SCHEDULING WITH QUALITY OF SERVICE AND FAIRNESS IN WEAKLY-HARD REAL-TIME SYSTEMS

Maryline Chetto, and Audrey Marchand

Institut de Recherche en Communications et Cybernétique de Nantes / University of Nantes
UMR CNRS 6597 - 1, rue de la Noé  44321 NANTES Cedex 03, France
E-mail: maryline.chetto@univ-nantes.fr

ABSTRACT
This paper deals with dynamic scheduling in real-time systems said to be weakly-hard and which may experience overload. We assume that tasks are periodic and may miss their deadlines, occasionally, as defined by the so-called Skip-Over model. The objective of the scheduler is to provide in line a schedule which provides both Quality of Service and fairness. QoS is a metric of the ratio of periodic task instances which complete before their deadline and fairness is defined in terms of individual success balancing. The work presented here focuses on one skip-over algorithm, namely BWP. The intent of the paper is to show how to improve fairness by combining Earliest Deadline First with two strategies, namely LF (Last Failure) and MS (Minimum Success). An experimental study is reported in order to compare results from these two point of view: fairness and QoS.

1. INTRODUCTION
Real time schedule theory is mainly applied to a kind of system, wherein the temporal aspects of their behavior are parts of their requirement. For applications with Hard Real Time requirements, no deadline missed is tolerated. A Weakly-Hard-Real Time schedule theory is a conceptual framework which investigates the characteristics of a system that can tolerate certain deadline missing under a precise distribution in a finite time window. Hence, the tolerance to deadline missing is established within a window of consecutive invocations of the tasks. In this paper we investigate the problem of scheduling periodic tasks in a Weakly-Hard Real-Time system with the objective to guarantee specific temporal constraints that represent a minimal expected Quality of Service while serving the tasks with fairness.

The remainder of the paper is organized as follows: Section II presents related work on the weakly-hard-real time schedule theory. In Section III, we describe the skip-over model. Section IV describes some previous works which have inspired two strategies to perform fairness for skippable tasks. In section V, we briefly describe the main results of an experiment that show improvements in terms of fairness thanks to two dynamic scheduling strategies, namely BWP-LF and BWP-MS. Finally, section VI concludes the paper.

2. RELATED WORK
To capture the situation that deadlines missing with a permitted distribution over a finite range can be tolerated, a second parameter describing the window of time within which the number of deadlines must hold, should be specified. The work has been first done by Koren and Shasha [1], who propose an approach of description of deadline missed with deterministic distribution, skip factor. A task which has a skip factor of s will have one instance skipped out of its s consecutive instances. A skip factor at least deterministically guarantees at most one deadline missed occurring over a finite time, s consecutive instances. Further, Hamdoui and Ramanathan expand the notion of the skip factor with (m, k)-firm, to specify a task that is desired to meet deadline of m instances among its consecutive k instances [2].

3. MODEL AND METRICS

3.1 The skip-over model
In what follows, we consider the problem of scheduling periodic tasks which allow occasional deadline violations (i.e., skippable periodic tasks)

---

1 Work supported by the French research office, grant number 01 K 0742.
on a uniprocessor system. We assume that tasks can be preempted at any time and they do not have precedence constraints. A task $T_i$ is characterized by a worst-case computation time $C_i$, a period $P_i$, a relative deadline equal to its period, and a skip parameter $s_i$. That means that the distance between two consecutive skips must be at least $s_i$ periods. When $s_i$ equals to infinity, no skips are allowed and $T_i$ is a hard periodic task. Every task $T_i$ is divided into instances where each instance occurs during a single period of the task. Every instance of a task is either red or blue [1]. A red task instance must complete before its deadline whereas a blue task instance can be aborted at any time and so is optional. However, if a blue instance completes successfully, the next instance of the same task is still blue.

3.2 RTO and BWP algorithms

Koren and Shasha introduced about ten years ago the Red Tasks Only (RTO) algorithm and the Blue When Possible (BWP) algorithm. Under RTO, Red instances are scheduled as soon as possible according to Earliest Deadline First (EDF) algorithm [3], while blue ones are always rejected. BWP algorithm is an improvement of the first one since it schedules blue instances whenever their execution do not prevent the red ones from completing within their deadlines. In other words, blue instances are served in background relatively to red instances.

3.3 QoS criteria

Our interest here is not quantifying absolute QoS (defined here as the global success ratio of task instances while guaranteeing skip constraints) but measuring it in a relative manner in order to compare several scheduling strategies. Obviously here, RTO provides the worst QoS since no blue task instances is executed. We focus on BWP and show how to provide a fair scheduling with acceptable QoS. Indeed, recently, a dynamic scheduling algorithm, called RLP (Red as Late as Possible) based on a slack-stealing approach known as EDL (Earliest Deadline as Late as Possible) strategy [4] has been recently proposed in order to improve the QoS of basic Skip Over algorithms. Results are reported in [5]. Nevertheless, we want to bring to light how to improve fairness while guaranteeing a Quality of Service as good as possible.

3.4 Fairness criteria

By definition, RTO scheduling algorithm is very fair since it never attempts to schedule blue task instances. Consequently, tasks have always the same individual success ratio related to red task completions, but this is coupled with a poor global QoS (for $s_i = 2$ where one instance every two has to complete within its deadline, the global success ratio is equal to 50%). That’s why we will focus here on the behavior of the system in terms of QoS and fairness, only for the BWP scheduling strategy. The objective is to underline the lack of fairness of this strategy when blue task instances are ordered by increasing deadlines, i.e. according to Earliest Deadline (ED). And, we will denote this strategy by BWP-ED.

4. CONTRIBUTIONS

4.1 Previous works

Every hard real time system can be subject to perturbations which appear as timing failures, i.e. deadlines missing, which is unacceptable. The Deadline Mechanism enables to construct a reliable system by implementing for each task a program said primary that produces a good quality service but in an unknown and possibly infinite length of time and on the other hand a back-up program said alternate that produces an acceptable result in a fixed length of time [4]. Chetto et al have developed and evaluated a scheduling strategy proved to be optimal for the Deadline Mechanism, with the objective to maximize the global success ratio of primary tasks using a slack stealing approach, known as the Last Chance strategy. While a maximal global success ratio for primaries was attained with ED, the experiment proved the necessity to provide other strategies for scheduling primaries due to unfairness of ED. The need of designing control systems which are both robust and stable has led us to bring to light two primary scheduling strategies respectively called MF (Maximum Failure first) and MS (Minimum Success first) which have inspired their adaptation to the skip-over model as described below [6].

4.2 A way to improve fairness

From our results on the Deadline Mechanism reported previously, we propose to evaluate the two following strategies in the context of a weakly-hard real-time system:

- BWP-LF (Blue When Possible - Last Failure) algorithm schedules at each time
instant, the ready blue task whose number of successive successes from the last failure is least. The earliest deadline rule is used to break ties between blue tasks of equal priorities.

- BWP-MS (Blue When Possible - Minimum Success) algorithm schedules at each time instant, the ready blue task whose individual success ratio memorized from the initialization time is least. As for the BWP-LF case, ties are broken in favour of the task with the earliest deadline.

5. IMPLEMENTATION AND EXPERIMENT

5.1. Simulation study

The simulation context includes 50 periodic task sets, each consisting of 10 tasks with a least common multiple equal to 3360. Tasks are defined under QoS constraints with uniform $s_i = 2$. Their worst-case execution time depends on the setting of the periodic load $U_p$ and is randomly generated in order to reflect the greatest number of applications. Deadlines are equal to the periods and greater than or equal to the computation times. Simulations have been processed over 10 hyper-periods.

5.2. Comparative evaluation

Figure 1 enables us to conclude that BWP-ED is better than all other variants of BWP since Earliest Deadline is the optimal scheduling algorithm for deadline critical tasks. Consequently, if fairness is of no importance for a given application, we can consider BWP-ED as the best one with a global success ratio that attains 100% when the processor is under-loaded. We note that in overload situations, BWP-MS is the worst one, even if the gap with BWP-LF is less than 5%.

Results, not reported here show that BWP-ED leads to give more importance to some tasks and consequently does not provide a fair service. The distribution of the individual success ratios has been significantly improved by the proposed algorithms, BWP-LF and BWP-MS. Individual success ratio of tasks with short period has been increased by 10% on average. Whereas the mean distance between two individual success ratios is respectively equal to 30.88% for BWP-ED (not reported here) and 12.65% for BWP-LF, it is reduced to 1.24% for BWP-MS on the average. Similarly, as regards the maximal difference of individual success ratios, we observe a big gap between the different strategies (see Figures 2 and 3).

![Figure 1: Measurement of the QoS for BWP variants](image-url)
6. CONCLUDING REMARKS

With the emergence of lots of weakly-hard real time applications that can tolerate a certain deadline missed, the understanding real time must be improved instead of just guaranteeing deadline of each instance of a task. The real time schedule theory need to be expanded to investigate two criteria: fairness and QoS. The experimental study reported in this paper has permitted to compare results from these two point of view under the so-called skip-over model.

REFERENCES