Journal never presented on conference: Hard Real-Time Guarantees in Feedback-based Resource Reservations

Alessandro V. Papadopoulos, *Mälardalen University, Sweden*
Martina Maggio, *Lund University, Sweden*
Alberto Leva, *Politecnico di Milano, Italy*
Enrico Bini, *University of Turin, Italy*

1. Introduction

Resource reservation servers [1], [2] provide application *isolation*, enforcing the periodic allocation of a resource budget — mainly used to control CPU allocation. Usually, when the budget is exhausted, servers release the CPU. However, some circumstances may prevent the immediate CPU release. This article proposes the use of feedback to compensate for run-time budget variations. As a result of the use of feedback, the system is capable of allocating a target budget also in the presence of runtime circumstances that were originally unaccounted for.

The use of feedback in resource management problems was also investigated by Stankovic et al. [3], Cervin and Eker [4], Lu et al. [5], and Abeni et al. [6], among others. In these works, feedback was used to adapt the resource allocation to a time-varying workload with soft real-time constraints (the typical application was multimedia). In this paper, instead, we assume that the load generated by the applications is known and has hard deadlines. Feedback is used to compensate for run-time events, which may induce deviations with respect to the target budget allocation.

The journal article [7] illustrates the Self-Adaptive Server (SAS). Its analysis allows us to offer hard real-time guarantees, using the concept of "supply bound functions" [8], [9], [10].

2. Supply bound functions

The *supply bound function* $\mathsf{sbf}(t)$ abstraction is a convenient way to model the minimum amount of resource provided by resource reservation servers in any interval of length t. Let $s : \mathbb{R} \to \{0, 1\}$ be the indicator function of any resource allocation over time. The $\mathsf{sbf}(t)$ is such that

$$\forall t_0, t, \quad \mathsf{sbf}(t) \leq \int_{t_0}^{t_0+t} s(\tau)\ d\tau. \tag{1}$$

This means that at least $\mathsf{sbf}(t)$ resource is available to the application in any interval of length t.

A convenient model of a server schedule is given by a sequence of supply intervals, interleaved with a sequence of idle intervals. The lengths of the supply intervals are represented by the sequence $\{S(k)\}_{k=1,2,\ldots}$, while lengths of idle intervals are represented by the sequence $\{Z(k)\}_{k=1,2,\ldots}$. Without loss of generality, we set the time $t = 0$ at the instant when the first resource supply $S(1)$ starts. Figure 1 illustrates the resource provisioning over time according to this notation.

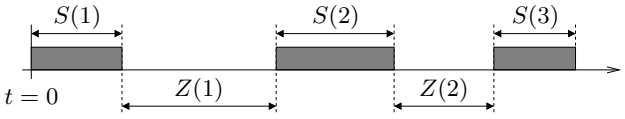

Figure 1. Budget provisioning in servers.

The derivation of a valid supply function $\mathsf{sbf}(t)$ satisfying (1) is made by standard techniques [8]. The following Lemma applies well known results adapting them to the introduced notation of $S(k)$ and $Z(k)$.

Lemma 1 (Lemma 1 in [7]). A server characterized by a sequence of supply intervals of length $\{S(k)\}_{k=1,2,\ldots}$ and idle intervals of length $\{Z(k)\}_{k=1,2,\ldots}$ has the following supply bound function

$$\mathsf{sbf}(t) = \min\left\{t - \sigma_Z(n), \sigma_S(n)\right\}, \quad t \in I_n, n \in \mathbb{N} \tag{2}$$

with the sequence of intervals $\{I_n\}_{n\in\mathbb{N}}$ defined as

$$I_n = \begin{cases} \big[0, \sigma_Z(1)\big) & n = 0 \\ \big[\sigma_Z(n)+\sigma_S(n-1), \sigma_Z(n+1)+\sigma_S(n)\big) & n \geq 1 \end{cases} \tag{3}$$

and with

$$\sigma_S(n) = \inf_{n_0} \sum_{k=n_0}^{n_0+n-1} S(k), \quad \sigma_Z(n) = \sup_{n_0} \sum_{k=n_0}^{n_0+n-1} Z(k), \tag{4}$$

properly extended at $n = 0$ with $\sigma_S(0) = \sigma_Z(0) = 0$.

The $\mathsf{sbf}(t)$ of (2) is illustrated in Figure 2. On top, the intervals I_n are drawn.

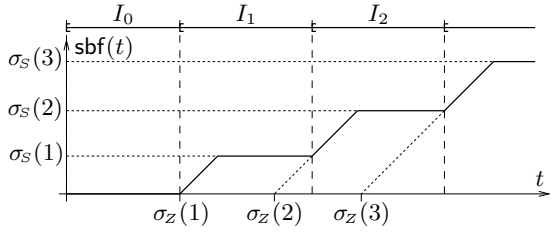

Figure 2. An example of supply bound function.

The expression of $\mathsf{sbf}(t)$ in (2) generalizes other resource models. For example, by setting the minimum sum of n consecutive budgets as $\sigma_S(n) = nQ$ and the maximum sum of n consecutive idle intervals as $\sigma_Z(n) = n(P-Q)+D-Q$, the resulting $\mathsf{sbf}(t)$ of (2) is equivalent to the supply function of the EDP resource model [11] with budget Q, period P, and deadline D.

3. The Self-Adaptive Server (SAS)

In an ideal periodic server, a budget $\bar{Q}$ is allocated every period $\bar{P}$. The real allocation may differ from the ideal one, due to disturbances caused by: (i) the usage of shared resources, preventing the processor release when the budget is expired; (ii) an application self-suspending earlier than the budget completion; (iii) synchronization with I/O event; (iv) the presence of a system tick, forcing scheduling events to occur at predetermined instants; and many other causes. These disturbances disrupt the ideal behavior of the periodic server, undermining real-time guarantees.

We compensate for these variations with feedback, proposing the Self-Adaptive Server (SAS) [7]. A SAS server aims to provide a budget $\bar{Q}$, every period $\bar{P}$. At every round, the SAS server allocates resource non-preemptively. At the k-th activation round the actual amount of service time $S(k)$ may differ from the set value for the desired budget allocation $Q(k)$. We denote the difference — i.e., the *disturbance* — by $\varepsilon_S(k) = S(k) - Q(k)$. $S(k)$ is therefore both the service time and the length of the supply interval during k-th round. Figure 3 shows the control logic of a SAS server.

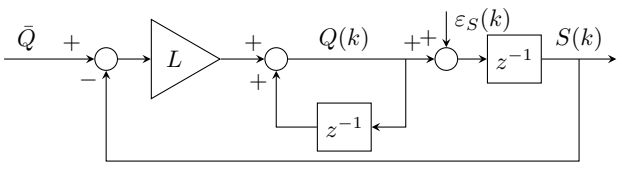

Figure 3. Block diagram of the controller of SAS servers.

The logic is on purpose very simple, to avoid consuming too much computation time in executing the control logic itself, and boils down to the following equations.

$$S(0) = Q(0) = \bar{Q}, \tag{5}$$

$$S(k+1) = Q(k) + \varepsilon_S(k), \tag{6}$$

$$Q(k+1) = Q(k) + L(\bar{Q} - S(k)). \tag{7}$$

In absence of disturbances — when $\varepsilon_S(k) = 0$ — the allocated budget $S(k)$ is constantly equal to the target value $\bar{Q}$, as desired. When disturbances occur, the controller gain L adjusts the budget in response to deviations from the target value $\bar{Q}$. If $L \in (0, 1)$ the controlled system is stable and capable of rejecting constant disturbances [7].

A condition of interest is when N SAS servers coexist and are scheduled in loop, as shown in Figure 4. By assuming, that the first server is the one under analysis, while the servers from the second to the N-th are the "adversaries",

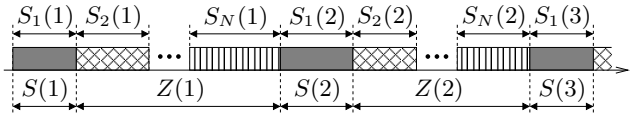

Figure 4. N servers in loop.

i.e., the time allocated to them is the idle time for the first server, then the intervals $Z(k)$ of idle time are exactly the sum of the $N-1$ server budgets. The hypothesis of serving N servers in loop implies that all the N servers share a common period $\bar{P}$. The journal article [7] demonstrates that the dynamics of the N-servers case is analogous to the one of (5)–(7) with a suitable replacement of the involved variables. Next, we show that it is possible to offer real-time guarantees on the resource allocation by means of the supply bound function.

4. Supply function of SAS servers

The definition of $\mathsf{sbf}(t)$ of Lemma 1 depends on the values of $\sigma_S(n)$ and $\sigma_Z(n)$ of (4). However, we need to clarify how to compute the expressions of (4) for SAS servers. First, disturbances $\varepsilon_S(k)$ and $\varepsilon_Z(k) := \sum_{i=2}^{N} \varepsilon_i(k)$ must be bounded, a bound often being easy to derive.

$$\forall k \in \mathbb{N}, \qquad |\varepsilon_S(k)| \le \bar{\varepsilon}_S, \quad |\varepsilon_Z(k)| \le \bar{\varepsilon}_Z. \tag{8}$$

If disturbances are not bounded, it is not possible to guarantee any service time, with any resource allocation policy. With bounded disturbances, the quantities $\sigma_S(n)$ and $\sigma_Z(n)$, necessary to define the supply function from (2), are:

$$\begin{aligned} \sigma_S(n) &= \inf_{|\varepsilon_S(k)| \le \bar{\varepsilon}_S, n_0} \sum_{k=n_0}^{n_0+n-1} S(k), \\ \sigma_Z(n) &= \sup_{|\varepsilon_Z(k)| \le \bar{\varepsilon}_Z, n_0} \sum_{k=n_0}^{n_0+n-1} Z(k). \end{aligned} \tag{9}$$

The controller governing both the supply and idle intervals is a linear time-invariant (LTI) system. The linearity of the dynamics and the bounds of (8) enables to find $\sigma_S(n)$ and $\sigma_Z(n)$, which are the key ingredients for computing the $\mathsf{sbf}(t)$ via Lemma 1.

Theorem 2 (Theorem 1 in [7]). Given a SAS server with controller gain L, let $g(k)$ be the response to step disturbance, and let the disturbances $\varepsilon_S(k)$ and $\varepsilon_Z(k)$ be bounded by $\bar{\varepsilon}_S$ and $\bar{\varepsilon}_Z$, respectively, as in (8). Then:

$$\sigma_S(n) = n\bar{Q} - \bar{\varepsilon}_S \mathcal{N}(n, L) \tag{10}$$

$$\sigma_Z(n) = n(\bar{P} - \bar{Q}) + \bar{\varepsilon}_Z \mathcal{N}(n, L), \tag{11}$$

with

$$\mathcal{N}(n, L) = \sum_{k=0}^{\infty} |g(k) - g(k-n)|. \tag{12}$$

The combination of Lemma 1 with the expressions of $\sigma_S(n)$ and $\sigma_Z(n)$ of Theorem 2 allows the derivation of $\mathsf{sbf}(t)$ as a

function of the gain L of the controller. The relationship between the controller gain L of SAS servers and the delivered supply function is given through the expression $\mathcal{N}(n, L)$ of (12). In Figure 5, we plot $\mathcal{N}(n, L)$ for the values of $L \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, L^\star\}$, with $L^\star = \frac{3-\sqrt{5}}{2}$, while in Figure 6 the supply function $\mathsf{sbf}(t)$, as characterized in Lemma 1, is drawn for $L \in \{0, \frac{1}{4}, \frac{3}{4}, L^\star\}$.

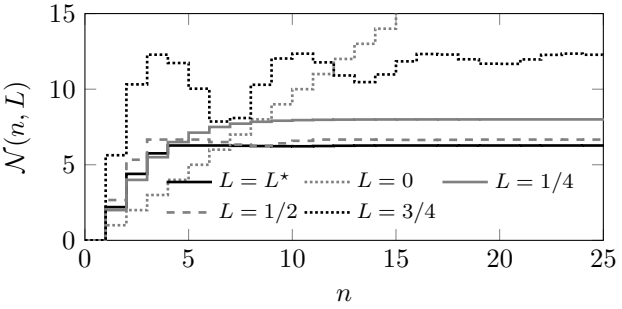

Figure 5. Value of $\mathcal{N}(n, L)$, for some L.

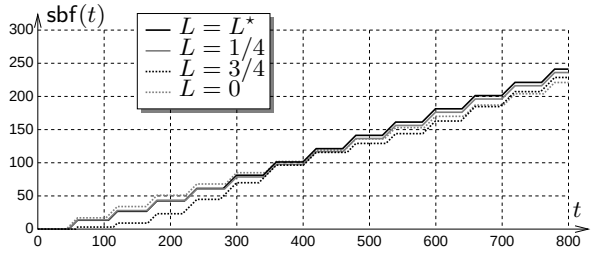

Figure 6. Supply function of SAS servers when $P = 60$, $Q = 20$, and $\bar{\varepsilon}_S = \bar{\varepsilon}_Z = 3$.

Observing Figure 6, for small values of t the supply function $\mathsf{sbf}(t)$ is larger when $L = 0$, which corresponds not to compensate for disturbances. However, such a choice is not capable to asymptotically guarantee the target bandwidth of $\bar{Q}/\bar{P}$. Instead, any controller with $L \in (0, 1)$ can guarantee the asymptotic bandwidth of $\bar{Q}/\bar{P}$. Among these values, the choice of $L = L^\star = \frac{3-\sqrt{5}}{2} \approx 0.38197$, it is the optimal value since it achieves the asymptotically largest possible supply bound function $\mathsf{sbf}(t)$ [7].

Finally, we find the condition which guarantees that all budgets are always non-negative. In fact, as it can be observed in (10), for large $\bar{\varepsilon}_S$ the supply function can indeed be negative, meaning that the necessary compensation may exceed the budget. Next Lemma establishes an upper bound on the maximum controllable disturbance.

Lemma 3 (Lemma 3 in [7]). If the disturbance $\bar{\varepsilon}_S$ and the budget Q are such that

$$\frac{\bar{\varepsilon}_S}{Q} \leq \frac{1}{\mathcal{N}(1, L)}, \tag{13}$$

then it is always $\sigma_S(n) \geq 0$.

Previous Lemma is a feasibility condition: if condition (13) does not hold, then the control policy of SAS servers may require a negative (infeasible) server budget. In Figure 7 we plot $1/\mathcal{N}(n,L)$ as function of L. If the maximum disturbance $\bar{\varepsilon}_S$ which needs to be compensated is larger than half of the target budget Q, then SAS servers are not suited.

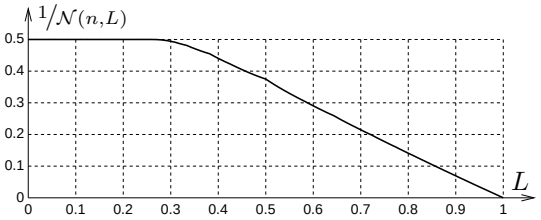

Figure 7. Maximum disturbance handled by SAS servers.

In conclusion, we can assert that the controller gain which maximizes the linear lower bound of the $\mathsf{sbf}(t)$ is $L^\star = \frac{3-\sqrt{5}}{2}$. Although the linear lower bound is used to drive the design of the SAS server, the exact $\mathsf{sbf}(t)$ of (2) can be used to guarantee real-time tasks running within the SAS server. Hence, the guarantee test does not suffer from the typical approximation error of the linear lower bound.

References

[1] C. W. Mercer, S. Savage, and H. Tokuda, "Processor capacity reserves: An abstraction for managing processor usage," in *Proceedings of the 4th Workshop on Workstation Operating Systems*. IEEE, 1993, pp. 129–134.

[2] L. Abeni and G. Buttazzo, "Integrating multimedia applications in hard real-time systems," in *Proceedings of the 19th IEEE Real-Time Systems Symposium*, Madrid, Spain, Dec. 1998, pp. 4–13.

[3] J. A. Stankovic, C. Lu, and S. H. Son, "The case for feedback control in real-time scheduling," in *Proceedings of the Euromicro Conference on Real-Time*, York, U.K., Jun. 1999.

[4] A. Cervin and J. Eker, "Feedback scheduling of control tasks," in *Proceedings of the 39th IEEE Conference on Decision and Control*, 2000, pp. 4871–4876.

[5] C. Lu, T. F. Abdelzaber, J. A. Stankovic, and S. H. Son, "A feedback control approach for guaranteeing relative delays in web servers," in *Proceedings of the 7th IEEE Real-Time Technology and Applications Symposium*, 2001, pp. 51–62.

[6] L. Abeni, L. Palopoli, G. Lipari, and J. Walpole, "Analysis of a reservation-based feedback scheduler," in *Proceedings of the 23rd IEEE Real-Time Systems Symposium*, Austix (TX), USA, Dec. 2002, pp. 71–80.

[7] A. V. Papadopoulos, M. Maggio, A. Leva, and E. Bini, "Hard real-time guarantees in feedback-based resource reservations," *Real-Time Systems*, vol. 51, no. 3, pp. 221–246, Jun. 2015.

[8] A. K. Mok, X. Feng, and D. Chen, "Resource partition for real-time systems," in *Proceedings of the 7th IEEE Real-Time Technology and Applications Symposium*, Taipei, Taiwan, May 2001, pp. 75–84.

[9] G. Lipari and E. Bini, "Resource partitioning among real-time applications," in *Proceedings of the 15-th Euromicro Conference on Real-Time Systems*, Porto, Portugal, Jul. 2003, pp. 151–158.

[10] I. Shin and I. Lee, "Periodic resource model for compositional real-time guarantees," in *Proceedings of the 24th Real-Time Systems Symposium*, Cancun, Mexico, Dec. 2003, pp. 2–13.

[11] A. Easwaran, M. Anand, and I. Lee, "Compositional analysis framework using EDP resource models," in *Proceedings of the 28th IEEE International Real-Time Systems Symposium*. Tucson, AZ, USA: IEEE Computer Society, 2007, pp. 129–138.