is the total number of jobs. Once the number of jobs to scheduled is computed, a proper selection policy must be applied to get the $S_{i,j}$ assignment (line 31). This policy depends on the specific scenario and application requirements. A couple of possible trivial approaches are: (1) to select only the first *nr_jobs* tasks or (2) to generate a uniformly

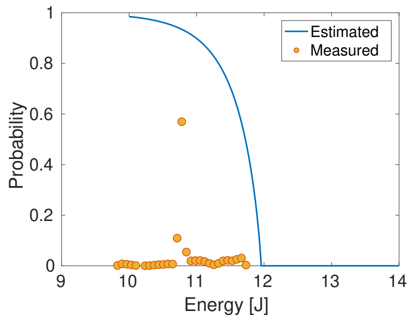


Fig. 2: The WCEC estimation validation. The solid line represents our worst-case model, while the dots indicate the frequency of the energy consumption observed probability.

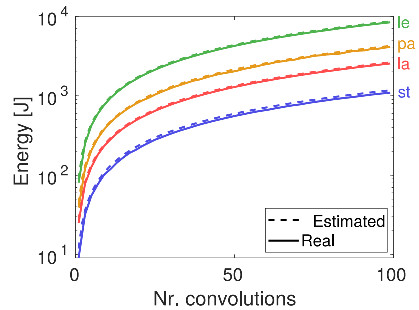


Fig. 3: The energy estimated and measured varying the number of convolutions for the four benchmarks considered.

distributed assignment. In our experimental evaluation we applied the latter approach. Finally, $B^{NULL} = B(T^*)$ is the remaining not yet assigned energy. Sorting the tasks by energy consumption (line 19) guarantees that no dropped job exists with WCEC lower than $B(T^*)$, thus the algorithm is *optimal*.

V. Experimental validation

For the experimental validation, we performed two classes of tests. The first one aimed at verifying (1) the upper-bound of the energy consumption estimation of the single job execution of each task, and (2) how the convolution operator helps in carrying out tight but still safe estimations. In the second one instead, we evaluated the job admission algorithm, checking its ability to guarantee the energy budget constraint, while keeping a tight estimation of the energy consumption.

Experimental setup. The hardware platform selected is an Odroid XU-3 equipped with a `big.LITTLE` CPU, featuring 4 Cortex-A7 and 4 Cortex-A15 cores. The selection of such a complex architecture has been made to show the validity and effectiveness of probabilistic approaches in the black-box estimation of energy consumption. This platform would be really difficult to be analyzed with traditional model-based approaches. The tasks under analysis were pinned onto the `big` cores, while the scheduler and the energy measuring task onto the `LITTLE` ones. The DVFS has been disabled forcing the frequency of big cores to be 1.8 GHz. This prevents the unwanted thermal throttling effect that would also the reproducibility of the experiments. The pWCEC estimation is theoretically agnostic w.r.t. the presence of a DVFS mechanism. However, further dedicated studies are required to assess the validity of the i.i.d. EVT condition when DVFS is enabled. The energy consumption we refer to in this Section is measured for the `big` cores only, where the tasks under analysis were running, exploiting on-chip power sensors.

The considered workload was made up of 4 multi-threaded applications from the RODINIA benchmark suite [18] (`lavaMD (la)`, `streamcluster (st)`, `leukocyte (le)`, and `particlefilter (pa)`); characterized by the same order of magnitude of execution times. This for the readability of the data plotting. The Linux operating system was equipped with the PREEMPT_RT [19] patch, in order to maximize the software determinism and increase the experiment reproducibility.

Energy estimation upper bounding. The first set of experiments focused on the estimation of the WCEC and on the verification of the performance of the convolution operation from Equation 3. First, we ran 500 jobs of each task to get the $X_1, X_2, ..., X_n$ measures necessary to perform the EVT estimation. The number of jobs used in the measurement phase affects the reliability of the final distribution [20], however, its characterization has been left as future work. The i.i.d. hypothesis of the samples has been verified by using the Ljung-Box statistical test. Then, the distribution estimation was performed. For instance, the pWCEC of the `st` benchmark is the distribution $GEV(11.596025, 0.425034, -1.178425)$, with its cumulative distribution function depicted in Figure 2. By setting failure probability value to $p = 10^{-9}$, the corresponding $WCEC$ is $11.9567J$. In other words, the probability of observing a job consuming more than $11.9567J$ is 10^{-9} or lower. To empirically check the quality of our estimation, we acquired other 10 000 samples of `st` job energy consumption. The observed probability has been computed from these samples and depicted in Figure 2. The GEV distribution safely over-estimates the real energy consumption, as expected by the theory.

Once we estimated the *pWCEC* for each task, we applied the convolution operator to the interval $j = [1; 100]$ to estimate the energy consumption of the sequences of jobs of size $1, 2, ..., 100$, considering a violation probability $p = 10^{-9}$. In a real scenario, the violation probability depends on the criticality of the application. This estimation is shown in Figure 3 (dashed lines) with respect to the number of jobs j. We verified the results by running 100 jobs of each task and by measuring the cumulative energy consumption. The real energy consumption is depicted in Figure 3 with solid lines. As we can see, the convolution operator produced tight estimations (the maximum overestimation value is 7%) and no underestimations. By running the same experiment, while using the mean energy value instead of the worst-case provided by EVT, we found that 24% of the cumulative scenarios of `pa` were underestimated and thus unsafe.

Job scheduling and energy budget. From the previous results on the WCEC per-job estimations, we used the proposed job admission/scheduling algorithm in three different scenarios. The benchmarks `la` and `pa` have been considered as HI-crit tasks, while `le` and `st` represented two LO-crit tasks. The scheduling periods have been configured as follows: $T_{\texttt{la}} = 30s$, $T_{\texttt{pa}} = 100s$, $T_{\texttt{le}} = 100s$, $T_{\texttt{st}} = 60s$. The four tasks have been scheduled, considering the following three scenarios and a high level of confidence ($p = 10^{-9}$):

Scenario	**Energy budget** B^*	**Survival period** T^*
(1)	$100\,000\,J$	$10\,h = 36\,000\,s$
(2)	$50\,000\,J$	$7\,h = 25\,200\,s$
(3)	$8\,000\,J$	$1\,h = 3\,600\,s$

The proposed algorithm generates HLSs with the following number and percentage of admitted jobs:

Scenario	`la` **(HI)**	`pa` **(HI)**	`le` **(LO)**	`st` **(LO)**
(1)	1200 (100%)	360 (100%)	290 (81%)	600 (100%)
(2)	840 (100%)	252 (100%)	15 (6%)	110 (26%)
(3)	120 (100%)	36 (100%)	7 (19.4%)	52 (86.7%)

As expected, the HI-crit jobs are all executed, while some LO-crit jobs are rejected to meet the energy budget requirements. The difference between the percentage of allocated jobs between `le` and `st` is due to the energy-fair scheduling: the `le` energy consumption is much higher than `st`.

These three scenarios have been executed using the computed HLS and the following energy results have been observed:

Scen.	**Theoretical Energy**	**Real Energy**	**Overestim.**
(1)	$99\,982.3\,J$	$96\,858.9\,J$	$3.23\,\%$
(2)	$49\,988.0\,J$	$48\,206.8\,J$	$3.70\,\%$
(3)	$7\,996.3\,J$	$7\,739.6\,J$	$3.32\,\%$

In the previous table the *theoretical energy* is the energy budget subtracted by the unused energy, i.e. $B^* - B(T^*)$, while *real energy* is the energy measured during the whole HLS execution. The proposed approach, as expected by the theoretical guarantees, has never under-estimated the real energy in the three considered scenarios. At the same time, the estimation was maintained very tight, showing an overestimation in the $3-4\%$ range compared to $20-30\%$ of the previously cited state-of-the-art tools.

VI. Conclusions

Accurate estimations of tasks energy consumption usually require detailed information about the target hardware and the workload. These are not always available, making it hard to provide any guarantee for energy-constrained systems. In this work, we proposed a probabilistic measurement-based approach to address the problem of determine the Worst-Case Energy Consumption (WCEC), in a platform-agnostic manner. The theoretical foundations have been presented, together with a job admission algorithm exploiting the WCEC analysis. The validation of the proposed approach is based on real energy measures collected from a real hardware platform running multi-thread benchmarks. Several possible future works may arise from this paper, including further exploitation in more complex resource management algorithms [21], the study of DVFS effects on the estimations reliability and the extension of the algorithm to multiple criticality scenarios.

References

[1] W. Fornaciari, G. Agosta, and D. e. a. Atienza, "Reliable power and time-constraints-aware predictive management of heterogeneous exascale systems," in *18th Int. Conf. on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*. ACM, 2018, pp. 187–194, doi:10.1145/3229631.3239368.

[2] D. Zoni, L. Cremona, and W. Fornaciari, "All-digital energy-constrained controller for general-purpose accelerators and cpus," *IEEE Embedded Systems Letters*, 2019, doi:10.1109/LES.2019.2914136.

[3] H. Hanson, S. W. Keckler, S. Ghiasi, K. Rajamani, F. Rawson, and J. Rubio, "Thermal response to dvfs: analysis with an intel pentium m," in *Proceedings of the 2007 international symposium on Low power electronics and design (ISLPED '07)*, Aug 2007, pp. 219–224.

[4] M. Malewski, D. M. J. Cowell, and S. Freear, "Review of battery powered embedded systems design for mission-critical low-power applications," *International Journal of Electronics*, vol. 105, no. 6, pp. 893–909, 2018.

[5] S. Asyaban, M. Kargahi, L. Thiele, and M. Mohaqeqi, "Analysis and scheduling of a battery-less mixed-criticality system with energy uncertainty," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 1, pp. 23:1–23:26, Oct. 2016.

[6] A. Burns and R. Davis, "Mixed criticality systems-a review," 2013–2018. [Online]. Available: https://www-users.cs.york.ac.uk/burns/review.pdf

[7] R. Jayaseelan, T. Mitra, and X. Li, "Estimating the worst-case energy consumption of embedded software," in *12th IEEE Real-Time and Embedded Technology and Applications Symp. (RTAS)*, 2006, pp. 81–90.

[8] P. Wägemann, T. Distler, T. Hnig, H. Janker, R. Kapitza, and W. Schrder-Preikschat, "Worst-case energy consumption analysis for energy-constrained embedded systems," in *2015 27th Euromicro Conference on Real-Time Systems*, July 2015, pp. 105–114.

[9] P. Wägemann, C. Dietrich, T. Distler, P. Ulbrich, and W. Schröder-Preikschat, "Whole-System Worst-Case Energy-Consumption Analysis for Energy-Constrained Real-Time Systems," in *30th Euromicro Conference on Real-Time Systems (ECRTS 2018)*, vol. 106, 2018, pp. 1–25.

[10] J. Pallister, S. Kerrison, J. Morse, and K. Eder, "Data dependent energy modeling for worst case energy consumption analysis," in *Proceedings of the 20th International Workshop on Software and Compilers for Embedded Systems*. New York, NY, USA: ACM, 2017, pp. 51–59.

[11] J. Morse, S. Kerrison, and K. Eder, "On the limitations of analyzing worst-case dynamic energy of processing," *ACM Trans. Embed. Comput. Syst.*, vol. 17, no. 3, pp. 59:1–59:22, Feb. 2018.

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

[13] M. Völp, M. Hähnel, and A. Lackorzynski, "Has energy surpassed timeliness? scheduling energy-constrained mixed-criticality systems," in *IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, April 2014, pp. 275–284.

[14] B. Emmanuel, "Microcontroller-based intelligent power management system (ipdms) for satellite application," *ARPN Journal of Engineering and Applied Sciences*, vol. 7, no. 3, pp. 377–384, 2012.

[15] B. Gnedenko, "Sur la distribution limite du terme maximum d'une serie aleatoire," *Annals of Mathematics*, vol. 44, no. 3, pp. 423–453, 1943.

[16] D. McFadden, "Modeling the choice of residential location," *Transportation Research Record*, no. 673, 1978.

[17] F. Reghenzani, G. Massari, and W. Fornaciari, "The misconception of exponential tail upper-bounding in probabilistic real-time," *IEEE Embedded Systems Letters*, pp. 1–1, 2018, doi:10.1109/LES.2018.2889114.

[18] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, S.-H. Lee, and K. Skadron, "Rodinia: A benchmark suite for heterogeneous computing," in *Int. Symp. on Workload Characterization*. IEEE, 2009, pp. 44–54.

[19] F. Reghenzani, G. Massari, and W. Fornaciari, "The real-time linux kernel: A survey on PREEMPT_RT," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 18:1–18:36, Feb. 2019, doi:10.1145/3297714.

[20] F. Reghenzani, G. Massari, W. Fornaciari, and A. Galimberti, "Probabilistic-WCET reliability: On the experimental validation of evt hypotheses," in *Int. Conf. on Omni-Layer Intelligent Systems*, ser. COINS. ACM, 2019, pp. 229–234, doi:10.1145/3312614.3312660.

[21] P. Bellasi, G. Massari, and W. Fornaciari, "Effective runtime resource management using linux control groups with the barbequertrm framework," *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 2, pp. 39:1–39:17, Mar. 2015, doi:10.1145/2658990.