Empirical Study of Adaptive EDF

WU Qiong (1310021)

School of Information Science,
Japan Advanced Institute of Science and Technology,

February 12, 2015

Keywords: RTOS, Scheduling, Deadline, EDF, RM, TBS, Adaptive EDF, Retrospective Releasing, ITRON

1 Introduction

Real-time operating systems (RTOS) are designed to handle real time-applications. RTOS should guarantee to finish tasks within a certain time. RTOS manages tasks using well-defined and sophisticated scheduling algorithms, such as Earliest Deadline First (EDF), Rate Monotonic (RM), Total Bandwidth Server (TBS) and Adaptive EDF. In this research, a basic EDF scheduler is extended with Adaptive EDF [1] and a new retrospective releasing technique [5]. This combination is implemented and tested. The four schedulers will be compared in the evaluations.

In this research, both EDF and Adaptive EDF have been implemented and evaluated by using an actual real-time operating system (RTOS) and real programs running on a CPU simulator. In addition, a new additional function “retrospective releasing” to EDF and Adaptive EDF, which is newly proposed scheduling technique is implemented, and the effectiveness is evaluated in the evaluation.

2 Related Work

Real-time operating system, EDF, RM and TBS algorithms with examples are introduced in related works. A real-time system is a class of systems that when some events happen it processes data with specified time constraints. There are two main goals for any RTOS. The first is meeting the tasks timing constraints, which requires timely execution of those tasks. The second is shortening response times and reducing jitters of important tasks compared to the other application tasks. EDF is a widely used real-time task scheduler based on dynamic priority assignment. RM is a scheduling algorithm which fixes tasks' priorities preferentially. TBS is a scheduling algorithm that handles both periodic and aperiodic tasks. TBS schedules tasks based on EDF.

3 Adaptive EDF

Adaptive EDF is a scheduling algorithm for periodic tasks which aims to reduce response times of certain important tasks as well as their jitter in real-time systems. Adaptive EDF was devised based on Earliest Deadline First (EDF). According to the evaluation in [1] [2], the decrease of response times of a particular (target or important) task is better compared to most of the existing algorithms. Adaptive EDF can make target important tasks run before the other tasks even when the target tasks have longer periods or later deadlines. Therefore Adaptive EDF achieves shorter response time and smaller jitters for the target important tasks. Adaptive EDF algorithm and an example are also presented.

4 Retrospective Releasing

Retrospective Releasing as a new technique which is basically a function added on top of both EDF and Adaptive EDF. Retrospective releasing is an enhancement of the Total Bandwidth Server (TBS) and used to achieve shorter response times and small jitters for particular periodic tasks, by assuming an advanced release (invocation) time. Retrospective releasing technique, tries to assume an earlier release time to obtain an earlier deadline. Thus, it can produce the shorter response times and smaller jitters. The retrospective releasing technique advances release times but never

influences the past schedules. This means that the retrospective releasing technique can be performed only when tasks' scheduling order never changes between the virtual release time and the actual release time. Therefore, in order to use this kind of technique, previous deadline, empty slot and maximum used deadline must be considered.

5 Implementation

The proposed methods are implemented in the ITRON RTOS kernel. First, EDF is implemented, and then the Adaptive EDF is added with a small change. Further, the retrospective releasing technique is included. All the additional codes are written in a few functions; one inserting a task in the ready queue, scheduler function, and timer handling functions.

6 Evaluation

In this research, a clock-cycle-based CPU simulator is used for evaluating the scheduling algorithms. The CPU simulator executes binary executable codes in a cycle-based fashion. In this evaluation, binary codes consisting of tasks' codes and OS codes are input to the simulator. Therefore, this evaluation enables accurate quantitative assessment of the whole system. For the evaluation, there are many task sets are prepared. Each task is an actual C language program code. From the simulation results, Adaptive EDF is found to exhibit shorter response times and jitters than other techniques. In addition, retrospective releasing technique is effective especially combined with the conventional EDF.

7 Conclusion and Future Work

In this research, the Adaptive EDF with the retrospective releasing technique is proposed to obtain earlier deadlines and reduce response times and jitters. The retrospective releasing can be combined with not only Adaptive EDF but the conventional EDF with a small amount of overheads. In the evaluation based on clock-cycle simulations including the RTOS runtime overheads, the Adaptive EDF exhibited better real-time performances than the conventional techniques. The retrospective releasing has considerable effectiveness when combined with the conventional EDF. However, it does not a substantial benefit with the Adaptive EDF. This is

because the Adaptive EDF can provide the target tasks with deadlines early enough and also because the execution overhead of the retrospective releasing becomes an obstacle to further improvement. Therefore, implementation of this technique with more light-weight complexity should be explored in the future work.

References

[1] K.Tanaka, "Adaptive EDF: Using predictive Execution Time," ACM SIGBED Review, Vol.10, No.4, pp.41-44, 2013.

[2] K.Tanaka, "Improvement of Adaptive EDF," ACM SIGBED Review, Vol.11, No.3, pp.40-43, 2014.

[3] G.C. Buttazzo, "Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications," 3rd edition, Springer, 2011.

[4] C.L.Liu and J.W.Layland, "Scheduling Algorithms for Multiprogramming in Hard-Real-Time Environment," Journal of the Association for Computing Machinery, Vol. 20, No. 1, pp. 46-61, 1973.

[5] K.Tanaka, "Real-Time Scheduling for Reducing Jitters of Periodic Tasks," Proc. of ISPJ Embedded Systems Symposium, pp.36-45, 2004.