Towards budgeting in real-time calculus : deferrable servers

Towards Budgeting in Real-Time Calculus: Deferrable Servers

Pieter J.L. Cuijpers and Reinder J. Bril

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science
Den Dolech 2, 5600 MB Eindhoven, The Netherlands

Abstract. Budgeting of resources is an often used solution for guaranteeing performance of lower priority tasks. In this paper, we take a formal approach to the modeling of a deferrable server budgeting strategy, using real-time calculus. We prove a scheduling theorem for deferrable servers, and as a corollary show that an earlier claim of Davis and Burns, that periodic servers dominate deferrable servers with respect to schedulability, no longer holds when the context of the comparison is slightly generalized.

1 Introduction

One of the main scheduling approaches in real-time computing systems is given by fixed-priority preemptive scheduling (FPPS) [5,6]. The approach is founded on a fixed-priority scheduling theory, supported by a suite of open software standards, commercially available schedulability analysis tools and real-time operating systems, and adopted by leading companies and institutions world-wide.

Scheduling a set of real-time tasks sharing a resource gives rise to the so-called *temporal interference* problem, i.e. a malfunctioning task may cause other tasks to fail to meet their time constraints. An often used solution for this problem is to introduce *resource budgets* for tasks, which provide temporal protection between tasks by guaranteeing a minimal amount of resources [8]. Those budgets are often implemented using so-called *servers* that dispatch the available resources to the tasks that are appointed to them.

In general, a server for a shared processing resource, such as a CPU, is characterized by a *capacity* and a *replenishment period* [1]. The capacity is the maximum amount of resources (i.e. the maximum amount of processing time) that a server can provide to its associated tasks during its replenishment period. The replenishment period is the minimum time between replenishments of the capacity. Servers typically differ with respect to the amount and moment in time of the replenishments and to the preservation of the remaining capacity when none of the appointed tasks is ready to use it.

In this paper, we consider so-called *deferrable servers* [10], which have been studied as implementations of resource budgets in [2,9], amongst others. We focus on [2], because its results improve on earlier work. Deferrable servers are replenished *periodically*, at fixed intervals of time, and have the following preservation policy. When a deferrable server has access to the shared resource, its capacity is provided if one of the tasks is ready to use it. If the tasks are not ready to use it, the deferrable server suspends its access to the resource, preserving its remaining capacity. Capacity can be preserved until the end of the replenishment period. At the end of the server's period any remaining capacity is lost.

Our main contribution, is a formal model of deferrable servers using the real-time calculus of [11]. Real-time calculus is a branch of network calculus [4], which uses max-plus algebra for the algebraic analysis of real-time systems. Our specific interest in formalizing the behavior of deferrable servers using this calculus, was due to a suspicion that the worst-case response time analysis of deferrable servers in [2] becomes pessimistic, rather than exact, in a slightly generalized context. Using our real-time calculus model, we derive a schedulability theorem for deferrable servers with a single task. Application of this theorem for a deferrable server with highest priority leads to an example showing that, in the absence of a low-priority soft real-time task, the analysis in [2] indeed becomes pessimistic, rather than exact. As a consequence, the applicability of deferrable servers may be better than was suggested in [2].

This paper is structured as follows. First, in Section 2, we recapitulate the real-time calculus theory of [11]. Next, in Section 3 we explain how to model servers in this theory, and extend the theory with a model of a deferrable server. Our schedulability theorem is the topic of Section 4. The consequences of our schedulability theorem are discussed in Section 5, where we compare it with the results of [2]. Section 6 summarizes the main contributions of the paper and suggests directions for future research.

2 Real-Time Calculus Preliminaries

In this section, we recall part of the real-time calculus theory of [11]. The starting point of this calculus, is to consider cumulative requests streams $R(t) : \mathbb{R} \to \mathbb{R}$ (from time to amount of requested resources) and cumulative resource streams $C(t) : \mathbb{R} \to \mathbb{R}$ (from time to amount of available resources) in a system. The request stream models the total amount of requested tasks that have entered the system at a certain time, while the resource stream models the total amount of processing power that has been offered to a server. The total amount of tasks that have been processed is modeled by a cumulative request stream $R'(t)$, while the total amount of resources that remains unused is processed by a cumulative resource stream $C'(t)$. For convenience we choose $C(t) = R(t) = C'(t) = R'(t) = 0$ whenever $t < 0$, reflecting that the system is turned on at $t = 0$.

Processing of a sequence of tasks can be depicted as in figure 1. It is obvious, that the total amount of tasks processed at time t can never exceed the amount of tasks processed already at a time $u \leq t$, plus the amount of resources offered between u and t. Furthermore, the amount of processed tasks can never be

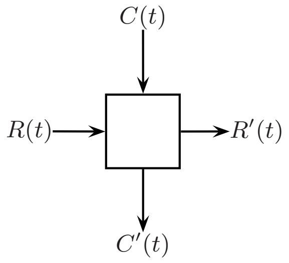

Fig. 1. Basic processing in Real-time Calculus

more than the amount of requested tasks. This lower bound on the output R' is captured in the following formula[1].

$$R'(t) \leq \big(R'(u) + C(t) - C(u)\big) \wedge R(t).$$

With a little calculation we can rewrite this into:

$$R'(t) \leq \inf_{u \leq t} \{R(u) + C(t) - C(u)\}.$$

Now, if we furthermore assume that *tasks may be buffered until resources become available for it*, and that a server is *eager* in the sense that it processes each task as soon as resources are available for it, we can derive a lower bound on $R'(t)$ as well. We define t_0 as the latest point before t at which the resource buffer was empty, assuming $R(0) = R'(0)$ for convenience to show there exists such a point.

$$t_0 \triangleq \sup\{\tau \leq t \mid R(\tau) = R'(\tau)\}$$

Between t_0 and t, the task buffer is always non-empty, which means that all resources that arrive are used for processing tasks. So we find the equality:

$$R'(t) = \big(R'(t_0) + C(t) - C(t_0)\big) \wedge R(t).$$

For a right-continuous resource stream $R(t)$ we then find $R(t_0) = R'(t_0)$ (using the definition of t_0) and thus:

$$\begin{aligned} R'(t) &= (R(t_0) + C(t) - C(t_0)) \wedge R(t), \\ &\geq \inf_{u \leq t} \{R(u) + C(t) - C(u)\} \wedge R(t), \\ &= \inf_{u \leq t} \{R(u) + C(t) - C(u)\}. \end{aligned}$$

In conclusion, $R'(t)$ is exactly determined by

[1] In this paper, we use $x \wedge y$ to denote the minimum of x and y, and $\inf_u\{f(u)\}$ to denote the infimum of $f(u)$ over u. Furthermore, $x \vee y$ denotes the maximum of x and y, and $\sup_u\{f(u)\}$ denotes the supremum of $f(u)$ over u.

$$R'(t) = \inf_{u \le t} \{R(u) + C(t) - C(u)\}.$$

According to [4] the same formula holds if $R(t)$ is left-continuous, but the proof is more complicated. The amount of resources that is left unused can be found easily by subtracting what is used from what is delivered.

$$\begin{aligned} C'(t) &= C(t) - R'(t), \\ &= C(t) - \inf_{u \le t} \{R(u) + C(t) - C(u)\}, \\ &= \sup_{u \le t} \{C(u) - R(u)\}. \end{aligned}$$

In figure 2 we have depicted an example of a task $R(t) = 3 \cdot \lceil \frac{t}{3} \rceil \vee 0$, modeling the arrival of three requests every three time-units. This task is serviced by a resource $C(t) = 2 \cdot t \vee 0$, which brings continuous service to a task at two resource-units per time-unit. Note, that by definition $C'(t)$ is flat whenever $R'(t)$ rises, since eager processing requires that all incoming resources are used as long as the input buffer is non-empty. Vice versa, $R'(t)$ is flat when $C'(t)$ rises, for the same reason.

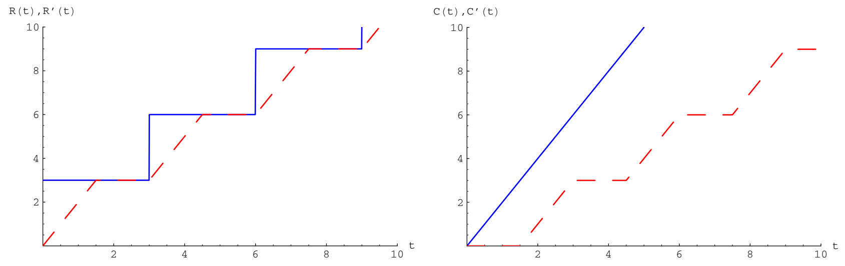

Fig. 2. Requests and resources in RTC (inputs straight, outputs dashed)

3 Deferrable Server Model

In the previous section, we have shown how the processing of a single sequence of tasks can be modeled using real-time calculus. But from a more high-level point of view, the same equations can also be used to represent a server that dispatches resources to a number of tasks. The input stream R_S of the server then is the sum of the input streams of the appointed tasks (whenever one of its tasks has unfinished requests, the server must dispatch incoming resources to it) and the output stream R'_S does not represent finished tasks, but represents resources that have been reserved for the tasks operating under the server. This scheme is depicted in figure 3, where the block labeled S represents a server and

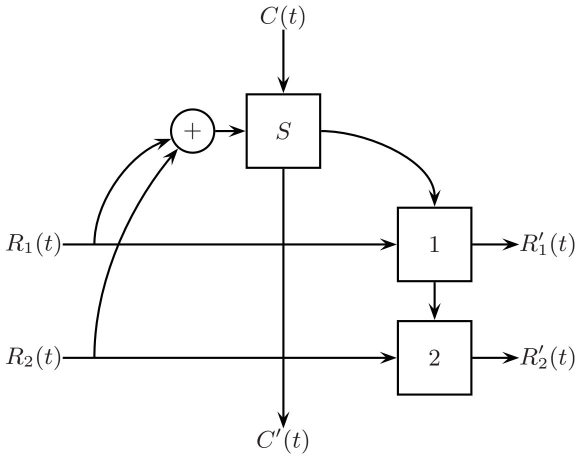

Fig. 3. Server modeling in real-time calculus

blocks 1 and 2 represent tasks that are dispatched to this server. So, we have $R_S = R_1 + R_2$, $C_1 = R_S'$ and $C_2 = C_1'$.

The insight that the equations for a server and for a task are the same, albeit with a different interpretation of the meaning of the variables, still holds when we shift our focus to servers with a budgeting strategy. In other words, this hierarchical way of modeling allows us to model the budgeting of a single task, and use this model for a budgeting server as well. For a budgeting server, we still assume that tasks are processed eagerly, and that tasks are buffered while waiting for resources to arrive. Therefore, the previously derived upper bound on $R'(t)$ is still valid.

$$R'(t) \leq \inf_{u \leq t} \{R(u) + C(t) - C(u)\}.$$

But, additionally, a deferrable server periodically limits the resources it provides to a maximum capacity Q. Replenishment of the capacity takes place at the start of each period T, which at a time t is determined (right-continuously) by $T \cdot \lceil \frac{t}{T} - 1 \rceil$. The output $R'(t)$ of a deferrable server cannot be greater than the output $R'\left(T \cdot \lceil \frac{t}{T} - 1 \rceil\right)$ at the start of the period plus the processing capacity Q. So, we refine the upper bound to capture this behavior.

$$\begin{aligned} R'(t) \leq & \inf_{u \leq t} \{R(u) + C(t) - C(u)\} \\ & \wedge R'\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + Q \\ & \wedge R'\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + C(t) - C\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) \end{aligned}$$

Furthermore, for $t_0 \triangleq \sup\{\tau \leq t \mid R(\tau) = R'(\tau)\} \geq 0$ we find

$$\begin{aligned} R'(t) &= R(t_0) + C(t) - C(t_0) \\ &\quad \wedge R'\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + Q \\ &\quad \wedge R'\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + C(t) - C\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) \\ &\geq \inf_{u \leq t} \{R(u) + C(t) - C(u)\} \\ &\quad \wedge R'\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + Q \\ &\quad \wedge R'\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + C(t) - C\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right). \end{aligned}$$

What results is a recursive specification for the output of the deferrable server

$$\begin{aligned} R'(t) &= \inf_{u \leq t} \{R(u) + C(t) - C(u)\} \\ &\quad \wedge R'\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + Q \\ &\quad \wedge R'\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + C(t) - C\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right). \end{aligned}$$

To prove that this recursive specification has a unique solution for $R'(t)$, we rewrite it to

$$\begin{aligned} R'(t+T) &= \inf_{u \leq t+T} \{R(u) + C(t+T) - C(u)\} \\ &\quad \wedge R'\left(T \cdot \left\lceil \frac{t}{T} \right\rceil\right) + Q \\ &\quad \wedge R'\left(T \cdot \left\lceil \frac{t}{T} \right\rceil\right) + C(t+T) - C\left(T \cdot \left\lceil \frac{t}{T} \right\rceil\right). \end{aligned}$$

From this representation it is clear that, if the solution of the recursive specification is unique upto a point t, then it is unique upto $t + T$. Furthermore, we find uniqueness for $t < 0$ where we have $R'(t) = 0$. So, the solution is unique upto 0, and with induction upto $n \cdot T$ for any $n \in \mathbb{N}$. We may conclude that $R'(t)$ is well defined, and may even solve the recursive specification to find:

$$\begin{aligned} R'(t) = \inf_{n \in \mathbb{N}} \inf_{u \leq t \wedge T \cdot \lceil \frac{t}{T} - n \rceil} \Bigg\{ & R(u) + C(t \wedge T \cdot \left\lceil \frac{t}{T} - n \right\rceil) - C(u) \\ & + \sum_{m=0}^{n-1} Q \wedge \left(C(t \wedge T \cdot \left\lceil \frac{t}{T} - m \right\rceil) - C(T \cdot \left\lceil \frac{t}{T} - m - 1 \right\rceil) \right) \Bigg\}. \end{aligned}$$

As before, we find $C'(t)$ by subtracting what is used from what is delivered. With a little calculation, we obtain:

$$C'(t) = \sup_{n\in\mathbb{N}} \sup_{u\leq t\wedge T\cdot\lceil\frac{t}{T}-n\rceil}$$
$$\left\{C(u) - R(u) + \sum_{m=0}^{n-1}\left(C\left(t\wedge T\cdot\left\lceil\frac{t}{T}-m\right\rceil\right) - C\left(T\cdot\left\lceil\frac{t}{T}-m-1\right\rceil\right) - Q\right)\vee 0\right\}.$$

In figures 4 we have depicted an example of a task $R(t) = 3\cdot\lceil\frac{t}{3}\rceil\vee 0$, serviced by a resource $C(t) = 2\cdot t\vee 0$, on a deferrable server with $Q = 2$ and $T = 2$. Note, that as before, $C'(t)$ is flat whenever $R'(t)$ rises, and vice versa, but the pattern is different from the normal RTC processing.

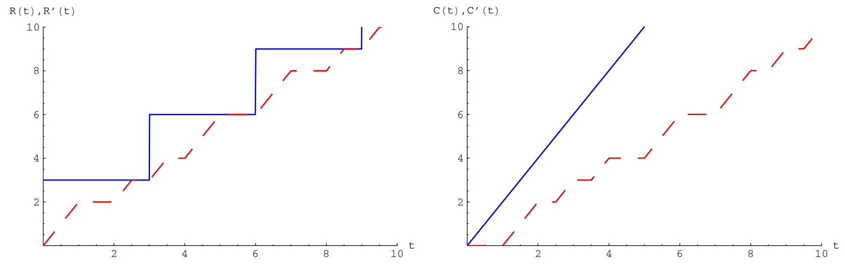

Fig. 4. Requests and resources in a deferrable server, (inputs straight, outputs dashed)

4 Scheduling Theorem

The delay of a task entering at time t is the time between its request and its completion. If we assume that *tasks are completed in the order in which they arrive*, the delay is the earliest time τ at which $R'(t+\tau)\geq R(t)$. The maximum delay Δ in the processing of a sequence of tasks is then defined by:

$$\Delta \triangleq \sup_t \inf\{\tau \mid R'(t)\geq R(t-\tau)\}.$$

Based on this definition, and the model of a deferrable server found in the previous section, we will now prove the following schedulability theorem. This theorem roughly states that task with a deadline greater than the sum of the minimum interarrival time of the task and the maximum delay between arrival of resources (due to other servers in the network) is schedulable (i.e. makes its deadline) provided that the tasks utilization is smaller than the utilization of the server and smaller than the utilization of the arriving resources.

Theorem 1 (Schedulability of a deferrable server). *Consider a deferrable server with period T and capacity Q. Assume that there is an upper bound S on the arrival time of tasks $R(t)$ such that for all $s\in\mathbb{R}$:*

$$\frac{R(s+S)-R(s)}{S} \leq \frac{Q}{T}.$$

Furthermore, assume that there is a lower bound $U \leq T$ on the arrival times of resources, such that for all $u \in \mathbb{R}$:

$$C(u+T) - C(u) \geq Q,$$
$$\frac{C(u+U)-C(u)}{U} \geq \frac{R(s+S)-R(s)}{S}.$$

Then, all relative deadlines of at least $S + 2 \cdot U$ are met, i.e.

$$\Delta \triangleq \sup_t \inf\{\tau \mid R'(t) \geq R(t-\tau)\} \leq S + 2 \cdot U.$$

Proof. The complete proof is by algebraic manipulation of the equations of the deferrable server model and can be found in the appendix.

In the special but in practice not uncommon case (see [1]) where the deferrable server has highest priority, the resource stream $C(t)$ can be considered to be linear, i.e. $C(t) = C \cdot t$. As a corollary, we then find that deadlines of at least S are met.

Corollary 1. *Consider a highest-priority deferrable server with period T and capacity Q. Furthermore, assume that we have an upper bound S on the arrival time of tasks $R(t)$ and a linear resource stream $C(t) = C \cdot t$, such that the respective utilizations satisfy the following inequalities for all $s \in \mathbb{R}$:*

$$C \geq \frac{Q}{T} \geq \frac{R(s+S)-R(s)}{S}.$$

Then, all relative deadlines of at least S are met, i.e.

$$\Delta \triangleq \sup_t \inf\{\tau \mid R'(t) \geq R(t-\tau)\} \leq S.$$

Proof. By assumption, we have for all $s \in \mathbb{R}$ that $\frac{R(s+S)-R(s)}{S} \leq C$. From linearity, it follows that for all $U > 0$ and all $u \in \mathbb{R}$ we have $\frac{C(u+U)-C(u)}{U} = C$, so $\frac{R(s+S)-R(s)}{S} \leq \frac{C(u+U)-C(u)}{U}$ and $C(u+T) - C(u) \geq Q$. Using our main theorem, we find for all $U > 0$, that relative deadlines greater than $S + 2 \cdot U$ are met, and hence $\Delta \triangleq \sup_t \inf\{\tau \mid R'(t) \geq R(t-\tau)\} \leq \inf_{U>0}\{S + 2 \cdot U\} = S$. Which concludes our proof.

5 Discussion

As mentioned in the introduction, our reason for making a formal model of the deferrable server strategy, was our suspicion that the schedulability analysis in [2] for real-time tasks under hierarchical fixed-priority preemptive scheduling and a deferrable server, is in general pessimistic rather than exact. In this section, we will discuss this claim in more detail, with an apology for the necessary cluttering in use of terminology. Throughout the paper we have used the terminology that is usual in real-time calculus as much as possible, but in this section we must transliterate some of the results to fit the terminology of [2].

In [2], a comparison is made between deferrable servers and *periodic servers*, which only differ from each other in that a periodic server does not preserve its remaining capacity if resources are provided but tasks are not ready to use them, while a deferrable server does. The comparison is carried out under the assumption that there is a lowest priority, soft real-time task present (block 2 in figure 3), of which the interarrival time is unknown. Davis and Burns show that, under this assumption, the worst-case schedulability of tasks is better for periodic servers. However, we feel their subsequent conclusion that periodic servers dominate deferrable servers is somewhat misleading, because the presence of the soft real-time task severely influences the behavior of the deferrable server, while it does not influence the behavior of the periodic server. More precisely, in the worst case scenario, the buffer of the soft real-time task is never empty. This causes the deferrable server to loose capacity to this task at all times, and in effect behave in the same way as the periodic server.

Using Corollary 1, we will show that a deferrable server can indeed outperform a periodic server when this unknown soft real-time task is not present. The simplest example that shows this, is a system consisting of a single budgeted task, with a deadline equal to its period. In the remainder of this section, we first briefly relate our terminology with the terminology used in [2], and subsequently transliterate and refine Corollary 1 for our example system. Next, we recapitulate worst-case response time analysis given in [2] by presenting a dedicated equation for our special case. Under the aforementioned conditions, the analysis in [2] is exact for deferrable servers, i.e. provides a necessary and sufficient schedulability condition for the task. However, our corollary shows that the analysis is pessimistic for deferrable servers, when the unknown lowest priority task is not present or its interarrival time becomes known. This we illustrate using the aforementioned example.

In [2], a periodic task τ is characterized by a *period* (or *inter-arrival time*) T^τ, a *worst-case computation time* C^τ, and a *relative deadline* D^τ. We assume that the task's period and deadline are equal, i.e. $T^\tau = D^\tau$. A server σ is characterized by a *replenishment period* T^σ and a *capacity* C^σ. Based on these notions, the utilization U^τ of the task is given by $\frac{C^\tau}{T^\tau}$ and the utilization U^σ of the server by $\frac{C^\sigma}{T^\sigma}$.

The task τ can be either *bound* or *unbound*. The task τ is bound if it has a period that is an exact multiple of the server's period and an arrival time that coincides with the replenishment of the server's capacity. Otherwise τ is unbound. We assume an unbound task. Without loss of generality, we assume that the server σ is replenished for the first time at time $\varphi^\sigma = 0$. Moreover, we assume that τ is released for the first time at time $\varphi^\tau \geq 0$, i.e. *at* or *after* the first replenishment of σ. With this terminology in place, we can write

$$\begin{aligned} R(t) &= C^\tau \cdot \left\lceil \frac{t - \varphi^\tau}{T^\tau} \right\rceil \\ C(t) &= t \\ Q &= C^\sigma \\ S &= T^\tau \\ T &= T^\sigma \end{aligned}$$

For our system, we can now transliterate and refine Corollary 1.

Corollary 2. *Consider a highest-priority deferrable server σ with period T^σ and capacity C^σ. Furthermore, assume that the server is associated with a periodic task τ with period T^τ and worst-case computation time C^τ, where the first release of τ takes place at or after the first replenishment of σ. When the respective utilizations satisfy the following inequality*

$$U^\tau \le U^\sigma \le 1,$$

the deadline $D^\tau = T^\tau$ of τ is met.

Note that our transliterated corollary holds for both a bound task and an unbound task. Furthermore, note that (2) is a necessary and sufficient (i.e. exact) schedulability condition for both the task and the server. Finally, note that

$$U^{\mathcal{T}} \le U^\sigma \le 1,$$

is a *necessary* schedulability condition for a set $\mathcal{T}$ of independent hard real-time tasks with utilization $U^{\mathcal{T}}$ with an associated server σ with utilization U^σ

We will now derive a schedulability condition for our system from [2] starting from an equation to determine the task's worst-case response time. The task's *worst-case response time* WR^τ is the longest possible time from its arrival to its completion. Similarly, the server's *worst-case response time* WR^σ is the longest possible time from the server being replenished to its capacity being exhausted, given the task is ready to use all of its capacity. The task is said to be *schedulable* if (and only if)

$$WR^\tau \le D^\tau.$$

Similarly, the server is schedulable if (and only if) $WR^\sigma \le T^\sigma$.

For our system, the server is schedulable when $C^\sigma \le T^\sigma$. Based on [2], we derive for our system that WR^τ is given by

$$WR^\tau = C^\tau + \left\lceil \frac{C^\tau}{C^\sigma} \right\rceil (T^\sigma - C^\sigma), \tag{1}$$

which leads to the following condition for schedulability of a task with a deadline D^τ equal to its period T^τ

$$C^\tau + \left\lceil \frac{C^\tau}{C^\sigma} \right\rceil (T^\sigma - C^\sigma) \le T^\tau. \tag{2}$$

Now, as an example, we fix C^τ and T^τ and plot the minimum utilization U^σ_{min} of the server as a function of T^σ, i.e. we plot

$$U^\sigma_{min}(T^\sigma) = \min\left\{ \frac{C^\sigma}{T^\sigma} \mid T^\tau \ge C^\tau + \left\lceil \frac{C^\tau}{C^\sigma} \right\rceil (T^\sigma - C^\sigma), C^\sigma \ge 0 \right\}.$$

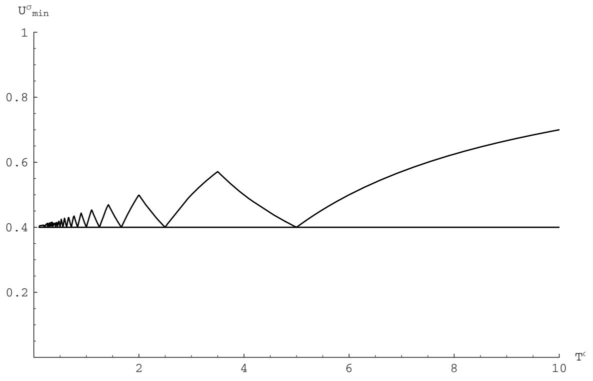

Fig. 5. Minimum server utilization U^{σ}_{min} for $C^{\tau} = 2$ and $T^{\tau} = 5$ as a function of T^{σ}

The result for $C^{\tau} = 2$ and $T^{\tau} = 5$ is depicted in figure 5. The horizontal line in this figure, shows the utilization $U^{\tau} = \frac{C^{\tau}}{T^{\tau}}$ of the task.

The figure illustrates that according to [2] only for values of T^{σ} equal to an integral fraction of T^{τ}, i.e. $T^{\sigma} = \frac{T^{\tau}}{n}$ for $n \in \mathbb{N}^{+}$, the minimum server utilization U^{σ}_{min} needed for schedulability is equal to the task utilization U^{τ}. For other values of T^{σ}, U^{σ}_{min} is higher than U^{τ}. However, according to our theorem, the server utilization U^{σ} may be chosen equal to the task utilization U^{τ} when using a deferrable server, irrespective of T^{σ}. From this example, we conclude that the schedulability condition expressed by (2) is sufficient but not necessary for an unbound task with an associated deferrable server. As a result, equation (1) is pessimistic, and the worst-case response time analysis presented in [2] for unbound tasks with associated deferrable servers is therefore pessimistic, and not exact, when the assumption of an unknown soft real-time lowest priority task is dropped.

According to [2], their worst-case response time analysis is exact for both deferrable servers and periodic servers. Based on that result, they claim that periodic servers dominate deferrable servers with respect to schedulability of tasks, i.e. "*there are no systems ... that can be scheduled using a set of deferrable servers that cannot also be scheduled using an equivalent set of periodic servers*". We have shown here, that this claim heavily relies on the assumed presence of an *unknown* soft real-time lowest priority task, and that it may not hold if such a task is not present, or if we somehow have more information on the interarrival time of this lowest priority task.

6 Conclusive Remarks and Future Work

We presented a formal model of a deferrable server budgeting strategy [10], using the real-time calculus of [11]. Using this model we derived a schedulability theorem stating that a set of tasks with deadlines greater than the sum of the minimum interarrival times of the tasks and the maximum delay between arrival of resources (due to other servers in the network) is schedulable provided that the utilization of the tasks is smaller than the utilization of the server and smaller than the utilization of the arriving resources. Application of this theorem to the special case of a highest-priority server has led to an example where deferrable servers outperform periodic servers in the absence of soft real-time tasks. Hence, the claim in [2] that periodic servers outperform deferrable servers, holds in the presence of soft real-time tasks, but not in their absence. Worst-case response time analysis of deferrable servers therefore still requires further research.

In this paper, we used real-time calculus as an aid to prove our schedulability theorem for deferrable servers. In particular, we have proved the schedulability theorem using direct symbolic manipulation of our model of the deferrable server. Another, indirect way to analyse the behaviour of real-time calculus models, is through so-called *service curves* [4], comparable to analysis using Fourier and Laplace transformations in system theory. Service curve transformations define a system in terms of upper and lower bounds on the input and output streams rather than defining the streams exactly. Initial investigations suggest that service curve transformations can also be found for deferrable servers, but will probably not be *tight*, i.e. they will not give optimal bounds. Our attempts to use service curve transformations to prove our schedulability theorem failed, because the exact timing of replenishments turned out to be crucial in this proof, and is lost during the transformations.

Finally, the treatment of budgeting servers in general does not stop with the treatment of deferrable servers. First of all, there is a great variety of budgeting servers already in use in practice and theory. And secondly, the use of budgeting servers has lead to the introduction of a hierarchical approach to scheduling. If multiple tasks are run on a single server, one may first divide the available resources over the servers, and in a second stage schedule the tasks that run on each server independently of what runs on the other servers. Recent theory about the schedulability of tasks under different kinds of budgeting servers was presented in [2,3,7,9]. In the beginning of section 3 we have shown that there is a connection between the modeling of individual tasks and the modeling of servers. Of course, this concept can be lifted to meta-servers, etc. But, further analysis is still needed to obtain a truly hierarchical approach, in which we first abstract from lower level tasks and analyse the connections between servers, and afterwards analyse the properties of the served tasks.

Acknowledgements. We would like to thank Lothar Thiele of ETH Zürich, and Robert I. Davis and Alan Burns of the University of York, for their valuable feedback on our initial results and for helping us to better understand their respective work.

References

1. Buttazzo, G.C.: Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. Springer Science+Business Media, Inc. (2005)
2. Davis, R.I., Burns, A.: Hierarchical fixed priority pre-emptive scheduling. In: Proc. 26^{th} IEEE Real-Time Systems Symposium (RTSS), pp. 389–398 (December 2005)

3. Deng, Z., Liu, J.W.-S.: Scheduling real-time applications in open environment. In: Proc. 18^{th} IEEE Real-Time Systems Symposium (RTSS), pp. 308–319 (December 1997)
4. Thiran, P., Le Boudec, J.-Y. (eds.): Network Calculus. LNCS, vol. 2050. Springer, Berlin (2001)
5. Joseph, M. (ed.): Real-Time Systems: Specification, Verification and Analysis. Prentice-Hall, Englewood Cliffs (1996)
6. Klein, M.H., Ralya, T., Pollak, B., Obenza, R., González-Harbour, M.: A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems. Kluwer Academic Publishers, Dordrecht (1993)
7. Kuo, T.-W., Li, C.-H.: A fixed-priority-driven open environment for real-time applications. In: Proc. 20^{th} IEEE Real-Time Systems Symposium (RTSS), pp. 256–267 (December 1999)
8. Rajkumar, R., Juvva, K., Molano, A., Oikawa, S.: Resource kernels: A resource-centric approach to real-time and multimedia systems. In: Proc. SPIE, Conference on Multimedia Computing and Networking (CMCN), vol. 3310, pp. 150–164 (January 1998)
9. Saewong, S., Rajkumar, R., Lehocky, J.P., Klein, M.H.: Analysis of hierarchical fixed-priority scheduling. In: Proc. 14^{th} Euromicro Conference on Real-Time Systems (ECRTS), pp. 152–160 (2002)
10. Strosnider, J.K., Lehoczky, J.P., Sha, L.: The deferrable server algorithm for enhanced aperiodic responsiveness in hard real-time environments. IEEE Transactions on Computers 44(1), 73–91 (1995)
11. Thiele, L., Chakraborty, S., Naedele, M.: Real-time calculus for scheduling hard real-time systems. In: Proc. IEEE International Symposium on Circuits and Systems (ISCAS), vol. 4, pp. 101–104 (2000)

A Scheduling Theorem: Proof

To prove $\sup_t \inf\{\tau \mid R'(t) \geq R(t-\tau)\} \leq S + 2 \cdot U$, it is necessary and sufficient to prove that $R'(t) \geq R(t - S - 2 \cdot U)$ for all t. For this, we start from our previously obtained solution for $R'(t)$.

$$\begin{aligned} R'(t) = \inf_{n\in\mathbb{N}} \inf_{u \leq t \wedge T \cdot \lceil \frac{t}{T} - n \rceil} \Bigg\{ & R(u) + C\left(t \wedge T \cdot \left\lceil \frac{t}{T} - n \right\rceil\right) - C(u) \\ & + \sum_{m=0}^{n-1} Q \wedge \left(C\left(t \wedge T \cdot \left\lceil \frac{t}{T} - m \right\rceil\right) - C\left(T \cdot \left\lceil \frac{t}{T} - m - 1 \right\rceil\right)\right)\Bigg\} \end{aligned}$$

We split off the special cases where $n = 0$ and $m = 0$, and find:

$$\begin{aligned} &= \inf_{u \leq t} \{R(u) + C(t) - C(u)\} \\ &\wedge \inf_{n \geq 1} \inf_{u \leq T \cdot \lceil \frac{t}{T} - n \rceil} \Bigg\{ R(u) + C\left(T \cdot \left\lceil \frac{t}{T} - n \right\rceil\right) - C(u) \\ &+ Q \wedge \left(C(t) - C\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right)\right) \\ &+ \sum_{m=1}^{n-1} Q \wedge \left(C\left(T \cdot \left\lceil \frac{t}{T} - m \right\rceil\right) - C\left(T \cdot \left\lceil \frac{t}{T} - m - 1 \right\rceil\right)\right)\Bigg\} \end{aligned}$$

Again, we find two cases, depending on whether Q or $C(t) - C\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right)$ is larger.

$$\begin{aligned}
&= \inf_{u \le t} \{R(u) + C(t) - C(u)\} \\
&\wedge \inf_{n \ge 1} \inf_{u \le T \cdot \left\lceil \frac{t}{T} - n \right\rceil} \Bigg\{ R(u) + C\left(T \cdot \left\lceil \frac{t}{T} - n \right\rceil\right) - C(u) \\
&+ Q + \sum_{m=1}^{n-1} Q \wedge \left(C\left(t \wedge T \cdot \left\lceil \frac{t}{T} - m \right\rceil\right) - C\left(T \cdot \left\lceil \frac{t}{T} - m - 1 \right\rceil\right)\right)\Bigg\} \\
\wedge &\inf_{n \ge 1} \inf_{u \le T \cdot \left\lceil \frac{t}{T} - n \right\rceil} \Bigg\{ R(u) + C(t) - C\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + C\left(T \cdot \left\lceil \frac{t}{T} - n \right\rceil\right) - C(u) \\
+ &\sum_{m=1}^{n-1} Q \wedge \left(C\left(T \cdot \left\lceil \frac{t}{T} - m \right\rceil\right) - C\left(T \cdot \left\lceil \frac{t}{T} - m - 1 \right\rceil\right)\right)\Bigg\}
\end{aligned}$$

Take $u = T \cdot \left\lceil \frac{t}{T} - m - 1 \right\rceil$ in the assumption on resource arrivals to find $C\left(T \cdot \left\lceil \frac{t}{T} - m \right\rceil\right) - C\left(T \cdot \left\lceil \frac{t}{T} - m - 1 \right\rceil\right) \ge Q$.

$$\begin{aligned}
&= \inf_{u \le t} \{R(u) + C(t) - C(u)\} \\
&\wedge \inf_{n \ge 1} \inf_{u \le T \cdot \left\lceil \frac{t}{T} - n \right\rceil} \left\{ R(u) + C\left(T \cdot \left\lceil \frac{t}{T} - n \right\rceil\right) - C(u) + n \cdot Q \right\} \\
&\wedge \inf_{n \ge 1} \inf_{u \le T \cdot \left\lceil \frac{t}{T} - n \right\rceil} \Bigg\{ R(u) + C(t) - C\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) \\
&+ C\left(T \cdot \left\lceil \frac{t}{T} - n \right\rceil\right) - C(u) + (n-1) \cdot Q \Bigg\}
\end{aligned}$$

Then, we truncate the argument of $C(.)$ to a multiple of U with a convenient remainder. Using monotonicity of $C(t)$ we find:

$$\begin{aligned}
&\ge \inf_{u \le t} \left\{ R(u) + C\left(\left\lfloor \frac{t-u}{U} \right\rfloor \cdot U + u \right) - C(u) \right\} \\
&\wedge \inf_{n \ge 1} \inf_{u \le T \cdot \left\lceil \frac{t}{T} - n \right\rceil} \left\{ R(u) + C\left(\left\lfloor \frac{T \cdot \left\lceil \frac{t}{T} - n \right\rceil - u}{U} \right\rfloor \cdot U + u \right) - C(u) + n \cdot Q \right\} \\
&\wedge \inf_{n \ge 1} \inf_{u \le T \cdot \left\lceil \frac{t}{T} - n \right\rceil} \Bigg\{ R(u) + C\left(\left\lfloor \frac{t - T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil}{U} \right\rfloor \cdot U + T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil \right) \\
&- C\left(T \cdot \left\lceil \frac{t}{T} - 1 \right\rceil\right) + C\left(\left\lfloor \frac{T \cdot \left\lceil \frac{t}{T} - n \right\rceil - u}{U} \right\rfloor \cdot U + u \right) - C(u) + (n-1) \cdot Q \Bigg\}
\end{aligned}$$

Now, we define X as a lower bound on the utilization of $R(t)$, and find:

$$\inf_{u} \left\{ \frac{C(u+U) - C(u)}{U} \right\} \ge \sup_{s} \left\{ \frac{R(s+S) - R(s)}{S} \right\} \triangleq X.$$

Using X, we eliminate $C(t)$ and Q.

$$\begin{aligned}
&\geq \inf_{u \leq t} \left\{ R(u) + \left\lfloor \frac{t-u}{U} \right\rfloor \cdot U \cdot X \right\} \\
&\wedge \inf_{n \geq 1} \inf_{u \leq T \cdot \lceil \frac{t}{T} - n \rceil} \left\{ R(u) + \left\lfloor \frac{T \cdot \lceil \frac{t}{T} - n \rceil - u}{U} \right\rfloor \cdot U \cdot X + n \cdot T \cdot X \right\} \\
&\wedge \inf_{n \geq 1} \inf_{u \leq T \cdot \lceil \frac{t}{T} - n \rceil} \left\{ R(u) + \left\lfloor \frac{t - T \cdot \lceil \frac{t}{T} - 1 \rceil}{U} \right\rfloor \cdot U \cdot X \right. \\
&\left. + \left\lfloor \frac{T \cdot \lceil \frac{t}{T} - n \rceil - u}{U} \right\rfloor \cdot U \cdot X + (n-1) \cdot T \cdot X \right\}
\end{aligned}$$

Observe that for all x we have $x \cdot X \geq \lfloor \frac{x}{S} \rfloor \cdot S \cdot X$. And since $S \cdot X$ serves as an upper bound on $R(s+S) - R(s)$, this gives us

$$\begin{aligned}
&\geq \inf_{u \leq t} \left\{ R\left(u + \left\lfloor \left\lfloor \frac{t-u}{U} \right\rfloor \cdot \frac{U}{S} \right\rfloor \cdot S \right) \right\} \\
&\wedge \inf_{n \geq 1} \inf_{u \leq T \cdot \lceil \frac{t}{T} - n \rceil} \left\{ R\left(u + \left\lfloor \left\lfloor \frac{T \cdot \lceil \frac{t}{T} - n \rceil - u}{U} \right\rfloor \cdot \frac{U}{S} + n \cdot \frac{T}{S} \right\rfloor \cdot S \right) \right\} \\
&\wedge \inf_{n \geq 1} \inf_{u \leq T \cdot \lceil \frac{t}{T} - n \rceil} \\
&\quad \left\{ R\left(u + \left\lfloor \left\lfloor \frac{t - T \cdot \lceil \frac{t}{T} - 1 \rceil}{U} \right\rfloor \cdot \frac{U}{S} + \left\lfloor \frac{T \cdot \lceil \frac{t}{T} - n \rceil - u}{U} \right\rfloor \cdot \frac{U}{S} + (n-1) \cdot \frac{T}{S} \right\rfloor \cdot S \right) \right\}
\end{aligned}$$

And using monotonicity of $R(t)$, together with the observation that $\lfloor x \rfloor \cdot y \geq x \cdot y - y$ we find:

$$\begin{aligned}
&\geq \inf_{u \leq t} \left\{ R\left(u + \left\lfloor \frac{t-u}{U} \right\rfloor \cdot U - S \right) \right\} \\
&\wedge \inf_{n \geq 1} \inf_{u \leq T \cdot \lceil \frac{t}{T} - n \rceil} \left\{ R\left(u + \left\lfloor \frac{T \cdot \lceil \frac{t}{T} - n \rceil - u}{U} \right\rfloor \cdot U + n \cdot T - S \right) \right\} \\
&\wedge \inf_{n \geq 1} \inf_{u \leq T \cdot \lceil \frac{t}{T} - n \rceil} \\
&\quad \left\{ R\left(u + \left\lfloor \frac{t - T \cdot \lceil \frac{t}{T} - 1 \rceil}{U} \right\rfloor \cdot U + \left\lfloor \frac{T \cdot \lceil \frac{t}{T} - n \rceil - u}{U} \right\rfloor \cdot U + (n-1) \cdot T - S \right) \right\} \\
&\geq \inf_{u \leq t} \{ R(u + t - u - U - S) \} \\
&\wedge \inf_{n \geq 1} \inf_{u \leq T \cdot \lceil \frac{t}{T} - n \rceil} \left\{ R\left(u + T \cdot \left\lceil \frac{t}{T} - n \right\rceil - u - U + n \cdot T - S \right) \right\} \\
&\wedge \inf_{n \geq 1} \inf_{u \leq T \cdot \lceil \frac{t}{T} - n \rceil}
\end{aligned}$$

$$\left\{R\left(u+t-T\cdot\left\lceil\frac{t}{T}-1\right\rceil-U+T\cdot\left\lceil\frac{t}{T}-n\right\rceil-u-U+(n-1)\cdot T-S\right)\right\}$$

$$\begin{aligned}
&= R\left(t-U-S\right)\\
&\wedge\; R\left(T\cdot\left\lceil\frac{t}{T}\right\rceil-U-S\right)\\
&\wedge\; R\left(t-2\cdot U-S\right)\\
&= R\left(t-2\cdot U-S\right)
\end{aligned}$$

Which concludes our proof.