Feedback-Based Adaptive Resource Control in QoS-Aware SOA Systems with Soft Real-Time Requirements

Francisco José Monaco and Pedro Northon Nobile

University of São Paulo,
Department of Computer Systems, ICMC
Av. Trab. Saocarlense, 400, São Carlos, SP,
CEP 13560-970, Cx.Po 668, Brazil
`monaco@icmc.usp.br`

Abstract. When deployed as operational components of production systems, novel computer services are supposed to respond synchronously to real-world events associated to the business process they implement, thereby needing to meet temporal constraints dictated by the dynamics of the environment in which they operate. This elicits a real-time system approach. One emerging concept to cope with such unpredictability of large-scale distributed computer applications is the use of feedback control principles. This paper introduces a feedback-based adaptive resource control algorithm for composite applications implementing real-time business process. The study is based on recent achievements in the field and ongoing progresses. A brief background on the field, the rationales of the proposed techniques and development results are presented.

Keywords: QoS, Real-Time, SOA, Control.

1 Introduction

Continuous technological advancements give rise to the dissemination of networked computer systems throughout our physical environment. From personal and house appliances like smartphones and digital assistants to mission-critical systems such as vehicular equipment and medical instrumentation, distributed computer application become commonplace in our ordinary routine. The more the applications supported by those systems are integrated into the implementation of on-line services upon which we rely for daily activities, the more relevant become their requisites of performance and dependability.

Unlike the asynchronous transactions characteristic of e-mail system, regular file transfer and other conventional Internet applications, those novel computer services are supposed to respond synchronously to real-world events associated to the business process they implement, and therefore need to meet temporal constraints dictated by the environment in which they operate. On this very need lies, in turn, the concept of real-time systems, whose specification poses

N. Bartolini et al. (Eds.): QShine/AAA-IDEA 2009, LNICST 22, pp. 799–810, 2009.

requirements on the expected response times, usually stated in the form of delay upper bounds. In this extent, when deployed as operational components of production systems, on-line services supporting security surveillance systems, computer-supported collaborative tools, mobile context-aware applications and even auctions management engines in an e-commerce system, to name a few examples, follow into this category, inasmuch as they need to timely respond to events that occur with respect to the natural (outer ambient) time, and thereby fulfill temporal responsiveness constraints.

In this field, an important paradigm gaining relevance in the design of large-scale distributed applications is that of service-oriented architecture (SOA). Based on the composition of complex application out of independently developed, loosely coupled, component services, the SOA approach relies on the combination of autonomous building blocks that interact to provide the desired functionality. This is achieved though the coordination of individual services to make up a composite service by means of asynchronous communication over an agreed implementation-independent protocol.

Most of the classical theory on real-time systems, nonetheless, arises from the domain of automation engineering, where the deterministic timing characteristics of industrial processes have made it possible the development of analytical techniques for the design and verification of mechanisms meant to operate under strict constraints. Contrasts with this scenario that of the typical infrastructure of todays interactive computer systems. The stochastic load patterns that applies to both the complex interactions of software and hardware resources in the system host, and the time-varying routing conditions of the network connecting them, yield a poorly predictable environment. In addition, unlike the typical periodic behavior of former automation systems, interactive computer services are inherently driven by event-oriented dynamics. Synchronization and fault-tolerance schemes in such asynchronous distributed systems become exceedingly more complex. As result, ensuring quality of service (QoS) in terms of performance requisites is considerably difficult in SOA systems. The extension of methodological and technical results from the real-time theory to address the non-deterministic features of interactive services is a research-demanding area [1].

The herein reported research work is aligned under this perspective focusing on the challenges of novel QoS-aware service architectures with performance requirements. This paper introduces a feedback-based adaptive resource control algorithm for large-scale composite applications implementing real-time business process. The study is based on recent achievements in the field and ongoing progresses. A brief background on the field, the rationales of the proposed techniques and development results comprise the subject of the next sections.

2 Background

In an intuitive sense, providing quality of service means fulfilling application requirements related to user-perceivable service effectiveness — being the user, in SOA context, either a human or a system component implementing another service. A quality metric thus may be any performance or dependability parameter with meaningful value for the focused application such as throughput, latency, reliability or security requirements. Therefore, while requisites of real-time traffic for multimedia transmission on the Internet has been one major demand for the development of new resource allocation techniques and routing protocols for QoS provision at the network level, it is rather interesting that in an integrated mechanism the upper layers of the architecture cooperate with the policies implemented by the lower ones — an argument which has recently attracted attentions as a key feature for the deployment of effective computer services [2, 3, 4, 5].

2.1 Challenges in Real-Time SOA

Among different parameters through which QoS can be formulated, responsiveness is a key-metric for interactive systems which gains increasing importance as the deployment of on-line services takes place in the control of business processes supported by the network infrastructure. More than influencing subjective user experience, responsiveness becomes a critical parameter when we move from time-independent data transfer to pervasive interactive applications. In this scenario, methodologies for design and evaluation of QoS-aware SOA elicits a real-time system approach.

An application designed with the assumption that any deadline miss implies in a service failure is known as a hard real-time (hard-RT) system. Associated to temporally non-deterministic infrastructures and workload conditions, however, typical large-scale SOA implementations do not lend themselves to such hard response-time requirements at a viable cost, unless under unrealistic worst-case assumptions. On the other hand, if a limited fraction of deadline misses does not imply in failure but, instead, in service degradation, the hard-RT specification can be relaxed. Real-time systems that can tolerate sporadic deadline misses have gained increasing importance in a context where the capability of meeting temporal requirements is associated to the QoS concept.

For this class of applications, stochastic responsiveness constraints are not only more technically and economically feasible, but also effectively meaningful regarding the needs of a wide range of application.

2.2 Average Performance Guarantees

Concerning services with non-rigid real-time requisites, the conventional metric to assess the delivered quality is the deadline miss ratio (DMR). In its simplest form, the dependability requirement may be stated as a DMR upper bound, which suffices to quantify the system reliability with respect to service failure rate, as well as the efficiency (effective throughput) regarding reissued requests (e.g. package retransmissions in a reliable communication mechanism). It does not, however, measure the distribution of delays in time. For the cases in which this is relevant (as in audio streaming of packet-switched networks), DMR has been extended to consider either fixed [6] or sliding windows [7]. The $(m, k) - firm$ [8], the *skip factor* [9] constraints comprise known examples. A more elaborated alternative based on Markov chains has also been recently introduced [10].

Those metrics are suited to firm real-time (firm-RT) systems, namely those in which requests whose deadlines are missed become useless, and are thus discarded (e.g. over-delayed live multimedia frames that should not be played if overdue). On the other hand, if tardy requests have degraded — rather than null — utility, and therefore should not be discarded, soft real-time (soft-RT) constraints are said to apply. This is a plausible scenarios for real-time SOA applications where, for a good consumer experience, the service delivering is expected to exhibit an upper-bounded average response time. It may sporadically take longer than that, but not so frequently that costumers need to wait for too long before being served. One subtle difference in this case is that consumers are not "droppable"; they just queue up in a line and remain there for their turn.

For soft-RT systems, a service-level agreement (SLA) based on DMR, or even windowed DMR, is not a key metric. Surely one wants to know how often unsatisfied clients will eventually timeout or give up from being served. Nonetheless, even if no one declines the transaction, user-perceivable quality of service can also be measured by how long costumers stand in the line.

An alternative metric which relates the service times to their occurrence frequency into a single measure is the average response time (ART) [11]. As a SLA parameter, the average response time (ART) can be meaningful in many circumstances where constraints on aggregate performance metrics are relevant. This is the case, for example, of an online soft-RT system with a finite buffer. It may not be necessary that the service consume queued requests at a constant rate implying in a hard-RT operation; in this case it suffices that the average response time be upper-bounded by a value which prevent the buffer from being emptied during the processes. This is valid for SOA systems whose semantic of the operations in the real-time business process implies soft-RT constraints. ART-based SLA models have been successful exploited in research works on SOA QoS provision.

2.3 Feedback Scheduling

A substantial deal of work in the area of real-time systems has produced important theoretical results aimed at the analysis of temporally deterministic applications. Processes characterized by periodic dynamics — commonly present in monitoring and control systems, as in industrial environment and vehicular automation — have motivated the development of sound analytical approaches. The treatment of event-driven (asynchronous) systems, where requests exhibit non-deterministic inter-arrival and execution time, is otherwise considerably more complex. Real-time resource allocation under non-deterministic dynamics is admittedly challenging even with state-of-art techniques. Heuristic approaches, instead, prevails in this domain.

Alternatively, one concept that has been emerging in the field of real-time computing is the *Feedback Control* notion, already establish in other areas of Engineering. It grounds on the principle of self-adaptation by using the deviation of the system's output from the desired value as an input in such a way to force the system output contrarily to the deviation itself. This called a negative closed loop, as illustrated in Figure 1.

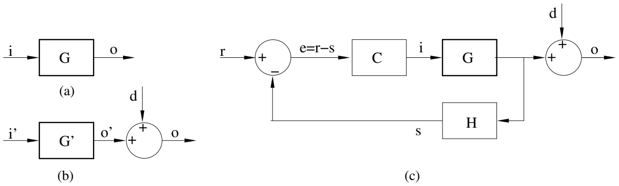


Fig. 1. Feedback control loop

In the diagram, a block represents a generic function $F(x(t))$ that converts the input signal $x(t)$ into an output signal $y(t)$. The controlled system is represented by the block G; i and o denote the input and output respectively.

To set the output o at the desired value it is necessary to adjust the input i accordingly. In open loop control, Figure 1 (a), this has to be accomplished manually. If either some internal parameters of the system changes over time (changing G to G') or an external disturbance (d) affects the output, i must be re-adjusted to compensate for those changes. In the closed loop approach, Figure 1 (b), the block H is a sensor which measures the system output o and produce a corresponding signal s. By applying the control system a reference signal r equivalent to that assumed by s when the output has the desired value, the difference $e = r - s$ is the "error", or the instant deviation between the reference and the current output. This error signal is then inject into the controller block C so as to produce the signal $i = C(e)$, which is injected into the system G, forcing the output to the desired value. To see that, it suffices to suppose that the both controller and the sensor are proportional blocks $C(x) = P.x$ and $H(x) = Q.x$, so that $i = P.e$ and $s = Q.o$. If the system is currently delivering the correct output, than i must have the corresponding correct value. Any undesired increase in o, due to either a change in G or an external disturbance, will cause s to increase proportionally. This raise in s will, in turn, reflects into a decrease in e, since r is a constant reference signal. The system input i is also reduced, having as effect a corresponding decreasing in o. As it can be concluded, an increase in o, in the negative closed loop, forces a decrease in o. Conversely, a decrease in o, has the opposite effect. The system is always driven towards an output value that corresponds to the reference signal, and is is thus much less susceptible to both internal parameter variations and external disturbances. Functions other than the bare proportional gain can be used as the controller block in order to shape the system's output as desired, and these include integrals and derivatives of the error time-function.

Control Theory has developed as a groundbreaking field in Engineering and in some natural science branches, and counts on a rich collection of mathematical modeling tools to describe the behavior of dynamic systems in terms of how they respond to different stimuli, and to verify stability, observability and controlability properties. It offers an extensive background for the design of control strategies applied to linear and non-linear systems.

The application of Control Theory in computer systems architectures for either performance or dependability optimization is nevertheless recent, and considerably less explored than in other domains. It has been employed, for instance, for dynamical clusterization of parallel processors in resource partitioning problems, and for throughput shaping in congestion control mechanisms. *Feedback scheduling* is considered a leading paradigm in real-time systems field [12,13,14,15], specially in applications meant for non-deterministic environments. In this case, runtime automatic parameter-tuning capability represents an advantageous alternative to the off-line presetting at design phase which may be only possible under over-pessimistic worst-case assumptions.

In the context of large-scale service-oriented architectures, self-adaptation constitutes an appealing concept. The simultaneous fulfillment of both the functional requirements concerning the overall business logic and the non-functional requirements concerning the QoS levels expected from the SOA system is a challenging goal in view of the stochastic dynamic of networked computer systems. Automatic control is a theoretically-grounded, well-established technique to cope with such unpredictability by means of closed-loop feedback, which has proven effective for building systems less susceptible to both outer and inner disturbances in other fields of Engineering. The exploitation of control theory principles in the design of self-adaptive QoS-aware SOA applications is a key approach towards the fulfillment of non-functional requirements in large distributed systems.

2.4 Scheduling for ART-Constraints

One important element influencing the performance of interactive services is the scheduler. In order to meet real-time constraints, appropriate scheduling policies should be applied to conveniently manage resource allocation.

For a set of service classes, let the goal be that of ensuring that the average response time calculated over a window of w past requests delivered to each system client be upper-bounded by a value agreed on a per-class basis. Intuitively, if one particular client is recurrently left aside in several consecutive scheduling cycles, the effective ART offered to it will tend to increase. One sensible approach is then to take the difference between both contracted and effective ART into consideration while assigning priorities to the resource allocation.

Classic results from the real-time theory have produced well-known scheduling algorithms [13], among which, the *earliest deadline first* (EDF) is an outstanding contribution. EDF is an optimum scheduling discipline for non-preemptive uniprocessors that assigns the highest priorities to the jobs with the shortest deadlines. It might seem that this urgency-based heuristic is a straightforward solution for the highlighted problem, and that serving first those clients whose contracts are closer to a violation is reasonable.

A more careful theoretical exam under the light of the control theory, nevertheless, will clearly reveal that this is not the case. If the difference between contracted and effective ART is injected into the system as a feedback signal, and is the only factor influencing the scheduling decision, then this difference (the error) will be minimized. This means that the contracts will be always on the limit (a well-served client will be overlooked until it becomes bad served).

The *exigency-based scheduling policy* [16] is a newly proposed real-time scheduling algorithm developed in the scope of a research project whose aim is to enable the provision of absolute QoS guarantees in ART-constrained soft-RT systems. EBS assigns the highest priorities to the jobs with the lowest product given by Eq. (1)

$$P = D.C \tag{1}$$

where D is the deadline and C is the expected execution time. Pondering the urgency of a request with the execution time borrows the rationale of the *shortest job first* algorithm. SJF prioritizes the jobs with lower processing times and is known to minimize the average response time.

The EBS rationale is then to prioritize jobs with approaching deadlines but only if they are not too time costly. Recent works have shown the property of the EBS in delivering a fair balance of resource allocation proportional to the demands imposed by each service class [16].

3 Adaptive Resource Scheduling

Figure 2 shows a feedback loop depicting the adaptive resource scheduling architecture developed in this research work. The system input is the vector $[\mu_i]_n^1$ representing the ART upper bounds $\mu_i, i = 1, 2....n$ specified by each of the n clients. The output is the corresponding vector $[m_i]_n^1$ with the effective delivery ARTs.

The workload is represented by a queue of pending requests, where each job is parameterized by both the arrival time T and the cost C (execution time).

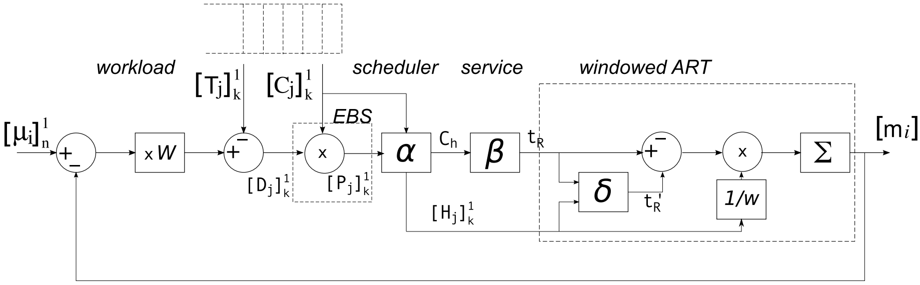


Fig. 2. The feedback loop of the adaptive resource allocation algorithm

Vectors $[T_j]_k^1$ and $[C_j]_k^1$ denote those two parameters, respectively, for all the k jobs in the queue.

If the job j currently in the queue was issued by the client i, then Eq. 2 represents the ART constraint:

$$\frac{m_i.w + Tj + Dj}{w} \leq \mu_i \tag{2}$$

The right side of the expression encompass the total service time client i has experienced so far, plus the time the job has already spent in the queue, plus the time it will still wait if not selected now. The condition says that the average response effectively delivered should not be superior to the contracted value.

The maximum value D_j may assume in this inequation is the deadline for execution of the job before the contract is violated. Eq 3 explicits this value:

$$D_j = (\mu_i - m_i).w - T_j \tag{3}$$

Notice that D_j is a then a time-varying deadline that is valid in every scheduling cycle and must be dynamically recalculated online.

Back to Figure 2, the EBS block receives the jobs' deadline and processing time and produces the priority vector. The block α represents the scheduler. It's function is to get the index h of the larges value in $[P_j]_k^1$ and then pick up the corresponding job in the queue. The other output of this block is a vector $[H_j]_k^1$ filled with zeros in all positions but h, where it holds the number 1.

The cost of the selected job is the input to the block β standing for the service, which is expected to exhibit a response time t_R.

The windowed upper-bounded ART constraint is depicted in the rightmost large block. The new average response time of the served client should be recalculate. An efficient way to perform this step is to calculate the new ART at a given instant from the ART at instant before by adding the value entering the window and subtracting the value leaving it at the other end, w positions back, as in Eq 4 :

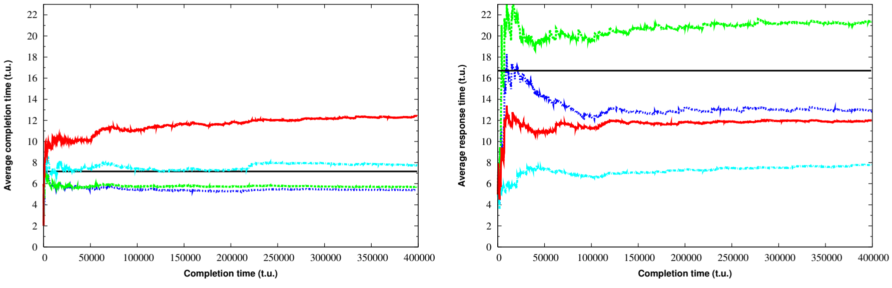
$$m_z = \frac{m_z * w + t_z - t_{z-w}}{m} \tag{4}$$

Block δ is a FIFO of w positions which buffers incoming values of response time and returns the oldest instance in the window. The difference between current and last response time is multiplied by $1/w$ and summed up with the current effective ART, for the served client, to produce the output.

The result of this feedback architecture is an adaptive resource allocation which schedules the access to the service proportionally to the demands (effective service level and workload) of each client at every moment.

4 Development Results

The graphics in Figure 3 show the results of a simulation experiment for a system with two service classes A and B, between which the clients are distributed in



(a) Class A (b) Class B

Fig. 3. Simulation results for the proposed architecture

the proportion of 1 : 3. The system is modeled as standard queue of independent jobs and a single non-preemptive processor.

The experiment was performed by means of a simulation program written in SMPL [17] discrete-event simulation library, which implements several queue scheduling algorithms.

A preliminary essay was run for a FIFO scheduler and the ART delivered to the clients was plotted. The horizontal axis represents the instants at which the service releases each job after having completely processed it, and the vertical axis denotes the ART calculated up to that point within a 200-job window. The graphic of Figure 3(a) refers to one client of class A, while that of Figure 3(b) is for one client of class B[1]. The x-axis represent the time at which a request is released by the service after being completely processed, and the y-axis represents the average response time calculated at that moment.

The QoS contract was then established so that A-class clients are supposed to be assured an ART 40% below that provided by the FIFO scheduler, while B-class clients' ART upper bound is allowed to be 40% higher than that value. The horizontal line marks the contract levels of each service class. As itcan be inferred, EDF heuristic performed disappointingly bad. That is because the scheduler directs all efforts to the clients whose contracts are near their limits, leaving aside the well-served ones until they also eventually approach their ART upper bound. The contracts are always at the limit and it is difficult to handle the stochastic load variations.

In the same graphics it is possible to see the simulation results for a *shortest job first* (SJF) scheduler. SJF prioritizes the jobs with lower processing times and is known to minimize the average response time. It is superior to EDF in this case but, unaware of any QoS contract, it performs equally good for both classes. It assigns more priority than need to fulfill contract B at the expenses of the more exigent contract A.

As it can be seen, out of the four essayed techniques, the proposed method (curve EBS in the graphic) is the only one capable of fulfilling both contracts,

---

[1] Service quality delivering is homogeneous for all the clients in a service class.

and it does so because it is able to loose in performance for class B, in order to decrease the effective ART delivery to class A. The constraint miss ratio calculated for this simulation was 98% for both classes, while the other algorithms achieved less than 40% for either A or B.

5 Conclusions

This paper introduces an adaptive resource control technique for QoS-aware distributed systems with soft real-time requirements.

The proposed architecture is modeled as a feedback loop and its operation rationales are explained. The research is motivated in the context of concurrent SOA implementing real-time business processes and investigates techniques aimed at request scheduling in concurrent services. Results of simulation experiments are used to discuss the properties of the algorithm with respect to the balance of the service scheduling among clients proportionally to the workload demands.

The present work explores the introduced formalization to describe recent results on QoS provision for ART-constrained systems, which are applicable in a wide varied of emerging network applications, specially those operating over communication infrastructures with stochastic performance. The practical use of the QoS model in the definition of novel SLA frameworks for network applications with responsiveness guarantees was addressed. It was also shown how the presented model can be explored in the design of scheduling strategies for resource allocation in computing and communication applications. The introduced adaptive resource scheduling architecture implements an efficient scheduling policy specially designed for ART-constrained real-time systems and represents an effective possibility for novel QoS approaches, and new business models for Internet service provision.

Related and ongoing works at the research group developing this work include and an extension of those results for heterogeneous multiprocessor systems, proposing a novel load balancing algorithm specially tailored for ART-constrained applications [18]. Another work introduces an adaptive admission control algorithm that protects the systems from overload by penalizing the QoS delivered to each client proportionally to the generated workload. A related project is devoted to the implementation of the proposed adaptive architecture as an extension for the mainstream Apache Web server, aimed at enabling service differentiation and performance guarantees in distributed Web architectures and SOA systems. Future works shall address other workload models such as heavy-tailed distributions and real-world.

Acknowledgments

Then authors thanks FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnol gico), and CAPESP (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) for the financial support offered to this research project.

References

1. Sha, L., Abdelzaher, T., Arzen, K.E., Cervin, A., Baker, T., Burns, A., Buttazzo, G., Caccamo, M., Lehoczky, J., Mok, A.K.: Real time scheduling theory: A historical perspective. Real-Time Syst. 28(2-3), 101–155 (2004)
2. Woodside, M., Menasce, D.: Guest editors' introduction: Application-level qos. IEEE Internet Computing 10(10), 13–15 (2006)
3. Wu, X., Li, M., Wu, J.: Enhanced Demand-driven Service Differentiation Algorithm in Web Clusters. In: Proceedings of the IEEE International Conference on e-Business Engineering, pp. 386–391 (2006)
4. Lee, S., Lui, J., Yau, D.: A Proportional-Delay DiffServ-Enabled Web Server: Admission Control and Dynamic Adaptation. IEEE Transactions on Parallel And Distributed Systems, 385–400 (2004)
5. Kang, C., Park, K., Kim, S.: A Differentiated Service Mechanism Considering SLA for Heterogeneous Cluster Web Systems. In: Software Technologies for Future Embedded and Ubiquitous Systems, 2006 and the 2006 Second International Workshop on Collaborative Computing, Integration, and Assurance. SEUS 2006/WCCIA 2006. The Fourth IEEE Workshop on, p. 6 (2006)
6. West, R., Poellabauer, C.: Analysis of a window-constrained scheduler for real-time andbest-effort packet streams. In: Real-Time Systems Symposium, 2000. Proceedings. The 21st IEEE, pp. 239–248 (2000)
7. Bernat, G., Cayssials, R.: Guaranteed on-line weakly-hard real-time systems. In: Proceedings of the 22nd IEEE Real-Time Systems Symposium, pp. 25–35 (2001)
8. Hamdaoui, M., Ramanathan, P.: A dynamic priority assignment technique for streams with (m, k)-firm deadlines. IEEE Transactions on Computers 44(12), 1443–1451 (1995)
9. Koren, G., Shasha, D.: Skip-over: Algorithms and complexity for overloaded systems that allow skips. In: Proceedings of the 16th IEEE Real-Time Systems Symposium, pp. 110–117 (1995)
10. Liu, D., Hu, X., Lemmon, M., Ling, Q.: Firm Real-Time System Scheduling Based on a Novel QoS Constraint. IEEE Transactions on Computers, 320–333 (2006)
11. Monaco, F.J., Mamani, E.L.C., Nery, M., Peixoto, M.L.: A novel qos modeling approach for soft real-time systems with performance guarantees. In: High Performance Computing and Simulation Conference - HPCS, Leipzig, Germany (June 2009) (to appear)
12. Lu, Y., Abdelzaher, T., Lu, C., Sha, L., Liu, X.: Feedback control with queueing-theoretic prediction for relative delay guarantees in web servers. In: Real-Time and Embedded Technology and Applications Symposium, 2003. Proceedings. The 9th IEEE, May 27-30, pp. 208–217 (2003)
13. Sha, L., Abdelzaher, T., Arzén, K.E., Cervin, A., Baker, T., Burns, A., Buttazzo, G., Caccamo, M., Lehoczky, J., Mok, A.K.: Real time scheduling theory: A historical perspective. Real-Time Syst. 28(2-3), 101–155 (2004)
14. Abdelzaher, T., Stankovic, J., Lu, C., Zhang, R., Lu, Y.: Feedback performance control in software services (2003)
15. Lu, C., Abdelzaher, T.F., Stankovic, J.A., Son, S.H.: A feedback control approach for guaranteeing relative delays in web servers. In: RTAS 2001: Proceedings of the Seventh Real-Time Technology and Applications Symposium (RTAS 2001), Washington, DC, USA, p. 51. IEEE Computer Society, Los Alamitos (2001)

16. Casagrande, L.S., Monaco, F.J., de Mello, R.F., Bertagna, R., Filho, J.A.A.: Exigency-based real-time scheduling policy to provide absolute qos for web services. In: SBAC-PAD 2007: Proceedings of the 19th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2006), Gramado, RS, Brazil. IEEE Computer Society, Los Alamitos (2007)
17. MacDougall, M.H.: Simulating computer systems: techniques and tools. MIT Press, Cambridge (1987)
18. Monaco, F.J., Nery, M., Peixoto, M.L.: An orthogonal multi-resource real-time scheduling architecture for responsiveness qos requirements in soa environments. In: 24th Annual ACM Symposium on Applied Computing - ACM SAC, Honolulu, USA, March 8-12, pp. 1990–1995 (2009)