New Optimal Preemptively Scheduling for Real-Time Reconfigurable Sporadic Tasks Based on Earliest Deadline First Algorithm

Hamza Gharellaoui, INSAT Institute & University of Carthage, Tunisia
Mohamed Khalgui, INSAT Institute, Tunisia, ITIA Institute - CNR Research Council, Italy, & Xidian University, China
Samir Ben Ahmed, INSAT Institute & University of Tunis El Manar, Tunisia

ABSTRACT

This paper examines the problem of scheduling the mixed workload of both sporadic (on-line) and periodic (off-line) tasks on uniprocessor in a hard real-time environment. The authors introduce an optimal earliest deadline scheduling algorithm to optimize response time while ensuring that all periodic tasks meet their deadlines and to accept as many sporadic tasks. A necessary and sufficient schedulability test is presented, and an efficient O(n+m) guarantee algorithm is proposed. This optimal algorithm results in dynamic scheduling solutions. They are presented by a proposed intelligent agent-based architecture where a software agent is used to evaluate the response time, to calculate the processor utilization factor and also to verify the satisfaction of real-time deadlines. The agent dynamically provides technical solutions for users where the system becomes unfeasible by sending sporadic tasks to idle times, by modifying the deadlines of tasks, the worst case execution times (WCETs), the activation time, by tolerating some non critical tasks according to the (m, n) firm and a reasonable cost, or in the worst case by removing some non hard (soft) tasks according to predefined heuristic. The authors implement the agent to support these services which are applied to extensive experiments with real-life design examples in order to demonstrate the effectiveness and the excellent performance of the new optimal algorithm in normal and overload conditions.

Keywords: Algorithm, Real-Time Environment, Real-Time Reconfigurable Sporadic Tasks, Scheduling, Worst Case Execution Times (WCETs)

INTRODUCTION

Nowadays, due to the growing class of portable systems, such as personal computing and communication devices, embedded and real-time systems contain complex software which is increasing by the time. This complexity is growing because many available software development models don’t take into account the specific needs of embedded and systems development.
The software engineering principles for embedded systems should address specific constraints such as hard timing constraints, limited memory and power use, predefined hardware platform technology, and hardware costs. The new generations of embedded control systems are addressing new criteria such as flexibility and agility (Gharsellaoui, Khalgui, Gharbi, & Ben Ahmed, 2012). For these reasons, there is a need to develop tools, methodologies in embedded software engineering and reconfigurable embedded control systems as an independent discipline. By response for this requirement of developing reconfigurable systems, many interesting academic and industrial studies have been made in recent years. A reconfiguration scenario means the addition, removal or update of tasks in order to save the whole system on the occurrence of hardware/software faults, or also to improve its performance when random disturbances happen at run time. A random disturbance is defined in the current paper as any random internal or external event allowing the addition of sporadic tasks or removal of sporadic/periodic tasks to adapt the system’s behavior.

Indeed, a hard real-time system typically has a mixture of off-line and on-line workloads. The off-line requests support the normal functions of the system while the on-line requests are sporadic tasks to handle external events such as operator commands and recovery actions which are usually unpredictable. This unpredictable nature of the external events makes on-line scheduling difficult.

Also due to the incomplete knowledge of the online sporadic tasks, the estimating of the future idle times is more difficult.

We consider in this paper the problem of preemptively scheduling that mixed a workload on uniprocessor to optimize the response time and while ensuring that all off-line requests meet their deadlines and to accept as many sporadic requests which must be guaranteed to meet their deadlines after a reconfiguration scenario by using the earliest deadline first (EDF) scheduling algorithm.

Indeed, many real-time systems rely on the EDF scheduling algorithm. This algorithm has been shown to be optimal under many different conditions.

For example, for independent, preemptable tasks, on a uniprocessor EDF is optimal in the sense that if any algorithm can find a schedule where all tasks meet their deadlines, then EDF can meet the deadlines (Dertouzos, 1974). Also, Jackson’s rule (Jackson, 1955) says that ordering a set of tasks by deadline will minimize the maximum lateness. Further, it has also been shown that EDF is optimal under certain stochastic conditions (Towsley & Panwar, 1991). In spite of these advantageous properties, EDF has one major negative aspect. That is, when using EDF in a dynamic system, if overload occurs, tasks may miss deadlines in an unpredictable manner, and in the worst case, the performance of the system can approach zero effective throughputs (Locke, 1986). Moreover, our approach combines many nice scheduling features, further enhancing its optimality. The main contribution of this paper is the development and the performance evaluation of an optimal version of the EDF algorithm. To illustrate the key point of the proposed dynamically approach, we define a new real-time embedded control system in the study

\[ \xi = Sys_1 \cup Sys_2 , \]

Where \( Sys_1 \) is a set of \( n \) periodic tasks, i.e., \( Sys_1 = \{\tau_1, \tau_2, \ldots, \tau_n\} \) and \( Sys_2 \) is a set of \( m \) active sporadic tasks \( \sigma_i \) ordered by increasing deadline in a linked list, i.e., \( Sys_2 = \{\sigma_1, \sigma_2, \ldots, \sigma_n\} \). \( \sigma_i \) being the task with the shortest absolute deadline.

According to Gharsellaoui, Khalgui, Gharbi, and Ben Ahmed (2012), each periodic task is described by an initial offset \( a_i \) (activation time), a worst-case execution time \( C_i \), a relative deadline \( D_i \) and a period \( T_i \).

According to Buttazzo and Stankovic (1993), each sporadic task is described by
Mobile and Pervasive Technology in Education and Training: Potential and Possibilities, Problems and Pitfalls
www.igi-global.com/chapter/mobile-pervasive-technology-education-training/37806?camid=4v1a