Deferrable Server (DS)

Heansuh Lee
Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
heansuh.lee@stud.hshl.de

***Abstract*—In this paper, the Deferrable Server (DS) will be covered in details. It is the simplest bandwidth-preserving server that improves response time of aperiodic tasks as compared to polling server.**

Unlike polling server, DS preserves its budget if no aperiodic requests are pending upon its invocation. It schedules aperiodic jobs that arrive later during its period until the budget is exhausted.

This paper also covers the usage of established mathematical proofs and how to implement them properly in practice, which cover (1) Consumption Rule, and (2) Replenishment Rule, for a better understanding of this concept.

DS allows users to have the knowledge to plan and set up a real-time system both on paper and in practice. It is critical to achieve timing correctness in embedded systems, which means to guarantee that the system reacts within the real-time requirements.

***Index Terms*—DPE, dynamic priority servers, real-time, dynamic,**

I. Introduction

This section describes briefly about Deferrable Server (DS).

The Dynamic Priority Exchange (DPE) server is an aperiodic service technique proposed by Spuri and Buttazzo, that can be viewed as an extension of the Priority Exchange server, adapted to work with a deadline-based scheduling algorithm. The main idea of the algorithm is to let the server trade its runtime with the runtime of lower-priority periodic tasks (under EDF this means a longer deadline) in case there are no aperiodic requests pending.

In this way, the server runtime is only exchanged with periodic tasks but never wasted (unless there are idle times). It is simply preserved, even if at a lower priority, and it can be later reclaimed when aperiodic requests enter the system [1, p.162].

A. Scheduling of Aperiodic Tasks

- talk about deferrable server briefly, and why is it called ”deferrable”

B. Definition

The algorithm is defined as follows [1, p.163]:

- The DPE server has a specified period T_s and a capacity C_s.
- At the beginning of each period, the server's aperiodic capacity is set to C_S^d, where d is the deadline of the current server period.
- Each deadline d associated to the instances (completed or not) of the ith periodic task has an aperiodic capacity, $C_{S_i}^d$, initially set to 0.
- Aperiodic capacities (those greater than 0) receive priorities according to their deadlines and the EDF algorithm, like all the periodic task instances (ties are broken in favor of capacities; that is, aperiodic requests).
- Whenever the highest-priority entity in the system is an aperiodic capacity of C units of time the following happens:
  - if there are aperiodic requests in the system, these are served until they complete or the capacity is exhausted (each request consumes a capacity equal to its execution time);
  - if there are no aperiodic requests pending, the periodic task having the shortest deadline is executed; a capacity equal to the length of the execution is added to the aperiodic capacity of the task deadline and is subtracted from C (that is, the deadlines of the highest-priority capacity and the periodic task are exchanged);
  - if neither aperiodic requests nor periodic task instances are pending, there is an idle time and the capacity C is consumed until, at most, it is exhausted.

C. Example

- example to be explained with Fig.1

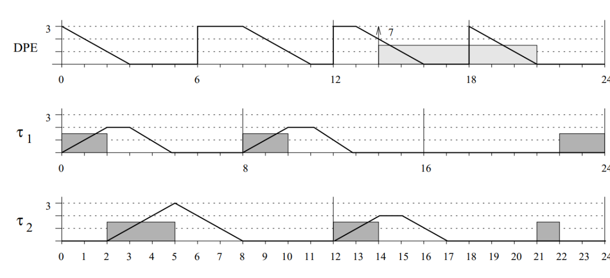

Fig. 1. Example of DPE server.

II. Schedulability Analysis

- about schedulability analysis

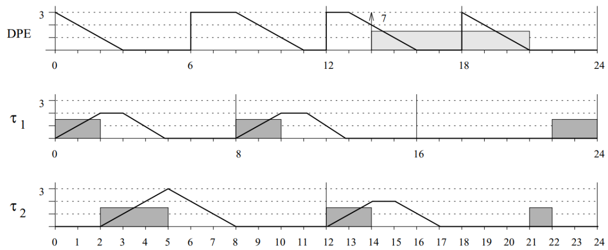

Fig. 2. DPE server schedulabililty.

III. Consumption Rule

- about the consumption rule

IV. Replenishment Rule

- about the replenishment rule

V. Evaluation

- evaluate pros and cons

A. Implementation

- how is it implemented

B. Advantages

- what are the advantages? (eg. periodic tasks are done and nothing goes to waste)

C. Constraints

- very hard to implement, according to the paper [3].

D. Comparison to current technology

- compare DPE server to other types of dynamic priority servers briefly

VI. Conclusion

- summary of the whole paper and discuss about evaluation parts mainly and briefly

References

[1] https://www.coursera.org/lecture/real-time-systems/the-deferrable-server-example-Jhjr8

[2] https://www.computer.org/csdl/journal/tc/1995/01/t0073/13rRUwvBy82

[3] https://cpen432.github.io/resources/bader-slides/9-TaskServers.pdf

[4] https://benchpartner.com/deferrable-server

[5] https://bears.ece.ucsb.edu/class/ece253/papers/spuri96.pdf - main paper from buttazzo

[6] https://www.d3s.mff.cuni.cz/legacy/teaching/embedded_realtime_systems/files/lectures/10b%20-%20dynamic%20priority%20servers.pdf - slides 3 to 7

[7] https://www.seas.upenn.edu/~lee/10cis541/lecs/lec-RT-sched-part2-v2-1x2.pdf - pg 8