Efficient Aperiodic Task Handling in Real-Time Systems Using the Deferrable Server Model

Sayed Galib Hossen Rizvi
Real-Time Systems Course
Hamm-Lippstadt University of Applied Sciences
Lippstadt, Germany
sayed-galib-hossen.rizvi@stud.hshl.de

***Abstract*—Predictable and timely completion of tasks is crucial in real-time systems, particularly when scheduling time-critical periodic and aperiodic tasks. A scheduler called Deferrable Server (DS) has been proposed to improve the responsiveness of service of aperiodic tasks in fixed-priority systems without sacrificing the timing guarantees offered to periodic tasks. In comparison to classical polling servers that watch for aperiodic requests at discrete times, the DS maintains its run budget throughout and allows faster reaction times upon receiving an aperiodic task. The Deferrable Server model is presented in detail here together with its methods of operation, advantages, and disadvantages. It also compares DS with other approaches such as Polling and Sporadic Servers and presents two case studies from the literature which illustrate its usability in practice in embedded and hierarchical scheduling environments. The discussion highlights that while the DS improves responsiveness, cautious configuration and analysis need to be conducted in order to make the system schedulable.**

***Index Terms*—Real-Time Systems, Aperiodic Task Scheduling, Deferrable Server, Fixed-Priority Scheduling, Embedded Systems, Schedulability Analysis, Response Time Optimization.**

I. Introduction

Real-time systems need to execute tasks on absolute deadlines to ensure their dependable and predictable functioning. These tasks are usually classified as periodic which happen at constant intervals, and aperiodic, which come at unpredictable intervals but need to be done within a certain time frame. Efficiently servicing aperiodic tasks without impacting the guaranteed timelines of periodic ones is a design problem of a high order.

To address this problem, the Deferrable Server DS algorithm set aside some CPU time for servicing aperiodic tasks, allowing that time to be utilized at any point during the reserving period, rather than at fixed checking points [1]. This flexibility makes a marked difference in responsiveness of aperiodic tasks over simpler methods like polling servers.

However, improper configuration of DS may lead to uncontrolled periodic task interference which creates missed deadlines in fixed priority preemptive scheduling [2]. To introduce any safety in deploying DS, a detailed schedulability analysis is required.

The aim of this document is to disentangle the concepts around the DS algorithm and compare it with server-based strategies that include Polling and Sporadic Servers, in addition to analyzing practical implementations that demonstr

II. Case study and Related Work

The Deferrable Server (DS) model has been thoroughly analyzed in the context of real-time systems to enhance the responsiveness of aperiodic tasks. In contrast to polling servers which enable a task to be executed only at specific time intervals, the Deferrable Server surrenders its execution budget within its entire period, making it immediately responsive to incoming aperiodic requests.

Lehoczky, Sha, and Strosnider originally suggested the concept where they presented the Deferrable Server intending to augment the aperiodic responsiveness without compromising the timing guarantees of the periodic tasks [1]. By ensuring capacity provision throughout the period, DS is able to guarantee better response times than polling servers.

Despite the best effort, DS might add interference in fixed-priority scheduling some interference. Jane W. S. Liu emphasized that Deferrable Servers adversely affect lower-priority periodic tasks without a thorough evaluation and design [2]. In systems with critical real-time constraints, this is certainly problematic.

A. Enhanced Aperiodic Responsiveness in Hard Real-Time Environments – Lehoczky, Sha, and Strosnider (1987)

Lehoczky and his colleagues were the first to use the Deferrable Server to improve aperiodic responsiveness in hard real time environments working [1]. The authors simulated the performance of DS with background and polling approaches and concluded that DS is able to reduce the response time of aperiodic tasks considerably without any loss of schedulability on periodic tasks.

With these results, they were able to show the feasibility of servicing aperiodic requests without considerable delay using the Deferrable Server—making it well suited for systems that need to respond rapidly to infrequent asynchronous events. This enabled the adaption of many other studies and extensions of server-based task handling algorithms.

B. Towards Exploiting the Preservation Strategy of Deferrable Servers – Bril, Lukkien, and Verhaegh (2008

Bril, Lukkien, and Verhaegh analyzed the use of Deferrable Servers in real time systems with hard time predictability for execution [3]. Their work focused on how such a DS could be designed using a preservation strategy to improve predictability in multi-layered scheduling systems.

III. Deferrable Server Concept

In real-time systems, tasks are usually categorized as periodic (execute at fixed rates) or aperiodic (execute at unpredictable but regular intervals). Handling aperiodic tasks is one of the main challenges, especially without disrupting the timing commitments of periodic tasks.

One of the solutions commonly utilized is the Deferrable Server (DS) mechanism, introduced for the first time by Lehoczky, Sha, and Strosnider in 1987. It is designed such that timely service of aperiodic requests is achieved through reservation of part of the CPU time in every period to be available at any moment during that period [4]. This is not like a Polling Server as it observes tasks only when every period is beginning.

"The central idea of a Deferrable Server is to preserve its execution budget for the entire period so that it can immediately react whenever an aperiodic task appears" [2]. This ability makes the DS particularly interesting in systems where the aperiodic tasks are time-sensitive.

While this flexibility is beneficial, it has a drawback. "Incorrect configuration of DS can cause back-to-back execution, where the server runs at the end of a period and again at the start of the next, generating high interference to lower-priority periodic tasks" [2]. This issue is very severe in fixed-priority preemptive scheduling and must be avoided by employing a proper schedulability analysis.

Bril, Lukkien, and Verhaegh took the idea of DS further by studying how its preservation strategy can be applied to improve multi-layered real-time systems. "They proposed extensions to the DS model to make it more predictable when integrated into hierarchical scheduling architectures" [5].

The Deferrable Server can work with Earliest Deadline First (EDF) as well as Rate Monotonic (RM) scheduling. Under EDF, it works well since it is dynamic in nature. "Its response time advantage renders DS suitable for systems that require soft real-time guarantees on aperiodic tasks" [6].

Briefly, the Deferrable Server improves aperiodic responsiveness without the necessity to resort to complex periodic task set restructuring. Yet, designers should carefully avoid interference and perform correct timing analysis.

IV. Advantages and Limitations

A. Advantages

B. Limitations

V. Comparison with Other Servers

VI. Applications

VII. Conclusion

Acknowledgment

I would like to express my sincere gratitude to Prof. Dr. Stefan Henkler for his valuable guidance and support throughout the Real-Time Systems course. His insights and encouragement played a significant role in the completion of this paper.

References

[1] J. P. Lehoczky, L. Sha, and J. K. Strosnider, "Enhanced aperiodic responsiveness in hard real-time environments," in *Proceedings of the IEEE Real-Time Systems Symposium*. IEEE, 1987, pp. 261–270.

[2] J. W. S. Liu, *Real-Time Systems*. Prentice Hall, 2000.

[3] R. J. Bril, J. J. Lukkien, and W. Verhaegh, "Towards exploiting the preservation strategy of deferrable servers," in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2008, pp. 1–4.

[4] J. P. Lehoczky, L. Sha, and J. K. Strosnider, "Enhanced aperiodic responsiveness in hard real-time environments," in *Proceedings of the IEEE Real-Time Systems Symposium*. IEEE, 1987, pp. 261–270.

[5] R. J. Bril, J. J. Lukkien, and W. Verhaegh, "Towards exploiting the preservation strategy of deferrable servers," in *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2008.

[6] G. C. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer, 2011.