Aperiodic Task Scheduling for Hard-Real-Time Systems

BRINKLEY SPRUNT  
Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

LUI SHA  
Department of Computer Science and Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213

JOHN LEHOCZKY  
Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract. A real-time system consists of both aperiodic and periodic tasks. Periodic tasks have regular arrival times and hard deadlines. Aperiodic tasks have irregular arrival times and either soft or hard deadlines. In this article, we present a new algorithm, the Sporadic Server algorithm, which greatly improves response times for soft deadline aperiodic tasks and can guarantee hard deadlines for both periodic and aperiodic tasks. The operation of the Sporadic Server algorithm, its performance, and schedulability analysis are discussed and compared with previously published aperiodic service algorithms.

1. Introduction: the real-time scheduling problem

Real-time systems are used to control physical processes that range in complexity from automobile ignition systems to controllers for flight systems and nuclear power plants. In these systems, the correctness of system functions depends upon not only the results of computation but also on the times at which results are produced.

A real-time task is generally placed into one of four categories based upon its arrival pattern and its deadline. If meeting a given task's deadline is critical to the system's operation, then the task's deadline is considered to be hard. If it is desirable to meet a task's deadline but occasionally missing the deadline can be tolerated, then the deadline is considered to be soft. Tasks with regular arrival times are called periodic tasks. A common use of periodic tasks is to process sensor data and update the current state of the real-time system on a regular basis. Periodic tasks, typically used in control and signal-processing applications, have hard deadlines. Tasks with irregular arrival times are aperiodic tasks. Aperiodic tasks are used to handle the processing requirements of random events such as operator requests. An aperiodic task typically has a soft deadline. Aperiodic tasks that have hard deadlines are called sporadic tasks. We assume that each task has a known worst-case execution time. In summary, we have
**Hard and soft deadline periodic tasks.** A periodic task has a regular interarrival time equal to its period and a deadline that coincides with the end of its current period. Periodic tasks usually have hard deadlines, but in some applications the deadlines can be soft.

**Soft deadline aperiodic tasks.** An aperiodic task is a stream of jobs arriving at irregular intervals. Soft deadline aperiodic tasks typically require a fast average response time.

**Sporadic tasks.** A sporadic task is an aperiodic task with a hard deadline and a minimum interarrival time (Mok 1983). Note that without a minimum interarrival time restriction, it is impossible to guarantee that a sporadic task’s deadline would always be met.

To meet the timing constraints of the system, a scheduler must coordinate the use of all system resources using a set of well-understood real-time scheduling algorithms that meet the following objectives:

- Guarantee that tasks with hard timing constraints will always meet their deadlines.
- Attain a high degree of schedulable utilization for hard deadline tasks (periodic and sporadic tasks). Schedulable utilization is the degree of resource utilization at or below which all hard deadlines can be guaranteed. The schedulable utilization attainable by an algorithm is a measure of the algorithm’s utility: the higher the schedulable utilization, the more applicable the algorithm is for a range of real-time systems.
- Provide fast average response times for tasks with soft deadlines (aperiodic tasks).
- Ensure scheduling stability under transient overload. In some applications, such as radar tracking, an overload situation can develop in which the computation requirements of the system exceed the schedulable resource utilization. A scheduler is said to be stable if during overload it can guarantee the deadlines of critical tasks even though it is impossible to meet all task deadlines.

The quality of a scheduling algorithm for real-time systems is judged by how well the algorithm meets these objectives.

This article develops advanced algorithms to schedule aperiodic tasks. For soft deadline aperiodic tasks, the goal is to provide fast average response times. For hard deadlines aperiodic tasks (sporadic tasks), the goal is to guarantee that their deadlines will always be met. The new algorithms presented here meet both of these goals and are still able to guarantee the deadlines of hard deadline periodic tasks. For simplicity, we assume that periodic tasks have hard deadlines and constant execution times. In Section 2 we review related work on scheduling periodic and aperiodic tasks. Section 3 introduces the Sporadic Server (SS) algorithm for scheduling aperiodic tasks and illustrates its operation with several examples. This section also addresses the schedulability issues of the SS algorithm and compares its performance with previous algorithms. Section 4 addresses the scheduling of sporadic tasks and discusses the use of a deadline monotonic priority assignment for scheduling sporadic tasks with short deadlines and long interarrival times. Finally, Section 5 presents a summary and conclusions.