A non-preemptive scheduling algorithm for soft real-time systems

Wenming Li, Krishna Kavi *, Robert Akl

The University of North Texas, Computer Science and Engineering, P.O. Box, 311366, Denton TX 76203, United States

Received 2 December 2005; accepted 10 April 2006
Available online 25 July 2006

Abstract

Real-time systems are often designed using preemptive scheduling and worst-case execution time estimates to guarantee the execution of high priority tasks. There is, however, an interest in exploring non-preemptive scheduling models for real-time systems, particularly for soft real-time multimedia applications. In this paper, we propose a new algorithm that uses multiple scheduling strategies for efficient non-preemptive scheduling of tasks. Our goal is to improve the success ratio of the well-known Earliest Deadline First (EDF) approach when the load on the system is very high and to improve the overall performance in both underloaded and overloaded conditions. Our approach, known as group-EDF (gEDF) is based on dynamic grouping of tasks with deadlines that are very close to each other, and using Shortest Job First (SJF) technique to schedule tasks within the group. We will present results comparing gEDF with other real-time algorithms including, EDF, Best-effort, and Guarantee, by using randomly generated tasks with varying execution times, release times, deadlines and tolerance to missing deadlines, under varying workloads. We believe that grouping tasks dynamically with similar deadlines and utilizing a secondary criteria, such as minimizing the total execution time (or other metrics such as power or resource availability) for scheduling tasks within a group, can lead to new and more efficient real-time scheduling algorithms. © 2006 Elsevier Ltd. All rights reserved.

Keywords: Soft real-time systems; Non-preemptive real-time scheduling; Earliest Deadline First (EDF); Shortest Job First (SJF); Best-effort scheduling; Group-EDF

1. Introduction

The Earliest Deadline First (EDF) algorithm is the most widely studied scheduling algorithm for real-time systems [1]. For a set of preemptive tasks (be they periodic, aperiodic, or sporadic), EDF will find a schedule if a schedule is possible [2]. The application of EDF for non-preemptive tasks is not as widely investigated. It is our contention that non-preemptive scheduling is more efficient, particularly for soft real-time applications and applications designed for multithreaded systems, than the preemptive approach since the non-preemptive model reduces the overhead needed for switching among tasks (or threads) [3,4]. EDF is optimal for sporadic
non-preemptive tasks, but EDF may not find an optimal schedule for periodic and aperiodic non-preemptive
tasks. It has been shown that scheduling periodic and aperiodic non-preemptive tasks is NP-hard [5–7]. How-
ever, non-preemptive EDF techniques have produced near optimal schedules for periodic and aperiodic tasks,
particularly when the system is lightly loaded. When the system is overloaded, however, it has been shown that
the EDF approach leads to very poor performance (low success rates) [8]. In this paper, a system load or uti-
lization is used to refer to the sum of the execution times of pending tasks as related to the time available to
complete the tasks. The poor performance of EDF is due to the fact that, as tasks that are scheduled based on
their deadlines miss their deadlines, other tasks waiting for their turn are likely to miss their deadlines also – an
outcome sometimes known as the domino effect. It should also be remembered that Worst-Case Execution
Time (WCET) estimates for tasks are used in most real-time systems. We believe that, in practice, WCET esti-
mates are very conservative, and more aggressive scheduling schemes based on average execution times for
soft real-time systems using either EDF or hybrid algorithms can lead to higher performance.

While investigating scheduling algorithms, we have analyzed a variation of EDF that can improve success
ratios (that is, the number of tasks that have been successfully scheduled to meet their deadlines), particularly
in overloaded conditions. The new algorithm can also decrease the average response time for tasks. We call
our algorithm group-EDF, or gEDF, where the tasks with “similar” deadlines are grouped together (i.e.,
deadlines that are very close to one another), and the Shortest Job First (SJF) algorithm is used for scheduling
tasks within a group. It should be noted that our approach is different from adaptive schemes that switch
between different scheduling strategies based on system load; gEDF is used in overloaded as well as under-
loaded conditions. The computational complexity of gEDF is the same as that of EDF. In this paper, we will
evaluate the performance of gEDF using randomly generated tasks with varying execution times, release
times, deadlines and tolerance to missing deadlines, under varying loads.

We believe that gEDF is particularly useful for soft real-time systems as well as applications known as
“anytime algorithms” and “approximate algorithms,” where applications generate more accurate results or
rewards with increased execution times [9,10]. Examples of such applications include search algorithms, neu-
ral-net based learning in AI, FFT and block-recursive filters used for audio and image processing. We model
such applications using a tolerance parameter that describes by how much a task can miss its deadline, or by
how much the task’s execution time can be truncated when the deadline is approaching.

This paper is organized as follows. In Section 2, we present related work. In Section 3, we present our real-
time system model. Numerical results are presented in Section 4. Conclusions are given in Section 5.

2. Related work

The EDF algorithm schedules real-time tasks based on their deadlines. Because of its optimality for peri-
odic, aperiodic, and sporadic preemptive tasks, its optimality for sporadic non-preemptive tasks, and its
acceptable performance for periodic and aperiodic non-preemptive tasks, EDF is widely studied as a dynamic
priority-driven scheduling scheme [5]. EDF is more efficient than many other scheduling algorithms, including
the static Rate-Monotonic scheduling algorithm. For preemptive tasks, EDF is able to reach the maximum
possible processor utilization when lightly loaded. Although finding an optimal schedule for periodic and ape-
riodic non-preemptive tasks is NP-hard [6,7], our experiments have shown that EDF can achieve very good
results even for non-preemptive tasks when the system is lightly loaded. However, when the processor is over-
loaded (i.e., the combined requirements of pending tasks exceed the capabilities of the system) EDF performs
poorly. Researchers have proposed several adaptive techniques for handling heavily loaded situations, but
they require the detection of the overload condition.

A Best-effort algorithm [8] is based on the assumption that the probability of a high value-density task
arriving is low. The value-density is defined by V/C, where V is the value of a task and C is its worst-case exe-
cution time. Given a set of tasks with defined values for successful completion, it can be shown that a sequence
of tasks in decreasing order by value-density will produce the maximum value as compared to any other sched-
uling technique. The Best-effort algorithm admits tasks based on their value-densities and schedules them
using the EDF policy. When higher value tasks are admitted, some lower value tasks may be deleted from
the schedule or delayed until no other tasks with higher value exist. One key consideration in implementing
such a policy is the estimation of current workload, which is either very difficult or very inaccurate in most
دریافت فوری متن کامل مقاله

امکان دانلود نسخه تمام متن مقالات انگلیسی
امکان دانلود نسخه ترجمه شده مقالات
پذیرش سفارش ترجمه تخصصی
امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
امکان دانلود رایگان ۲ صفحه اول هر مقاله
امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
دانلود فوری مقاله پس از پرداخت آنلاین
پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات