Handling Aperiodic Overloads

A seminar paper for Special Emphasis A - Real Time Systems

Yashodhan Vishvesh Deshpande
Hamm-Lippstadt University of Applied Sciences, Lippstadt, Germany
yashodhan-vishvesh.deshpande@stud.hshl.de

***Abstract*—The ever increasing demand for sophisticated products to improve modern lifestyle has created the need for developing complex real time systems. With the advancement in artificial intelligence the computation requirements of real time systems has grown exponentially. This can result in aperiodic overloads on the computing system which might lead to real time result failures. To avoid these failures it is necessary to have a method that addresses the issue of aperiodic overloads.**

This seminar paper is an in depth dive into methods of handling aperiodic overloads. It discusses the basic concepts related to handling aperiodic overloads and gives an overview of the algorithm needed to handle the overloads.

***Index Terms*—aperiodic, overloads**

I. Introduction

A. Periodic vs Aperiodic tasks

Periodic tasks are similar to each other in terms of time require to perform the task. Periodic tasks are performed in a fixed schedule since they are identical. Hence the starting time in the scheduling algorithm is at a constant speed.

Aperiodic tasks are also similar to each other in terms of computational times. However they are not performed in a scheduling algorithm. They can be started at any time.

B. Overload and types of overload

In real-time systems load on a particular processing system is calculated by the queuing theory formula. That is

Load = (average processing time of the processes) * (rate of process arrivals)

However this theory does not take into account the time intervals for a particular process. Hence it is only suitable for soft real time systems. For hard real-time systems a more sophisticated theory is needed so that the time intervals are taken into account since failing to meet a deadline can be fatal in hard real time systems.

To measure the load at specific time in a processing system a method was realized by Buttazzo and Stankovic. According to this method, load (p) of every job (J) causing on the system is calculated from the chosen time instance (t) until the deadline of the job (d). The c(t) is the total sum of remaining processing time of the selected job. The algorithm to measure the load is shown in Figure 1 : The highest load on the processor is the load p(t) of the job that has highest load among all the other jobs in the selected period of time.

A task processing system is overloaded when the total required processing time for the task in a fixed time slot is more than the available time of the processing system in the same time slot.

$$\rho_i(t) = \frac{\sum_{d_k \leq d_i} c_k(t)}{(d_i - t)},$$

Fig. 1. Instantaneous load calculation

There are two types of overloads:

1) Transient Overloads: Transient overloads are overloads with p(t) $>$ 1, for a specific period of time of of the whole processing duration. These types of overloads have average load value of smaller than or equal to 1.0 .

This type of overloads is caused in systems that works with event triggered scheduling algorithms. The external interrupts in a event triggered system can cause missing of a deadline due to sudden overload in a short duration. This can cause a serious hazard in a hard real time system as missing a deadline is fatal.

The earliest deadline first is an algorithm in which the tasks are computed in a priority based scheduling. The tasks with the earliest deadline or upcoming deadline is give the highest priority as compared to other tasks with later deadlines.

A processing system designed to work with earliest deadline first scheduling algorithm that is computing aperiodic tasks can run into several problems. One of the problematic phenomenon is called domino effect. In this phenomenon a hard real time system working reliably and predictably in handling its tasks is interrupted by an aperiodic external event. This event can cause an overload of the system in which the external task can have a higher priority due to its earlier deadline. The external task will be performed earlier than the other regular tasks. This rescheduling of the task priorities can cause missing of the regular task deadlines. Delaying one task can lead to delaying of other tasks in the schedule. This causes a domino effect of missing multiple deadlines of the other tasks in the schedule.

2) Permanent Overload: Permanent overloads are overloads with p(t) $>$ 1 for a uncertain period of time. This type of overload has average value of more than 1.0 for the whole duration of processing.