Impact of job dropping on the schedulability of uniprocessor probabilistic real-time systems with variable execution times

Olivier Buffet, Liliana Cucu-Grosjean

Impact of job dropping on the schedulability of uniprocessor probabilistic real-time systems with variable execution times

Olivier Buffet & Liliana Cucu-Grosjean

Abstract

In this paper we address the problem of uniprocessor probabilistic scheduling of real-time systems with variable execution times. For these systems the tasks have an associated probability of missing the deadline, i.e., some jobs may miss their deadlines without affecting the schedulability of the system. Therefore dropping these jobs does not affect the schedulability of the system and it could increase the probability of other jobs to meet their deadline. The problem of deciding what jobs to drop is not trivial and we discuss a possible solution.

1 Introduction

Requests in real-time environment are often of a recurring nature. Such systems are typically modeled as finite collections of simple, highly repetitive activities (e.g., tasks, messages). When the different instances of those activities are generated in a very predictable manner, we deal with periodic activities. The real-time performances of periodic activities on uniprocessor, distributed or network systems have been extensively studied when all their parameters are known. For some applications the parameters can be unknown until the time instant when the activity is released, or the environment can change forcing the application to adapt. Different approaches can be considered to address these uncertainties (probabilistic approaches, agent systems, learning or game theory, etc) and this paper uses a probabilistic formulation.

2 Problem statement

We deal with the uniprocessor scheduling problem of synchronous periodic tasks with variable execution times. We consider $\tau = \{\tau_1, \tau_2, \cdots, \tau_n\}$ a set of n periodic tasks.

Each task is characterized by an exact inter-arrival time T_i, a relative deadline D_i and a probability of meeting the deadline p_i. It means that the j^{th} activation of τ_i is released at time instant $(j-1)T_i$ and must finish its execution by time instant $(j-1)T_i + D_i$. Among all (representative) activations, at least p_i of them must finish their execution by their deadline. Each activation τ_i has an associated execution time given by a discrete random variable. We denote by $\mathcal{C}_i$[1] the random variable giving the execution time of τ_i (see Equation (1)). It is assumed that the random variables giving the execution times are independent.

$$\mathcal{C}_i = \begin{pmatrix} c_k \\ P(C = c_k) \end{pmatrix}_{k \in \{1, \cdots, k_i\}} \tag{1}$$

In Equation (1), $c_k \in [c_i^{min}, c_i^{max}]$ and $k_i \in \mathbb{N}^*$ is the number of values that the random variable $\mathcal{C}_i$ has. We consider that c_i^{min}, c_i^{max} are known.

We denote a task τ_i by $(\mathcal{C}_i, T_i, D_i, p_i)$.

A *schedule* is said *feasible* if any task τ_i has the probability of missing the deadline smaller than p_i.

A schedulability analysis like [1] calculates the response time of a job for the fixed-priority scheduling problem. All jobs released within a hyperperiod contribute to the response time of the corresponding task. Some jobs may always have the obtained response time larger than the deadline, i.e., the probability of missing the deadline is 100%. Therefore we may decide to drop such job before its execution without increasing the probability of missing the deadline for the corresponding task. Nevertheless this mechanism may decrease the probability of missing the deadline for other jobs.

The problem of deciding what jobs should be dropped in order to obtain a feasible schedule is not trivial. For instance in fixed-priority scheduling if we decide to drop all jobs that have the probability of missing the deadline of 100%, then we are too pessimistic. After dropping some jobs, lower priority jobs may decrease their probability of missing the deadline and among them, those with the probability originally equal to 100%.

3 Preliminary solution

Intuitively the problem is difficult, even if no such proof exists (to our best knowledge). Therefore we believe that approaches based on systematic search algorithms may propose interesting solutions. In this case (the value of) a solution might be tested using the sum of the probabilities to miss the deadlines of all jobs within a hyperperiod.

References

[1] J. Díaz, D. Garcia, K. Kim, C. Lee, L. Bello, L. J.M., and O. Mirabella, "Stochastic analysis of periodic real-time systems," in *23rd of the IEEE Real-Time Systems Symposium (RTSS02)*, 2002, pp. 289–300.

[1] In this paper we utilise calligraphic letters to denote random variables