Work-In-Progress: Precise Scheduling of Mixed-Criticality Tasks by Varying Processor Speed

Sai Sruti*, Ashikahmed Bhuiyan†, Zhishan Guo†
*Department of Computer Science, Missouri University of Science and Technology
†Department of Electric and Computer Engineering, University of Central Florida
*sxq6@mst.edu,†ashik@knights.ucf.edu,†zhishan.guo@ucf.edu

***Abstract*—The traditional mixed-criticality (MC) model does not allow less critical tasks to execute during an event of the error and exception. Recently, the imprecise MC (IMC) model has been proposed where, even for exceptional events, less critical tasks also receive some amount of (degraded) service, e.g., a task overruns its execution demand. In this work, we present our ongoing effort to extend the IMC model to the precise scheduling of tasks and integrate with the dynamic voltage and frequency scaling (DVFS) scheme to enable energy minimization. Precise scheduling of MC systems is highly challenging because of its requirement to simultaneously guarantee the timing correctness of all tasks under both pessimistic and less pessimistic assumptions. We propose an utilization-based schedulability test and sufficient schedulability conditions for such systems under earliest deadline first with virtual deadline (EDF-VD) scheduling policy. For this unified model, we present a quantitative study in the forms of speedup bound and approximation ratio. Finally, both theoretical and experimental analysis will be conducted to prove the correctness of our algorithm and to demonstrate its effectiveness.**

***Index Terms*—Mixed-criticality scheduling, DVFS, Speedup bound, Approximation ratio.**

I. Introduction

The real-time systems community has extensively studied the MC workload model. In the MC model, components with different criticality levels share a common platform to improve resource utilization [1]. There have been multiple extensions of this model analyzing the aspects of scheduling and schedulability conditions under various platforms. These scheduling strategies are centered around *guaranteeing resources to all the tasks under less pessimistic behaviors of the system* and *protecting* HI*-criticality tasks under more pessimistic behaviors, i.e., in the event of a task overrun.*

Existing work. State of the art literature focuses mainly on the real-time facet of dual-criticality MC systems. In such a platform, during normal conditions, the system starts in the LO-criticality mode where all tasks execute up to their normal execution requirement. During error and exception, the system switches to the HI-criticality mode and the more critical tasks (i.e., HI-criticality task) may require an extended execution time and demand more resources. If the available resources are inadequate to serve the extra computational requirements of the HI-criticality tasks, the less critical (i.e., LO-criticality) tasks will be sacrificed to free up resources [2]. Recently, this model has received criticism as it may not ensure *runtime robustness* which is a correctness requirement for an MC system [3]. As a result, the IMC model has received the attention which allows *graceful degradation* of LO-criticality tasks in HI-criticality mode [2], [4]. It embraces the concept of imprecise computing in which, upon mode-switch, each LO-criticality task can execute with inaccuracy in computing. It results in relatively short worst-case execution time (WCET), thus saving resources for more critical tasks.

Varying-speed processor: context and motivation. Increasing computational demand and the aim of reducing power bills and increasing battery life make it crucial to consider energy minimization technique in battery operated embedded platform. To achieve this goal, a significant amount of research has focused on intentionally varying the frequency of the processor. Changing frequency helps to minimize energy or utilize platform resources efficiently for various task models [5]–[11]. State of the art processors is equipped with the capability of dynamic voltage and frequency scaling (DVFS), where processor frequency is adjusted at runtime to reduce energy consumption. Huang et al. [7] proposed the integration of DVFS technique with the EDF-VD [12] scheduling scheme for dual-criticality systems to enable energy minimization. They established that when the system enters HI-criticality mode, increasing processor speed can minimize the expected energy consumption. Some recent works [6], [7], [9] have considered a stringent model where all the LO-criticality tasks are dropped upon an LO to HI mode switch. As mentioned earlier, such a model is criticized because of discarding all LO-criticality tasks (in HI-criticality mode). It can result in failures in timing assumptions for HI-criticality tasks [13], [14]. Burns and Baruah [2] exploit the elastic task model [15], where LO-criticality tasks continue to execute with extended time-periods, which generates accurate but delayed execution results.

To ensure sufficient safety and performance features, the IMC model has received attention [2], [4], [16], where each LO-criticality task receive some amount of (degraded) service after the system mode switch. However, Ernst and Di Natale [14] argues that the period and priority of a task are functional requirements and cannot be altered easily, and providing degraded services to the LO-criticality tasks can result in performance or service loss. Pathan et al. [17] observed that in case of utilization slack during execution of HI-criticality tasks in HI-criticality mode, all the LO-criticality tasks need not be penalized with degraded service. In this

2576-3172/18/$31.00 ©2018 IEEE
DOI 10.1109/RTSS.2018.00033

model, implicit-deadline IMC sporadic tasks were scheduled, in which some (if not all) LO-criticality tasks were provided full service during the HI-criticality behavior as well.

Contribution of this work. In this work, we aim to minimize energy consumption in platforms supporting MC applications. We aim to integrate the precision model in [17] to guarantee precise computing to all LO-criticality tasks in HI-criticality mode (which is to date unexplored) and the varying-speed model in [7] where we introduce an energy conserving speed or optimal minimum speed for the processor in LO-criticality mode. We adopt an off-line DVFS scheme to reduce energy utilization in the LO-criticality mode by choosing a *minimum speed* (≤ 1) for the processor while protecting MC schedulability of the framework.

(i) We combine precise scheduling of the LO-criticality tasks on varying-speed processors.

(ii) Conditions to derive the minimum speed for the processor to execute in LO-criticality mode (while correctly scheduling all the tasks in each mode of operation) are presented.

(iii) We propose a sufficient test for our precise-energy conserving model under EDF-VD. The reason for choosing EDF-VD lies in the better performance in both schedulability and run-time efficiency for scheduling MC task systems. Our ongoing effort is to prove a quantitative *speedup bound* on the worst-case performance of EDF-VD.

(iv) We derive a relationship between the per-mode utilizations of the input task and the *approximation ratio* (it compares the minimum possible degraded processor speed without speeding up upon the mode switch) of our algorithm.

II. Background and System Model

In this work, we consider scheduling of an implicit-deadline sporadic task set $\tau = \{\tau_1, \tau_2, \ldots, \tau_n\}$ on a preemptive uniprocessor platform. Each task $\tau_i \in \tau$ may generate an infinite number of MC jobs. Without loss of generality, we assume that all tasks in τ start at time 0.

MC instance. In this paper, we restrict our attention to dual-criticality MC systems where each task has two criticality levels, $\chi_i \in \{\text{LO}, \text{HI}\}$. Each task $\tau_i \in \tau$ is characterized by 5-tuple = $\{T_i, D_i, \chi_i, C_i^{\text{LO}}, C_i^{\text{HI}}\}$, where T_i represents the minimum inter-arrival time between any two consecutive job releases (by the same task), $C_i^{\text{LO}}, C_i^{\text{HI}} \in \mathbb{R}_+$ are the WCET estimations for LO and HI-criticality task respectively where $0 < C_i^{\text{LO}} \leq C_i^{\text{HI}} \leq T_i$. D_i is the deadline, and $\chi_i \in \{\text{LO}, \text{HI}\}$ represents the criticality level. The per-mode utilizations of each task τ_i and the whole task set τ are determined as follows:

$$\forall \tau_i \in \tau_{\text{LO}}, u_i^{\text{LO}} = \frac{C_i^{\text{LO}}}{T_i},$$
$$\forall \tau_i \in \tau_{\text{HI}}, u_i^{\text{HI}} = \frac{C_i^{\text{HI}}}{T_i}.$$

The total utilization for each mode of operation is represented as follows:

- For all LO-criticality tasks, the utilization is represented as: $U_{\text{LO}}^{\text{LO}} = U_{\text{LO}}^{\text{HI}} = \sum_{\tau_i \in \tau_{\text{LO}}} u_i^{\text{LO}}$. $U_{\text{LO}}^{\text{LO}}$ and $U_{\text{LO}}^{\text{HI}}$ are the same because we do not degrade services for LO-criticality tasks in HI-criticality mode.
- For all HI-criticality tasks, the utilization is represented as: $U_{\text{HI}}^{\text{LO}} = \sum_{\tau_i \in \tau_{\text{HI}}} u_i^{\text{LO}}$, $U_{\text{HI}}^{\text{HI}} = \sum_{\tau_i \in \tau_{\text{HI}}} u_i^{\text{HI}}$.

In this work, we consider implicit deadline task, so $T_i = D_i$.

Varying-speed processor. The processor frequency can be adjusted at runtime (o conserve energy) if they support the DVFS scheme [7]. We examine off-line DVFS to diminish energy utilization in the LO-criticality mode by choosing a minimum speed for the processor while protecting the MC correctness. The processor is characterized by a normal speed s (without loss of generality, $s \leftarrow 1$) and an energy-conserving speed ρ ($\rho \leq 1$). During LO-criticality mode, the processor is assumed to exhibit *energy conserving behavior* where its speed remains ρ. If the system switches to HI-criticality mode, the processor exhibits *normal behavior* where the speed of the processor is maximized ($s \leftarrow 1$). In this work, for any task τ_i, both the LO and HI-criticality WCET are determined with the assumption that it is executed on a maximum speed processor. We also consider a *linear* (i.e., half speed will lead to double execution time) relationship between the processor speed and the WCET. So, a task with execution requirement of C that is executing with an energy-conserving speed ρ will require a C/ρ amount of time to finish execution.

MC correctness. We define mode based *correctness* of the system as follows:

- In the LO-criticality mode, the processor is down-scaled by the energy-conserving speed ρ. All jobs receive up to their LO-criticality WCET and meet their deadlines.
- In the HI-criticality mode, the processor speed increases to 1, where all jobs may receive computation time up to their HI- criticality WCET and meet their deadlines.

Problem Statement. It is widely accepted that exhibiting HI-criticality behavior for any task (not signaling completion within its LO-criticality WCET) is rare. This fact motivates us to pay more attention to energy conservation when the system is executing LO-criticality behavior. In this paper, considering the linear relationship between the processor speed and the execution time, *we seek to reduce energy utilization in the* LO*-criticality mode by minimizing the energy-conserving speed ρ for the processor while protecting MC correctness of the system.*

III. Ongoing Work

In Subsection III-A, we briefly describe our on-going efforts in modifying the classic EDF-VD algorithm for the MC task system and provide a glimpse of the proof of its correctness in both execution modes. In this approach, in the LO-criticality mode, a *virtual deadline* is assigned to all HI-criticality tasks. Virtual deadline is determined by multiplying a *minimum scaling factor* with the actual deadline of a task. This scaling factor is the same for all the tasks in the system and calculated in such a way so that all the tasks can be successfully scheduled by EDF-VD and preserve the MC-correctness. If any of the HI-criticality tasks fails to finish completion within its virtual deadline, a system mode switch (from LO to HI) takes place, and all the HI-criticality tasks continue executing

according to their actual deadlines. Per job priority will be determined by the actual deadlines once the system enters HI-criticality mode. Finally in Subsection III-B, an indication of how we evaluate the effectiveness of our algorithm based on speedup bound, and approximation ratio is provided.

A. *EDF-VD for Precise Energy-Conserving Model and the Correctness Proof*

We consider the DVFS technique to conserve energy in LO-criticality mode by slowing down the processor to speed ρ. The modified EDF-VD algorithm determines if the task-set τ is schedulable or not. If YES, then it assigns a virtual deadline $\widehat{T}_i$ for all HI-criticality tasks in τ. We compute $\widehat{T}_i$ values as $\widehat{T}_i \leftarrow xT_i$, where x is the scaling factor and assigned as $x \leftarrow U_{\text{HI}}^{\text{LO}}/(\rho - U_{\text{LO}}^{\text{LO}})$ (refer to Equation (1)). Note that, in the traditional MC model, when the system executes in HI-criticality mode, only the HI-criticality tasks can execute up to their HI-criticality WCET and all LO-criticality tasks are discarded. Contradictory to the traditional model, instead of discarding all LO-criticality tasks (in HI-criticality mode), our model schedules both LO- and HI-criticality tasks with their HI-criticality WCET at full speed ($s \leftarrow 1$).

Correctness in LO-Criticality Mode. According to the EDF-VD, the virtual deadlines of all the HI-criticality tasks (in LO-criticality mode) are determined before runtime. Scaling down the deadline of the HI-criticality tasks indicates an increase in their utilization. If all the jobs finish execution within their LO-criticality WCET, the utilization bound of EDF for implicit-deadline tasks is equal to processor capacity [18]. We can, therefore, conclude that:

$$U_{\text{LO}}^{\text{LO}} + \frac{U_{\text{HI}}^{\text{LO}}}{x} \leq \rho \implies x \geq \frac{U_{\text{HI}}^{\text{LO}}}{\rho - U_{\text{LO}}^{\text{LO}}} \tag{1}$$

So this lower bound of x guarantees that EDF-VD correctly schedules all the tasks in LO-criticality mode.

Correctness in HI-Criticality Mode. Similar to the classical MC model, if a job does not signal completion after executing C_i^L units within $\widehat{T}_i$, the system enters the HI-criticality mode. In that case, we perform the following steps:

- for the active HI-criticality tasks, re-assignment of the deadline to T_i from $\widehat{T}_i$ (where, $\widehat{T}_i \leftarrow xT_i$).
- continue to execute the LO-criticality tasks.
- the speed of the processor increases to 1.

We assume a change of speed of the entire system even if a single task fails to finish completion by its LO-criticality WCET. This assumption is valid because failure to finish a task within its LO-criticality WCET triggers a system-wide mode switch. However, quite pessimistic tools are used to measure the HI-criticality WCETs, and during runtime, it is unlikely that the LO-criticality WCETs will be exceeded [19].

So, the utilization of a HI-criticality task is upper bounded by $C_i^{\text{HI}}/(1-x)T_i$ after a mode switch. Summing over all HI- and LO-criticality tasks according to the fact that EDF has a utilization bound equal to the processor capacity (in our case it is 1), we conclude that:

$$\sum_{\tau_i \in \tau_{\text{LO}}} \frac{C_i^{\text{LO}}}{T_i} + \sum_{\tau_i \in \tau_{\text{HI}}} \frac{C_i^{\text{HI}}}{(1-x)T_i} \leq 1$$
$$\implies U_{\text{LO}}^{\text{LO}} + \frac{U_{\text{HI}}^{\text{HI}}}{(1-x)} \leq 1$$

We can calculate the upper bound of x as follow:

$$x \leq \frac{1 - (U_{\text{HI}}^{\text{HI}} + U_{\text{LO}}^{\text{LO}})}{(1 - U_{\text{LO}}^{\text{LO}})} \tag{2}$$

This upper bound of x provides a sufficient condition for guaranteeing that EDF-VD correctly schedules all the assignments in HI-criticality mode. With the help of Equation (1) and (2) we can prove that, if τ satisfies

$$U_{\text{LO}}^{\text{LO}} + min\left(U_{\text{HI}}^{\text{HI}}, \frac{U_{\text{HI}}^{\text{LO}}}{\left(1 - \frac{U_{\text{HI}}^{\text{HI}}}{1 - U_{\text{LO}}^{\text{LO}}}\right)}\right) \leq \rho, \tag{3}$$

then it is schedulable by EDF-VD. Finally, it can be shown that if the following condition fulfilled:

$$U_{\text{LO}}^{\text{LO}} + \frac{U_{\text{HI}}^{\text{LO}}(1 - U_{\text{LO}}^{\text{LO}})}{1 - (U_{\text{HI}}^{\text{HI}} + U_{\text{LO}}^{\text{LO}})} \leq 1,$$

then the value of ρ (for the task set to be schedulable by EDF-VD) is lower bounded by:

$$min\left(U_{\text{LO}}^{\text{LO}} + U_{\text{HI}}^{\text{HI}}\ ,\ U_{\text{LO}}^{\text{LO}} + \frac{U_{\text{HI}}^{\text{LO}}(1 - U_{\text{LO}}^{\text{LO}})}{1 - (U_{\text{HI}}^{\text{HI}} + U_{\text{LO}}^{\text{LO}})}\right) \tag{4}$$

We leave the detailed proof as future work.

B. *Speedup Bound Analysis*

Speedup bound is a widely used metric to characterize the worst-case performance of MC scheduling algorithms. The closer the speedup bound is to 1 (which means the scheduler is near to the *optimality*), the better.

Definition 1. *[20] An algorithm $\mathcal{A}$ has a speedup bound of $\mathcal{S}$ if a task set τ that can be scheduled upon a processor with energy-conserving speed ρ and normal speed 1 by any hypothetical clairvoyant scheduling algorithm, then it can be correctly scheduled upon a processor with energy-conserving speed ρs and with normal speed $\mathcal{S}$ by algorithm $\mathcal{A}$.*

Based on the definition of speedup bound, we can prove the following conjecture:

Conjecture 1. *For the optimization problem described in Section II, our proposed modified EDF-VD has a speedup bound no larger than 2.*

Let us consider a processor has to be b times faster to schedule the task set τ using EDF-VD correctly. Then, τ should necessarily satisfy:

$$max\left(\frac{U_{\text{LO}}^{\text{LO}}}{\rho} + \frac{U_{\text{HI}}^{\text{LO}}}{\rho}, U_{\text{LO}}^{\text{LO}} + U_{\text{HI}}^{\text{HI}}\right) \leq b, \tag{5}$$

since its LO-criticality utilization ($U_{\text{LO}}^{\text{LO}} + U_{\text{HI}}^{\text{LO}}$) must be $\leq b\rho$ and its HI-criticality utilization ($U_{\text{LO}}^{\text{LO}} + U_{\text{HI}}^{\text{HI}}$) must be $\leq b$.

From Equation (1) and (2), we know that if there exists an x satisfying both conditions, both LO- and HI-criticality mode correctness are met and there will be no deadline miss. Using equation (1), (2) and (5) it is possible to derive the following relationship between LO-criticality mode utilization and the energy conserving speed:

$$U_{\text{LO}}^{\text{LO}} \leq \frac{\rho}{2} + \frac{1}{2},$$

which validates the claim proposed in Conjecture 1.

C. *Approximation Ratio*

An algorithm $\mathcal{A}$ has an approximation ratio of $\alpha \geq 1$ if and only if some non-clairvoyant on-line algorithm ensure MC correctness of τ with full speed of 1 and energy conserving speed of ρ, then $\mathcal{A}$ guarantees MC correctness to τ with full speed of 1 and energy conserving speed of $\alpha\rho$.

Conjecture 2. *For this considered model, modified EDF-VD has an approximation ratio no larger than*

$$1 + \frac{U_{\text{HI}}^{\text{LO}}(1 - U_{\text{LO}}^{\text{LO}})}{U_{\text{LO}}^{\text{LO}}(1 - (U_{\text{HI}}^{\text{HI}} + U_{\text{LO}}^{\text{LO}}))} \tag{6}$$

τ is MC-schedulable for an on-line algorithm $\mathcal{A}$, if $\alpha\rho \leq 1$. At first glance, it is evident that α is upper bounded by $1/\rho$ which is pessimistic. To generate a more viable upper bound for α (which is heavily dependent on ρ), we consider the following conditions (from Equation (3)):

Case 1: $\rho \geq U_{\text{LO}}^{\text{LO}} + min\big(U_{\text{HI}}^{\text{HI}}, \frac{U_{\text{HI}}^{\text{LO}}(1 - U_{\text{LO}}^{\text{LO}})}{1 - (U_{\text{HI}}^{\text{HI}} + U_{\text{LO}}^{\text{LO}})}\big)$.

Case 2: $\rho \leq U_{\text{LO}}^{\text{LO}} + min\big(U_{\text{HI}}^{\text{HI}}, \frac{U_{\text{HI}}^{\text{LO}}(1 - U_{\text{LO}}^{\text{LO}})}{1 - (U_{\text{HI}}^{\text{HI}} + U_{\text{LO}}^{\text{LO}})}\big)$.

Combining these conditions with Equation (4), we can derive the upper bound of α and leave it as future work.

IV. Conclusion and Future Work

The conventional MC model penalizes all the LO-criticality tasks in HI-criticality mode. Recent works overcome this setback partially (if not entirely) by accommodating LO-criticality tasks even under pessimistic behaviors. In this work, we propose an overview of an integrated model combining precise scheduling of LO-criticality tasks on energy-conserving platforms that adopt the DVFS strategy. In the future, we will provide a sufficient test for this unified model under EDF-VD scheduling algorithm. It will be evaluated theoretically with sound proofs and schedulability experiments on randomly generated workloads. We will provide results on calculating both the speed-up bound and approximation ratio to satisfy real-time requirements in a situation of overrun.

As future work, we will show that EDF-VD is an optimal non-clairvoyant algorithm (from the perspective of speedup bound). That is, no non-clairvoyant algorithm can meet all deadlines on a processor that is less than two times as fast as the processor available to the optimal clairvoyant algorithm. Besides, we seek to derive and prove a tighter bound for the approximation ratio (if one exists), work on counter-examples to show the minimum possible bound, and also explore schedulability conditions under a task-wise mode-switch, contrary to the system-wise mode switch adopted in this work. In this work, we restrict our attention on integrating DVFS with precise scheduling of the less critical tasks on a uniprocessor platform. It is an essential step forward towards the multi-processor platform for sequential or parallel task models. To verify the theoretical findings as well as the effectivenesses of the proposed algorithm, we plan to conduct an experimental study with onboard implementations. We expect this work will evolve many future research efforts upon completion.

References

[1] S. Vestal, "Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance," in *RTSS*. IEEE, 2007.

[2] A. Burns and S. Baruah, "Towards a more practical model for mixed criticality systems," in *Workshop on Mixed-Criticality Systems*, 2013.

[3] S. Baruah, "Mixed-criticality scheduling theory: Scope, promise, and limitations," *IEEE DESIGN AND TEST*, vol. 35, no. 2, pp. 31–37, 2018.

[4] S. Baruah, A. Burns, and Z. Guo, "Scheduling mixed-criticality systems to guarantee some service under all non-erroneous behaviors," in *ECRTS*. IEEE, 2016.

[5] S. Baruah and Z. Guo, "Mixed-criticality scheduling upon varying-speed processors," in *RTSS*. IEEE, 2013.

[6] ——, "Scheduling mixed-criticality implicit-deadline sporadic task systems upon a varying-speed processor," in *RTSS*. IEEE, 2014.

[7] P. Huang, P. Kumar, G. Giannopoulou, and L. Thiele, "Energy efficient DVFS scheduling for mixed-criticality systems," in *EMSOFT*. IEEE, 2014.

[8] A. Paolillo, P. Rodriguez, N. Veshchikov, J. Goossens, and B. Rodriguez, "Quantifying energy consumption for practical fork-join parallelism on an embedded real-time operating system," in *RTNS*. ACM, 2016.

[9] S. Narayana, P. Huang, G. Giannopoulou, L. Thiele, and R. V. Prasad, "Exploring energy saving for mixed-criticality systems on multi-cores," in *RTAS*. IEEE, 2016.

[10] A. Bhuiyan, Z. Guo, A. Saifullah, N. Guan, and H. Xiong, "Energy-efficient real-time scheduling of DAG tasks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 17, no. 5, p. 84, 2018.

[11] Z. Guo, A. Bhuiyan, A. Saifullah, N. Guan, and H. Xiong, "Energy-efficient multi-core scheduling for real-time DAG tasks," in *LIPIcs-Leibniz International Proceedings in Informatics*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[12] S. K. Baruah, V. Bonifaci, G. D'Angelo, A. Marchetti-Spaccamela, S. Van Der Ster, and L. Stougie, "Mixed-criticality scheduling of sporadic task systems," in *European Symposium on Algorithms*. Springer, 2011.

[13] A. Burns and R. I. Davis, "A survey of research into mixed criticality systems," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, p. 82, 2017.

[14] R. Ernst and M. Di Natale, "Mixed criticality systems a history of misconceptions?" *IEEE Design & Test*, vol. 33, no. 5, pp. 65–74, 2016.

[15] H. Su and D. Zhu, "An elastic mixed-criticality task model and its scheduling algorithm," in *DATE*. IEEE, 2013.

[16] D. Liu, J. Spasic, N. Guan, G. Chen, S. Liu, T. Stefanov, and W. Yi, "EDF-VD scheduling of mixed-criticality systems with degraded quality guarantees," in *RTSS*. IEEE, 2016.

[17] R. M. Pathan, "Improving the quality-of-service for scheduling mixed-criticality systems on multiprocessors," in *LIPIcs-Leibniz International Proceedings in Informatics*, vol. 76. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[18] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *Journal of the ACM (JACM)*, vol. 20, no. 1, pp. 46–61, 1973.

[19] Z. Guo, L. Santinelli, and K. Yang, "EDF schedulability analysis on mixed-criticality systems with permitted failure probability," in *RTCSA*. IEEE, 2015.

[20] B. Kalyanasundaram and K. Pruhs, "Speed is as powerful as clairvoyance," *Journal of the ACM (JACM)*, vol. 47, no. 4, pp. 617–643, 2000.