Title	A Study of Reducing Jitter and Energy Consumption in Hard Real-Time Systems using Intra-task DVFS Techniques
Author(s)	Tseng, Bo-Yu
Citation	
Issue Date	2018-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/15464
Rights	
Description	Supervisor:田中　清史，先端科学技術研究科，修士（情報科学）

A study of reducing jitter and energy consumption in hard real-time systems using Intra-task DVFS techniques

BO-YU TSENG (1610105)

School of Information Science, JAIST, tseng@jaist.ac.jp

Keywords— finish time jitter, Dynamic Voltage and Frequency Scaling, Control and Data Flow Analysis, response time, worst-case interference time, worst-case execution time

Extended Abstract

In the hard real time systems, every periodic task is required to perform deterministic timing behaviour, i.e., explicit timing constraint (specific deadline, low jitter). Among them, schedulability of multitasking is the highest criticality in hard real-time system which is not allowed to be violated. To maintain the schedulability, feasible task scheduling algorithms have been studied for decades. However, predictability of periodic tasks response times is also a critical concern in some real-time applications, e.g., control system or data acquisition. In the context of task scheduling, due to the task interaction (preemption, precedence constraint or synchronisation protocol), memory access latency, shared-resource contention or I/O device accesses, systems may face great runtime variation in response time, called finish time jitter. Large finish time jitter impacts predictability which can degrade stability or processing accuracy of system.

To reduce jitter, deadline assignment algorithm by linear programming was proposed. Deadline assignment alg. attempts to shorten relative deadlines of some periodic tasks whilst keeping the schedulability, by promoting priorities of certain tasks to reduces the number of preemptions. Variation in preemption duration makes contributes to jitter, accordingly the less the preemption, the less the jitter. However, this approach only takes preemptions into account. In fact, variation in execution time is another factor leading to jitter. Other works that exploit Dynamic Voltage and Frequency Scaling (DVFS) to handle jitter. DVFS has been widely utilised in energy efficiency, using slack time to scale down the operating frequency. In the meanwhile, DVFS enables the system to control the actual execution/response times of periodic tasks, thus it is applicable to reducing finish time jitter. Mochocki, et al. exploited only the suitable portion of slack time to scale down the operating frequency for some lower-priority tasks instances instead of aggressively using all slack time for energy reduction. Their work is based on Inter-task perspective, hence frequency

scaling can be performed only at the start time of instances. Phatrapornnant and Pont proposed a similar jitter-aware DVFS algorithm called TTC-jDVS algorithm, which incorporates jitter reduction in an Inter-task DVFS scheme. However, it reduces start time jitter only and ignores finishing time jitter.

Accordingly, the objective of this research is to reduce finishing time jitter under Rate-Monotonic scheduling (RM). A jitter-aware Intra-task DVFS scheme is proposed to make task scheduling adapt to runtime variations due to both interference and execution time, whilst keeping energy efficiency. The Intra-task DVFS approach promises finer granularity of frequency scaling within any instance of tasks' executions. Thus, it relatively outperforms the Inter-task DVFS approach in terms of energy reduction. Apart from the effect of energy efficiency, it is also expected that the Intra-task DVFS approach can manipulate finishing time jitter. This work is the first to control the finish time jitter using Intra-task DVFS, to the best of the authors' knowledge.

To avoid large finish time jitter, an usage of DVFS technique is proposed in this research which changes the execution speed to adapt to the actual execution flow of tasks program in the perspective of control flow graph and actual interference time, i.e., controls the actual response time by proactive approach. The procedures of the proposed intra-task DVFS scheme is organised by five phases below:

- **Runtime profiling.** The three additional control parameters are added into TCB_i to get required profiling information (recorded maximum response time $\mathbf{R_i^{max}}$, recorded minimum response time $\mathbf{R_i^{min}}$, and actual interference time $\mathbf{I^{actual}(i)}$). In addition, one global control parameter for the whole task set, global slack time $\mathbf{Slack_{global}}$. It represents the total difference between WCET and the actual execution time of the currently running task.
- **Control and data flow analysis.** Analysing the diversity of every periodic task's execution behaviours at source code level.
- **Execution cycle estimation.** Evaluating the processing cost of each task's execution behaviours.
- **Frequency-scaling point placement.** Determining the invocation points of DVFS operation within each task's runtime.
- **Frequency-updated ratio calculation.** Deciding the new operating frequency for meeting given jitter constraints.

.

For experiment, we built a CFG-based multitasking simulator in C++11. For target tasks, we selected four different CFGs of benchmark programs and made another simple CFG, showing in table II below. Two tasks sets made by target tasks were simulated with Rate-Monotonic scheduling for their hyperperiod. The frequency-power combinations used in the simulation are 300MHz-114.38mW, 600MHz-303.15mW, 720MHz-437.49mW, 800MHz-542.73mW, and 1000MHz-736.08mW. In the simulation, each target task set is simulated five times with different execution paths generated by the test pattern generator, and every task scheduling of each task set is run until the the system tick counts of 500000 nanosecond. The average value of (i) absolute finish time jitter and (ii) energy consumption of the jitter-sensitive tasks are used in the comparison. Through evaluation by multitasking simulation, it is shown that jitter can be reduced by up to 16.2%-19.4%, and energy saving by up to 13.6%-18.39% as side effect.

This paper proposed jitter-aware Intra-task DVFS techniques for reducing jitter in hard real-time systems. We exploited DVFS technique to reduce runtime variation in both interference and execution time, with the cooperation of control and data flow analysis. To decide effective frequency-scaling factor at every DVFS operation, a jitter margin was defined to clarify the lower and upper bounds of possible finish time jitter, also four control parameters were prepared for profiling runtime situation manipulated by system.

Currently, our ongoing work is trying to find a tradeoff between jitter and energy. Different power profiles are being mapped to the frequency settings used in this paper. Thorough assessment under various jitter and energy constraints are to be considered as our future extension. Together with the currently overlooked switching overhead, which could possibly limit the number of frequency-scaling points.