Probabilistic Schedulability Analysis for Real-time Tasks with Precedence Constraints on Partitioned Multi-core — Source link ↗

Slim Ben-Amor, Liliana Cucu-Grosjean, Mehdi Mezouak, Yves Sorel

Institutions: French Institute for Research in Computer Science and Automation

Related papers:

- Schedulability analysis of dependent probabilistic real-time tasks

- Schedulability Analysis for Communicating Tasks on a Multiprocessor System

- Uniprocessor scheduling under precedence constraints for embedded systems design

- Scheduling periodic task graphs with communication delays

- Non-preemptive scheduling algorithms and schedulability conditions for real-time systems with precedence and latency constraints

Abstract—The design of embedded systems is facing the explosion of new functionalities requiring increased computation capacities and, thus, the introduction of multi-core processors. Moreover, some functionalities may impose precedence constraints between the programs implementing them. In this paper, we consider partitioned scheduling of tasks with precedence constraints defined by multiple Directed Acyclic Graphs (DAGs). The variability of execution and communication times is taken into account by describing them with probability distributions. Our probabilistic response time analysis is validated on random generated task sets and on a PX4 drone autopilot.

Index Terms—DAGs, Multi-core, Partitioning, Fixed-priority

I. INTRODUCTION

Chip manufacturers are constantly seeking to improve hardware performance. They have incorporated several cores on the same processor to allow simultaneous processing, which offers a speedup for executing programs. In addition, software designer divide large tasks (programs) into smaller sub-tasks (threads) in order to exploit possible intra-task parallelism. This approach creates precedence constraints between different sub-tasks. Thus, a DAG task model is adopted to describe different independent programs as well as dependent threads inside them. Although in many cases, real-time systems require intensive computation resources, they do not take advantage of the parallel processing provided by multi-cores because traditional timing analysis techniques do not allow such systems to be validated. In fact, these validation processes tends to reject a design solution even if it is feasible under a significant number of execution scenarios but it is unfeasible with a worst-case timing parameters that appear rarely. This leads to a pessimistic analysis. Therefore, additional computation resources are required for validation and the system will be over-sized.

For the purpose of reducing pessimism and over-sizing, we present a probabilistic schedulability analysis that takes into consideration the variability of execution times and estimates the Deadline Miss Probability (DMP) of each task. If large values of the execution times are not frequent, then DMP may be small. Hence, the system becomes schedulable with a high confidence, which reduces the pessimism. Such an analysis could be applied on soft real-time systems to guarantee a high quality of service when the DMP is small. It could also be used on industrial systems with safety standards that require low probability of failure.

II. TASK MODEL AND NOTATIONS

We consider a real-time system of n sporadic tasks scheduled according to a partitioned preemptive fixed-priority scheduling policy on m identical cores. We denote by τ the set of n tasks $\tau_1, \tau_2, \ldots, \tau_n$ and by π the processor that has m identical cores $\pi_1, \pi_2, \ldots, \pi_m$. Each task τ_i is specified by a 3-tuple (G_i, D_i, T_i), where G_i is a DAG describing the internal structure of τ_i, D_i is its deadline and T_i the minimal inter-arrival time between two consecutive arrivals. We consider a constrained deadline i.e., $D_i \leq T_i$.

For a task τ_i, the associated DAG G_i is defined by (V_i, E_i), where $V_i = \{\tau_{i,j}\}_{1 \leq j \leq n_i}$ is a set of n_i sub-tasks of τ_i and E_i is the set of the precedence constraints between sub-tasks of task τ_i. A sub-task $\tau_{i,j}$ is characterized by its probabilistic worst-case execution time (pWCET) $\mathcal{C}_{i,j}$. We assume that the probability distributions are independent and given. Obtaining such distributions is beyond the purpose of our work.
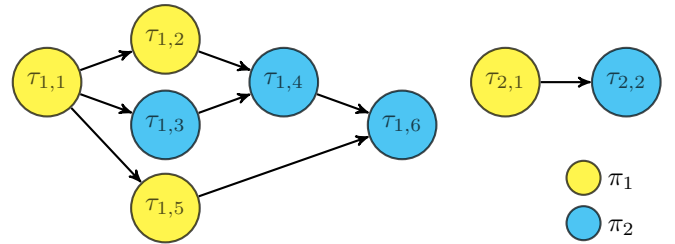
Fig. 1: Example of DAG task set with precedence constraints

Each sub-task $\tau_{i,j}$ is mapped to a core denoted $\pi(\tau_{i,j})$. We assume that this mapping is given. For instance, in Figure 1 the sub-tasks colored in the same color are assigned to the same core. We consider also that the priorities are assigned at sub-task level. We denote $hp(\tau_{i,j})$ as the set of sub-tasks $\tau_{p,q}$ with higher priority than $\tau_{i,j}$. For a sub-task $\tau_{i,j}$, we denote the set of its immediate successors by

$isucc(\tau_{i,j}) = \{\tau_{i,k} \mid \exists\ (\tau_{i,j}, \tau_{i,k}) \in E_i\}$. Moreover, other sub-tasks may be reachable from $\tau_{i,j}$ by directed paths. We denote the set of these sub-tasks by $succ(\tau_{i,j})$. We note that $isucc(\tau_{i,j}) \subseteq succ(\tau_{i,j})$. Similarly, we denote the set of immediate predecessors by $ipred(\tau_{i,j}) = \{\tau_{i,k} \mid \exists (\tau_{i,k}, \tau_{i,j}) \in E_i\}$ and by $pred(\tau_{i,j}) = \{\tau_{i,k} \mid \tau_{i,j} \in succ(\tau_{i,k})\}$.

Two sub-tasks that are not reachable with directed path one from another are called independent and they may execute in parallel on different cores. We denote by $parallel(\tau_{i,j})$ the set of sub-tasks independent of sub-task $\tau_{i,j}$. More precisely,

$$parallel(\tau_{i,j}) = \{\tau_{i,k} \mid \tau_{i,k} \in V_i \setminus \{pred(\tau_{i,j}) \cup succ(\tau_{i,j})\}\}$$

A weight $e_i(j, k)$ is associated to each precedence constraint $(\tau_{i,j}, \tau_{i,k})$. This weight accounts for communication costs between $\tau_{i,j}$ and $\tau_{i,k}$ and it is described by a probabilistic worst case communication time distribution (pWCCT).

III. Schedulability Analysis

In this section, we present our RTA for sub-tasks with precedence constraints described by DAGs with probabilistic WCET, scheduled according a given fixed-priority and partitioned policy. Our analysis is based on fixed-point response time equations inspired by Palencia results [1] that provide probability distributions for the worst-case response times (pWCRTs) of sub-tasks. We define the DMP for task τ_i as $DMP_i = P(\mathcal{R}_i > D_i)$, where $\mathcal{R}_i = \mathcal{R}_{i,sink}^{glob}$ is the probabilistic WCRT of task τ_i.

Here, we compute first the response time of the whole graph assuming no higher-priority DAG that could preempt the sub-task under study $\tau_{i,j}$ are running and we consider only sub-tasks that are predecessors of $\tau_{i,j}$. We call the resulting response time the **local response time**. Next, we define the **response time in isolation** which considers only sub-tasks from the same graph and discards the effect of higher-priority DAG tasks. Last, we compute the **global response time** by adding the effect of preemptions of higher-priority DAG tasks.

1) Local response time: For the calculation of the local response time of sub-task $\tau_{i,j}$, we sum the probabilistic execution time of $\tau_{i,j}$ and the maximum probabilistic response time over its predecessors:

$$\mathcal{R}_{i,j}^{local} = \mathcal{C}_{i,j} \otimes \max_{\tau_{i,l}} \left\{ \mathcal{R}_{i,l}^{local} \otimes e_i(l,j) \otimes \mathcal{I}_{i,l}(pred(\tau_{i,j})) \right\} \tag{1}$$

Where $\tau_{i,l} \in ipred(\tau_{i,j})$ and $\otimes$ is the convolution operator between probability distributions. $\mathcal{I}_{i,l}(pred(\tau_{i,j}))$ represent the interference caused by predecessors of $\tau_{i,j}$ on $\tau_{i,l}$ and its predecessors. We compute this interference by summing the execution time of any predecessor of $\tau_{i,j}$ that has a higher-priority, is executed on the same core and is parallel to $\tau_{i,l}$ or any of its predecessors. It is given by:

$$\mathcal{I}_{i,l}(pred(\tau_{i,j})) = \bigotimes_{\substack{\exists \tau_{i,a} \in pred(\tau_{i,l}) \cup \tau_{i,l} \\ \tau_{i,k} \in parallel(\tau_{i,a}) \\ \pi(\tau_{i,k}) = \pi(\tau_{i,a}) \\ \tau_{i,k} \in hp(\tau_{i,a})}} \mathcal{C}_{i,k}$$

2) Response time in isolation: The RTA in isolation discards only higher-priority sub-tasks from other DAG tasks. Since the local response time considers only predecessors sub-tasks, we add to this latter the sum of execution times of parallel sub-tasks with higher-priority from the same graph that are executed on the same core as the studied sub-task $\tau_{i,j}$ and that are not predecessors of $\tau_{i,j}$.

$$\mathcal{R}_{i,j}^{isol} = \mathcal{R}_{i,j}^{local} \otimes \bigotimes_{\substack{\exists \tau_{i,l} \in pred(\tau_{i,j}) \cup \tau_{i,j} \\ \tau_{i,k} \notin pred(\tau_{i,j}) \\ \tau_{i,k} \in parallel(\tau_{i,l}) \\ \pi(\tau_{i,k}) = \pi(\tau_{i,l}) \\ \tau_{i,k} \in hp(\tau_{i,l})}} \mathcal{C}_{i,k} \tag{2}$$

3) Global response time: The global response time takes into consideration all possible preemptions of higher-priority tasks. It is calculated recursively by Equation 3. We add to the response time in isolation the effect of higher-priority tasks. The iterative update of $\mathcal{R}_{i,j}^{glob}$ stops when it reaches a fixed point or when it becomes greater than the deadline.

$$\mathcal{R}_{i,j}^{glob} = \mathcal{R}_{i,j}^{isol} \otimes \bigotimes_{\substack{\exists \tau_{i,k} \in pred(\tau_{i,j}) \cup \tau_{i,j} \\ \pi(\tau_{p,q}) = \pi(\tau_{i,k}) \\ \tau_{p,q} \in hp(\tau_{i,k})}} \left\lceil \frac{\mathcal{R}_{i,j}^{glob} \otimes \mathcal{J}_{p,q}}{T_p} \right\rceil \mathcal{C}_{p,q} \tag{3}$$

Where the jitter $\mathcal{J}_{i,j} = \max_{\tau_{i,k} \in ipred(\tau_{i,j})} \left\{ \mathcal{R}_{i,k}^{glob} \right\}$

In Table I, we illustrate how response time equations work on the task set example defined in Figures 1. We assume that communication costs are equal to $1\ ms$ if related sub-tasks are mapped to different cores and $0\ ms$ otherwise. We note that the pWCRT of task τ_1 is equal to $\left(\begin{smallmatrix} 26 & 30 \\ .6 & .4 \end{smallmatrix} \right)$. However, if we use the approach adopted by Palencia et al [1], we find $R_1 = 46$ with the two possible values for $\mathcal{C}_{1,5}$. Hence, we observe that our analysis helps to reduce pessimism when estimating the WCRT of the DAG task. On the other hand, by using Fonseca et al. [2] approach, we find $R_1 = 26$ for $\mathcal{C}_{1,5} = 2$ and $R_1 = 31$ for $\mathcal{C}_{1,5} = 7$. However, our analysis is faster because [2] approach is based on an ILP problem. This latter found the result in 0.3 seconds, while our analysis require only 0.002.

TABLE I: Parameters and RTA of task set in Figures 1

Sub-task	$\mathcal{C}_{i,j}$	Prio	$\mathcal{R}_{i,j}^{local}$	$\mathcal{R}_{i,j}^{isol}$	$\mathcal{R}_{i,j}^{glob}$
$\tau_{1,1}$	$1\ ms$	3	1	1	9
$\tau_{1,2}$	$1\ ms$	4	2	2	10
$\tau_{1,3}$	$2\ ms$	6	4	4	22
$\tau_{1,4}$	$2\ ms$	7	6	6	24
$\tau_{1,5}$	$\left(\begin{smallmatrix} 2 & 7 \\ .6 & .4 \end{smallmatrix} \right)$	5	$\left(\begin{smallmatrix} 3 & 8 \\ .6 & .4 \end{smallmatrix} \right)$	$\left(\begin{smallmatrix} 4 & 9 \\ .6 & .4 \end{smallmatrix} \right)$	$\left(\begin{smallmatrix} 12 & 17 \\ .6 & .4 \end{smallmatrix} \right)$
$\tau_{1,6}$	$2\ ms$	8	$\left(\begin{smallmatrix} 8 & 12 \\ .6 & .4 \end{smallmatrix} \right)$	$\left(\begin{smallmatrix} 8 & 12 \\ .6 & .4 \end{smallmatrix} \right)$	$\left(\begin{smallmatrix} 26 & 30 \\ .6 & .4 \end{smallmatrix} \right)$
$\tau_{2,1}$	$8\ ms$	1	8	8	8
$\tau_{2,2}$	$10\ ms$	2	19	19	19

References

[1] P. Gutiérrez, J. García, and M. Harbour, "On the schedulability analysis for distributed hard real-time systems," in *RTS*, 1997.
[2] J. Fonseca, G. Nelissen, V. Nélis, and L. Pinho, "Response time analysis of sporadic DAG tasks under partitioned scheduling," in *SIES*, 2016.