View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Archivio della ricerca della Scuola Superiore Sant'Anna

Advance Reservations for Distributed Real-Time Workflows with Probabilistic Service Guarantees*

Tommaso Cucinotta
Real-Time Systems Laboratory
Scuola Superiore Sant'Anna, Pisa, Italy
Email: cucinotta@sssup.it

Kleopatra Konstanteli and Theodora Varvarigou
Advanced Distributed Computing Laboratory
National Technical University of Athens, Greece
Email: {kkonst,dora}@telecom.ntua.gr

***Abstract*—This paper addresses the problem of optimum allocation of distributed real-time workflows with probabilistic service guarantees over a Grid of physical resources made available by a provider. The discussion focuses on how such a problem may be mathematically formalised, both in terms of constraints and objective function to be optimized, which also accounts for possible business rules for regulating the deployment of the workflows. The presented formal problem constitutes a probabilistic admission control test that may be run by a provider in order to decide whether or not it is worth to admit new workflows into the system, and to decide what the optimum allocation of the workflow to the available resources is. Various options are presented which may be plugged into the formal problem description, depending on the specific needs of individual workflows.**

***Keywords*-advance reservations; real-time interactive workflows; probabilistic service guarantees.**

I. Introduction

Advance reservation mechanisms reserve available resources for a given time span so that the hosted applications may be run with acceptable Quality of Service (QoS) levels. However, the effectiveness of current advance reservation mechanisms is limited when it comes to interactive applications where the users may want to trigger the application at their own convenience and with a guaranteed QoS over a large time span, without having a fixed start time or having to make new reservation arrangements for each execution. Furthermore, dealing with interactive, real-time applications implies that the workflows terminate very shortly after each activation, possibly within a sub-second maximum response-time. This means that the completion time is orders of magnitude smaller than the granularity by which the advance reservation process is done.

In the context of this problem, largely inspired by the IRMOS project[1], a resource provider is a business entity who owns a set of physical resources, and establishes Service-Level Agreements (SLAs) with customers to allow for booking in advance a set of virtualized resources for hosting distributed, soft real-time, interactive applications. Due to the heterogeneous characteristics of the hosts and the interactive nature of the hosted services, a single workflow application rarely manages to saturate a host. Therefore the resource provider has a strong interest in time-sharing multiple concurrently running services on each one of them, up to the saturation. However, due to the tight timing constraints that characterize soft real-time applications, the use of a best-effort scheduling policy, like largely available on a General-Purpose (GP) Operating System (OS), is not adequate for this kind of applications. Appropriate schedulers need to be borrowed from the world of (soft) real-time scheduling. For example, the use of a Xen hypervisor with an S-EDF scheduling policy may fulfill such requirement. Alternatively, it is possible to use the virtualization capabilities of GP OSes, like KVM on Linux, along with an implementation of a soft real-time scheduling policy on the GP OS, like proposed in [6], [7]. More recently, implementations of soft real-time scheduling on Linux can be found in [12], [17] and [5].

The Proportional Share [18] and Pfair [2] techniques approximate the Generalized Processor Sharing theoretical concept of a fluid allocation, in which each application marks a progress proportional to a given weight. Another approach is based on the concept of Resource Reservations [16], in which the resource allocation is specified not only in terms of a share, but also in terms of time granularity. The Constant Bandwidth Server (CBS) [1] is an EDF-based scheduler which provides a strong theoretical foundation that is able to cope with aperiodic arrivals. Two extensions of the CBS were presented in [13][4] for allowing the sharing of resources between real-time tasks in dynamic real-time systems. However, the aforementioned techniques apply only to reservations that are followed immediately by an allocation, and not advance reservations.

More recently, a study on a task-level real-time scheduling algorithm that supports advance reservations was discussed in [15]. Studies on the performance of advance reservation mechanisms with rigid time constraints have shown that they lead in high fragmentation of the scheduling time,

(*) The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7 under grant agreement n.214777 IRMOS – Interactive Realtime Multimedia Applications on Service Oriented Infrastructures.

[1]More information is available at: http://www.irmosproject.eu.

which inevitably results in lower utilization [19]. To this end, techniques such as advance reservations with flexible time constraints [9] are considered as a means to enhance utilization.

The problem of allocation of real-time distributed tasks on a set of heterogeneous hosts has been investigated by Di Natale et al. in [8], in the context of automotive embedded real-time systems. Differently from the traditional real-time literature, this paper considers a probabilistic admission control setting, in which statistical knowledge of actual usage by the users is leveraged, in order to host more real-time tasks over the same physical node than it would be allowed by traditional deterministic real-time admission control tests. Due to the SLAs with the customers, the provider has to trade saturation levels for possible penalties in case of SLA violations. Also, contrary to most of the traditional literature about advance reservations over Grids, this paper considers *interactive* real-time applications with very tight timing-constraints, and time-shared computing units, where soft real-time scheduling policies are in-place to provide strong guarantees on the end-to-end response-times of the workflows.

This paper is organised as follows. In Section II, the definition of the necessary notation is presented, and is followed by the presentation of the SLA model in Section III. The formulation of the problem of allocating distributed applications under probabilistic service guarantees in described in Section IV, whereas Section V provides an illustrative numerical example. Finally, conclusions are drawn in Section VI.

II. Notation

In this section, some basic notation is introduced for referring to resources, applications, services, and advance reservation time intervals, in the subsequent discussion.

A. Resources Topology

The provider's resources may be generally considered as an interconnection of heterogeneous networks that interconnect their own computing nodes. For example, various LANs enclosing multi-processor computing nodes are interconnected by means of one or more WANs. To this direction, the network topology is characterised by the following elements:

- A set of computing nodes, or hosts: $\mathcal{H} = \{1, \ldots, N_H\}$. Each host $h \in \mathcal{H}$ is characterised by a capacity U_h, expressed in terms of availability of processor(s) share.
- A set of available subnets: $\mathcal{S} = \{1, \ldots, S\}$. Each subnet is characterized by a maximum aggregate bandwidth B_s, expressed in terms of bytes/s, and a latency L_s, that depend on the adopted type of medium, scheduling algorithm and protocol for QoS assurance.
- The network topology information, specifying what hosts $\mathcal{H}_s \subset \mathcal{H}$ are connected to each subnet $s \in \mathcal{S}$.

B. Application Workflows

The following notation is used to refer to applications:

- Set of application instances (referred to simply as applications from here on) $\mathcal{A} = \{1, \ldots, N_A\}$, comprising both the set of applications already hosted into the system, denoted by $\mathcal{A}_{old} \subset \mathcal{A}$, and the set of new applications to be admitted, denoted by $\mathcal{A}_{new} \subset \mathcal{A}$.
- Each application $a \in \mathcal{A}$ is a linear workflow of $n^{(a)}$ real-time services $\mathcal{A}^{(a)} \triangleq \left\{1, \ldots, n^{(a)}\right\}$, denoted also as $\left(\tau_1^{(a)}, \ldots, \tau_{n^{(a)}}^{(a)}\right)$. Each service performs some CPU-intensive computation, then transmits some data to the next service in the workflow, which in turn starts its own computations, and so on. Activation requests to the workflow arrive with a minimum $T^{(a)}$ inter-arrival time.

The following elements denote computing and networking requirements and timing constraints of applications:

- Computation time $c_{i,j}^{(a)}$ exhibited by each service $\tau_i^{(a)}$ of application $\mathcal{A}^{(a)}$, if deployed on physical node $j \in \mathcal{H}$.
- Number $m_i^{(a)}$ of bytes to be transmitted by each service $\tau_i^{(a)} \in \tilde{\mathcal{A}}^{(a)}$ of application $\mathcal{A}^{(a)}$, to $\tau_{i+1}^{(a)}$, each time $\tau_i^{(a)}$ completes, where $\tilde{\mathcal{A}}^{(a)} \triangleq \mathcal{A}^{(a)} \setminus \left\{\tau_{n^{(a)}}^{(a)}\right\}$.
- Response-time $\rho_i^{(a)}$ of each service $\tau_i^{(a)}$ of $\mathcal{A}^{(a)}$.
- End-to-end response-time $\rho^{(a)}$ of each application $\mathcal{A}^{(a)}$.

For the sake of simplicity, the issue of how and whether to exploit parallelism on the underlying host is not addressed, therefore each service is deployed within a real-time resource reservation hosting a single execution flow. However, it is straightforward to extend the framework so that applications are characterised as generic direct acyclic graphs, instead of linear workflows, which allows to have parallelism at least at the workflow level. In addition, the $c_{i,j}^{(a)}$ value is the worst-case execution time that would be obtained if the service were deployed alone on one of the CPUs of the host j. Due to the interactive nature of the considered applications, it is reasonable to assume that each workflow service be active only for a short time within the reserved time $I^{(a)}$.

In order to support multi-hop networking across services of an application, the workflow model needs to explicitly posses routing elements. Clearly, in case multi-hop is not required, then the solution to the allocation problem (presented in Section IV) will have the routing elements allocated to the same physical nodes as the attached computing elements.

C. Real-time Scheduling

It is assumed that the real-time scheduling algorithm on each host allows for a simple utilisation-based admission control test. For example, if a Pfair scheduler like the one developed in the $LITMUS^{RT}$ project[2] were available, then the maximum capacity U_h could be an integer denoting the number of processors available on the host. On the other hand, if a partitioned EDF scheduling policy were available, like in [5], then one could model each processor as an individual host, where processors would be interconnected by a virtual high-performance subnet. In both cases, each service is assumed to be deployed within a *resource-reservation [16]* with maximum budget $q_i^{(a)}$ and period $d_i^{(a)}$. This amounts to providing to the service a scheduling guarantee of $q_i^{(a)}$ time units every $d_i^{(a)}$ time units. As a consequence, it can be shown [17] that the time needed for each service to complete is bounded by:

$$\rho_i^{(a)} \leq \left\lceil \frac{q_i^{(a)}}{c_{i,\,j}^{(a)}} \right\rceil d_i^{(a)}. \tag{1}$$

However, if the budget is sufficient to sustain the worst-case execution time, then such value simply reduces to $d_i^{(a)}$.

It is also assumed that subnets exhibit proper packet scheduling capabilities, so that it is possible to assign a precise bandwidth $b_i^{(a)}$ to each data flow needed by $\tau_i^{(a)}$ for transmitting its result ($m_i^{(a)}$ bytes) to $\tau_{i+1}^{(a)}$, ensuring the temporal isolation among multiple data flows within a certain tolerance. For example, the WF^2Q+ [3] scheduling policy would meet such a requirement. This results in an end-to-end response-time that may generically be written as:

$$\rho^{(a)} = \sum_{i \in \mathcal{A}^{(a)}} \left(\left\lceil \frac{q_i^{(a)}}{c_{i,\,j}^{(a)}} \right\rceil d_i^{(a)} + \frac{m_i^{(a)}}{b_i^{(a)}} + \sum_{s \in \mathcal{S}} y_{i,\,s}^{(a)} L_s \right). \tag{2}$$

Note that, as a corner case, a subnet may also represent a point-to-point link, or the local "loopback" connection.

D. Advance Reservations

The following notation defines quantities of interest for the advance reservations framework:

- The available time-slots which may be booked in advance are all of the same duration of Δt time units, and are denoted by $\mathcal{T} \triangleq \{t_k\}_{k \in \mathbb{N}}$, i.e., $t_{k+1} = t_k + \Delta t$.
- Each request of advance reservation for an application $\mathcal{A}^{(a)}$ is associated a time-interval $I^{(a)} \triangleq \left\{t_{s^{(a)}}, \ldots, t_{f^{(a)}-1}\right\} \subset \mathcal{T}$, of duration $\left(f^{(a)} - s^{(a)}\right) \Delta t$, over which it may be activated by users' requests.

The actual time instants at which users will activate the workflow within $I^{(a)}$ is unknown, with the only constraint being that two consecutive activations should be requested at a distance of at least $T^{(a)}$ (or, should they arrive at shorter distances, they might be enqueued). However, as it will be highlighted in Section IV-B, the resource provider is assumed to posses a statistical knowledge on the actual expected activation pattern of the application by the end users.

[2]More information is available at http://www.cs.unc.edu/~anderson/litmus-rt.

In the following, the set of applications whose advance reservation time-intervals $\left\{I^{(a)}\right\}$ include a given time-slot t_k is denoted by $\mathcal{A}(t_k) \triangleq \left\{a \in \mathcal{A} \mid t_k \in I^{(a)}\right\}$, and similarly are defined the symbols $\mathcal{A}_{old}(t_k) \subset \mathcal{A}_{old}$ and $\mathcal{A}_{new}(t_k) \subset \mathcal{A}_{new}$. For the purpose of simplifying notation, in the rest of this paper, whenever the reference time-slot is implicitly identified, the symbols $\mathcal{A}$, $\mathcal{A}_{old}$ and $\mathcal{A}_{new}$ are used in place of the more formally correct ones with the time-slot indication.

III. Service Level Agreement Model

Before introducing the formal problem formulation, it is interesting to overview the possible optimization objectives that may be pursued, considering the resource provider perspective, and the constraints dictated within the customers' SLAs. When dealing with real-time workflows of services that can be distributed across multiple sites and activated in an arbitrary fashion across a long time span, like in our model, acceptance of SLA requests adversely affects the availability of the resources and results in increased rejections, reduced utilisation and consequently reduced revenue. To address this problem we exploit the arbitrary pattern of the workflows' activation and the principle that some clients may accept probabilistic guarantees based on overbooking. Summarising, the SLA for a $a \in \mathcal{A}_{new,}$ in our stochastic model may carry the following parameters:

- The description of the application workflow $\mathcal{A}^{(a)}$, which must be complemented by the computation and network requirements of nodes and links[3]: $c_{i,\,j}^{(a)}$ and $m_i^{(a)}$.
- The time-interval $I^{(a)}$ during which the application may be activated by the user.
- An upper bound $R^{(a)}$ for the end-to-end response-time $\rho^{(a)}$ of the workflow execution.
- A minimum probability $\phi^{(a)}$ that the time constraint $R^{(a)}$ is respected, i.e., the constraint is that the $\phi^{(a)} - th$ quantile of the observed $\rho^{(a)}$ distribution should be $\geq R^{(a)}$.
- A maximum value $\overline{R}^{(a)}$ for the average of $\rho^{(a)}$.
- A minimum probability $\xi^{(a)}$ that the workflow is actually available when the request arrives (within $I^{(a)}$).
- A revenue (gain) $G^{(a)}$ for the provider in case a new advance reservation is accepted.
- A penalty $P^{(a)}$ for the provider if the QoS constraints are violated.

Note that $\phi^{(a)}$ and $\xi^{(a)}$ constitute a formal metric for the "flexibility" of the client under a stochastic SLA. A stochastic SLA is a generalisation of a deterministic one, i.e., setting both $\phi^{(a)}$ and $\xi^{(a)}$ to 1, the client is allowed to require determinism in an SLA. However, it is expected

[3]Such values may not necessarily be specified by the customer, but rather be available to the provider by means of other mechanisms, i.e., by recurring to a proper monitoring/benchmarking.

that the pricing model used by providers will acknowledge more flexible consumers by awarding them with lower prices $G^{(a)}$, whereas consumers with less flexibility are charged at higher prices. The way $G^{(a)}$ and $P^{(a)}$ may be determined or negotiated is out of the scope of the present paper, and will be considered in future research.

IV. Formalization of the problem

Using the definitions in Section II, the problem under study may now be formalized. First of all, let us introduce the variables (unknown) to be computed:

- Set of allocations of services to hosts: $\forall a \in \mathcal{A}_{new}$, $\forall i \in \{1, \dots n^{(a)}\}$, $\forall j \in \mathcal{H}$, $x_{i,j}^{(a)} = 1$ if $\tau_i^{(a)}$ is deployed on host j and 0 otherwise.
- Set of allocations of services to subnets (variables introduced for the purpose of clarity): $\forall a \in \mathcal{A}_{new}$, $\forall s \in \mathcal{S}$, $y_{i,s}^{(a)} = 1$ if $\tau_i^{(a)}$ is deployed on some node $j \in \mathcal{H}_s$.
- Set of allocation periods/deadlines for computation: $\forall a \in \mathcal{A}_{new}$, $\forall i \in \{1, \dots n^{(a)}\}$, $d_i^{(a)}$, with a consequent utilization of $\frac{c_{i,j}^{(a)}}{d_i^{(a)}}$, where $j \in \mathcal{H}$ is the physical node where task $\tau_i^{(a)}$ has been deployed.
- Set of allocation bandwidth for networking: $\forall a \in \mathcal{A}_{new}$, $\forall i \in \{1, \dots n^{(a)}\}$, $b_i^{(a)}$, insisting on the subnet mediating the communications between $\tau_i^{(a)}$ and $\tau_{i+1}^{(a)}$.

A. Deterministic Formulation

Let us now focus on a single advance reservation time-slot t_k, thus in what follows $\mathcal{A}$ is a short-hand for $\mathcal{A}(t_k)$. The set of constraints for the problem is summarised as follows.

- Each service must be allocated to exactly one host:

$$\forall a \in \mathcal{A},\ \forall i \in \mathcal{A}^{(a)},\ \sum_{j \in \mathcal{H}} x_{i,j}^{(a)} = 1. \tag{3}$$

- Each service must be allocated to exactly one subnet:

$$\forall a \in \mathcal{A},\ \forall i \in \mathcal{A}^{(a)},\ \sum_{s \in \mathcal{S}} y_{i,s}^{(a)} = 1. \tag{4}$$

- Coherence between $x_{i,j}^{(a)}$ and $y_{i,s}^{(a)}$ allocations (i.e., the $y_{i,s}^{(a)}$ variables may be derived from the $x_{i,j}^{(a)}$ ones, however they are introduced for clarifying the exposition):

$$\forall a \in \mathcal{A},\ \forall i \in \mathcal{A}^{(a)} \forall s \in \mathcal{S},\ \sum_{j \in \mathcal{H}_s} x_{i,j}^{(a)} = y_{i,s}^{(a)}. \tag{5}$$

- Each pair of consecutive tasks needs to be connected to the same subnet:

$$\forall a \in \mathcal{A},\ \forall i \in \tilde{\mathcal{A}}^{(a)},\ \sum_{s \in \mathcal{S}} y_{i,s}^{(a)} y_{i+1,s}^{(a)} = 1. \tag{6}$$

- Maximum residual subnet capacity:

$$\forall s \in \mathcal{S},\quad \sum_{\substack{a \in \mathcal{A} \\ i \in \tilde{\mathcal{A}}^{(a)}}} y_{i,s}^{(a)} b_i^{(a)} \le B_s \tag{7}$$

- Maximum residual computing capacity for each host j :

$$\forall j \in \mathcal{H},\ \sum_{a \in \mathcal{A}} \frac{\sum_{i \in \mathcal{A}^{(a)}} x_{i,j}^{(a)} c_{i,j}^{(a)}}{d_i^{(a)}} \le U_j. \tag{8}$$

- Maximum value $R^{(a)}$ for $\rho^{(a)}$ (the budget is assumed equal to the WCET in Equation 2):

$$\forall a \in \mathcal{A},\ \sum_{i \in \mathcal{A}^{(a)}} \left(d_i^{(a)} + \frac{m_i^{(a)}}{b_i^{(a)}} + \sum_{s \in \mathcal{S}} y_{i,s}^{(a)} L_s \right) \le R^{(a)}. \tag{9}$$

The constraint in Equation 9 is valid under the assumption that the reservations on underlying physical resources is tuned so as to sustain at most one execution of the entire workflow every minimum workflow activation period $T^{(a)}$. This implies the following further constraints:

- Minimum processor shares ensuring a non-enqueueing semantics of tasks activations:

$$\forall a \in \mathcal{A},\ \forall i \in \mathcal{A}^{(a)},\ d_i^{(a)} \le T^{(a)}. \tag{10}$$

- Minimum network bandwidth ensuring a non-enqueueing semantics for consecutive messages (this allows for considering the traffic generated by $\tau_i^{(a)}$ as having a maximum burstiness of $m_i^{(a)}$ bytes, and a bandwidth requirements of $\frac{m_i^{(a)}}{T^{(a)}}$):

$$\forall a \in \mathcal{A},\ \forall i \in \mathcal{A}^{(a)},\ \frac{m_i^{(a)}}{b_i^{(a)}} + \sum_{s \in \mathcal{S}} y_{i,s}^{(a)} L_s \le T^{(a)}. \tag{11}$$

Equations from 3 to 11 define the set of constraints of the presented allocation problem. Note that, as of now, the new application workflows have been admitted *deterministically*, due to the presence of the saturation constraints 7 and 8, which do not allow for overbooking of the physical hosts. In the following, a possible objective function conforming to the provider's business policy is introduced complementing the deterministic problem formulation, and then a probabilistic rework is presented in Section IV-B.

1) Objective function: When deploying a set of applications within an IRMOS domain, the above formalized problem needs to be solved by optimizing some proper metrics expressing the goodness of the found allocation. Clearly, from a provider perspective, such metrics might be significant of the costs associated with each deployment solution. The problem set-up expressed in equations from 3 to 11 is deterministic, and, as such, a simple metrics to be optimized is due to the costs associated to the use of each host. So, assume each host j is associated a cost of ζ_j, which is incurred for each advance reservation time-slot in which the host is actually used for at least one reservation (i.e., it needs to be turned on, or bought/rented). Therefore, let $\mathcal{H}_{off}(I_h) \subset \mathcal{H}$ denote the set of hosts which have not been booked yet for any time-slot $t_k \in I_h$ (e.g., $\min I_h$), when $\mathcal{A}_{new}$ need to be deployed. The optimization goal becomes:

$$\min_{x_{i,j}^{(a)},\, y_{i,s}^{(a)},\, d_i^{(a)},\, b_i^{(a)}} \sum_{I_h \in \mathcal{G}} \sum_{j \in \mathcal{H}_{off}(I_h)} \zeta_j m_{j,h}, \tag{12}$$

where the $\{m_{j,h}\}$ are Boolean variables with a value of 1 if the j^{th} host is involved in the allocation specified by the $\left\{x_{i,j}^{(a)}\right\}$ at any time within I_h (e.g., $\min I_h$) and 0 otherwise. These variables may actually be computed as a logic combination of the $x_{i,j}^{(a)}$ values, and of the advance reservation time-intervals $\left\{I^{(a)}\right\}$, what may be formalised by adding to the problem the following constraints:

$$\begin{cases} K m_{j,\,h} \geq \sum_{a \in \mathcal{A}(\min I_h),\, i \in \mathcal{A}^{(a)}} x_{i,\,j}^{(a)} \\ m_{j,\,h} \leq \sum_{a \in \mathcal{A}(\min I_h),\, i \in \mathcal{A}^{(a)}} x_{i,\,j}^{(a)} \end{cases} \quad \forall I_h \in \mathcal{G} \tag{13}$$

for a sufficiently high constant $K \geq \sum_{a \in \mathcal{A}} n^{(a)}$. A more interesting perspective is the one in which the possibility to reject one or more applications is added to the problem. In this case, it is useful to introduce the gain $G^{(a)}$ got by the provider in case the application a is accepted. One possible way of dealing with this is by introducing the Boolean variable $x^{(a)}$ with a value of 1 if $\mathcal{A}^{(a)}$ is admitted and 0 otherwise. In such a case, the constraints in Equations 3 and 4 of the problem need to be rewritten as follows:

$$\begin{cases} \sum_{j \in \mathcal{H}} x_{i,\,j}^{(a)} = x^{(a)} & \forall a \in \mathcal{A},\, \forall i \in \mathcal{A}^{(a)} \\ \sum_{s \in \mathcal{S}} y_{i,\,s}^{(a)} = x^{(a)} & \forall a \in \mathcal{A},\, \forall i \in \mathcal{A}^{(a)} \end{cases} \tag{14}$$

The cost function then becomes:

$$\min_{x_{i,\,j}^{(a)},\, y_{i,\,s}^{(a)},\, d_i^{(a)},\, b_i^{(a)}} \sum_{I_h \in \mathcal{G}} \sum_{j \in \mathcal{H}_{off}(I_h)} \zeta_j m_{j,\,h} - \sum_{a \in \mathcal{A}} x^{(a)} G^{(a)}. \tag{15}$$

For example, looking at the problem of admitting a single application, the above shown optimization goal implies that it is accepted into the framework only if the additional costs incurred by the provider for the possibly needed additional hosts are lower than the revenue due to accepting the request.

2) Grouping of Advance Reservation Time-slots: Let $\mathcal{C}(t_k)$ denote the set of constraints from 3 to 11 (or equivalently their probabilistic variant introduced in Section IV-B1), where $\mathcal{A}(t_k)$ is actually considered as the set of applications over which the constraints are posed. In this paper, it is assumed that the allocation (both in terms of deployment decisions, and of scheduling parameters) of each application $\mathcal{A}^{(a)}$, once computed, is kept constant over the entire time-interval $I^{(a)}$. Therefore, the advance reservations already in place over the entire $\{t_k\}$ time-horizon are simply considered by intersecting constraints $\mathcal{C}(t_k)$ for all of the time-slots t_k in the union I of the time-intervals of all the new applications to be admitted: $I \triangleq \bigcup_{a \in \mathcal{A}} I^{(a)}$. So, the set of problem constraints, which correctly account for the advance reservation framework, is in principle:

$$\bigwedge_{t_k \in I} \mathcal{C}(t_k) \tag{16}$$

where the constrained variables are always the same, while the constraints parameters may change over the considered time-slots t_k. However, it is not necessary to actually consider each time-slot individually. In fact, it is possible to exploit a methodology for grouping advance reservations whose principles have already been introduced in [11]. Basically, the entire set of time-slots $t_k \in I$ is partitioned into G disjoint time-slices $\mathcal{G} \triangleq \{I_h\}_{h=1,\ldots,G}$, of non-uniform duration of tn_h time-slots, over which the set of applications does not change, formally: $\forall t_k \in I,\, t_k \in I_h \implies \forall t_j \in I_h,\, \mathcal{A}(t_j) = \mathcal{A}(t_k)$. Considering a reservation group h, it is clear that, in Equation 16, $\mathcal{C}(t_j)$ generate exactly identical constraints $\forall t_j \in I_h$, therefore they are redundant, and only one set of constraints should be considered, for each $t_j \in I_h$, for example the one relative to the earliest time. Equation 16 thus becomes:

$$\bigwedge_{t_k \in \{\min I_h | h=1,\ldots,G\}} \mathcal{C}(t_k). \tag{17}$$

In what follows, $\mathcal{A}(I_h)$ will constitute a short-hand for $\mathcal{A}(\min I_h)$. Finally, it is useful to introduce, for each application $a \in \mathcal{A}$, the set $\mathcal{G}^{(a)}$ of disjoint time-slices concerning a, i.e.: $\mathcal{G}^{(a)} \triangleq \{I_h \in \mathcal{G} \mid a \in \mathcal{A}(I_h)\}$. Note that $\mathcal{G}^{(a)}$ constitutes a partition of $I^{(a)}$, and that the overall number of time instants within is $f^{(a)} - s^{(a)}$.

B. Probabilistic Formalisation

1) Probabilistic response-time guarantees: The response-time constraints may be relaxed in a probabilistic sense, if, instead of relying on worst-case estimates for the computation requirements $\left\{c_{i,\,j}^{(a)}\right\}$, as well as the message sizes $\left\{m_i^{(a)}\right\}$, they are (more effectively, for multimedia) considered as non-completely known values, and modeled as stochastic variables. For the sake of simplicity, it is assumed that they are independent and identically distributed (i.i.d.), and that the provider has an estimate of a certain quantile of their distributions: $\Pr\left[c_{i,\,j}^{(a)} \leq C_{i,\,j}^{(a)}\right] \geq \alpha_i^{(a)}$, and $\Pr\left[m_i^{(a)} \leq M_i^{(a)}\right] \geq \beta_i^{(a)}$, with[4] $\prod_{i \in \mathcal{A}^{(a)}} \alpha_i^{(a)} \beta_i^{(a)} \geq \phi^{(a)}$. Then, in order for an application $\mathcal{A}^{(a)} \in \mathcal{A}_{new}$ to be admitted into the system, instead of guaranteeing that $\rho^{(a)} \leq R^{(a)}$ deterministically, now it may be sufficient to guarantee that $\Pr\left[\rho^{(a)} \leq R^{(a)}\right] \geq \phi^{(a)}$ (this guarantee should hold also for workflows that have already been accepted into the system). This is simply achieved by requiring that[5] the computation requirements $c_{i,\,j}^{(a)}$ be replaced with their quantiles $C_{i,\,j}^{(a)}$ in Equation 8, and the communication requirements $m_i^{(a)}$ with their quantiles $M_i^{(j)}$ in Equation 9. Note that, if $\phi^{(a)} = 1$, then the mentioned quantiles are all forced to become 100% quantiles, i.e., worst-case values (and the deterministic case is obtained as a particular case of the probabilistic one).

2) Probability of conflicting sets of advance-reservations: For any given group $I_h \in \mathcal{G}$, that derives from the grouping process described in [11] and recalled in Section IV-A2, considering the arbitrariety (and independence) in the time instants in which users may request activations of the services, it is possible to compute the probability $P_{j,\,\mathcal{B}}(I_h)$ of overlapping activation of the services in any possible subset $\mathcal{B}$ of $\mathcal{A}(I_h)$ on each given host $j \in \mathcal{H}$, within any interval $I_h \in \mathcal{G}$:

$$P_{j,\,\mathcal{B}}(I_h) = \prod_{a \in \mathcal{B}} \pi_{i,\,j}^{(a)} \prod_{a \in \mathcal{A}(I_h) \setminus \mathcal{B}} \overline{\pi_{i,\,j}^{(a)}} \tag{18}$$

[4]Keeping a sufficient number of quantiles for each service, the provider may find the proper ones that fulfill this condition for a given $\phi^{(a)}$ value from the SLA.

[5]Details are omitted for the sake of brevity, however they can be found in http://feanor.sssup.it/~tommaso/eng/papers-soca09.html.

where $\overline{\pi_{i,j}^{(a)}} \triangleq 1 - \pi_{i,j}^{(a)}$. This expression will constitute a basis for what follows.

3) Probabilistic availability guarantee: Assume the provider has knowledge about the rate $r^{(a)}$ of actual activation of each workflow application a, during the time-period $I^{(a)}$ it has been reserved (this is supposed to be consistently lower than the activation frequency $\frac{1}{T^{(a)}}$ of the application while it is actually being used). These rates may be translated into the probabilities $\pi_i^{(a)}$ as follows: $\pi_i^{(a)} = r^{(a)} \frac{\sum_{j \in \mathcal{H}} x_{i,j}^{(a)} C_{i,j}^{(a)}}{u_i^{(a)}}$, where the probability of finding a service active increases when decreasing the reserved CPU share $u_i^{(a)} = \frac{\sum_{j \in \mathcal{H}} x_{i,j}^{(a)} C_{i,j}^{(a)}}{d_i^{(a)}}$ to that service, because for each activation it will take longer for the service to complete.

Then, under the assumption that $r^{(a)} \ll \frac{1}{T^{(a)}}$, it may be convenient for the provider to overbook the physical resources, and for example host onto a computation node more applications than the ones that might actually fit due to the classical deterministic admission control rule (if the probability of them being all active at the same time is sufficiently low). The problem formulation may thus be enriched by constraining the probability for $\mathcal{A}^{(a)}$ to find enough available resources when actually activated to be higher than $\xi^{(a)}$, with the following formalization[6]:

- adding to the problem further Boolean variables $v_{\mathcal{B}}^j$ and $w_{\mathcal{B}}^s$ for each $\mathcal{B} \subset \mathcal{A}$, $j \in \mathcal{H}$ and $s \in \mathcal{S}$, encoding whether or not the spare computing capacity on j or network capacity on s suffice for hosting the applications in $\mathcal{B}$;
- introducing $u_i^{(a)} \triangleq \frac{1}{d_i^{(a)}} \sum_{j \in \mathcal{H}} x_{i,j}^{(a)} C_{i,j}^{(a)}$ representing the computation bandwidth of $\tau_i^{(a)}$;
- introducing the $\pi_i^{(a)} \triangleq r^{(a)} \frac{\sum_{j \in \mathcal{H}} x_{i,j}^{(a)} C_{i,j}^{(a)}}{u_i^{(a)}}$, $\pi_{i,j}^{(a)} \triangleq 1 - (1 - \pi_i^{(a)}) x_{i,j}^{(a)}$ and $\pi_{i,\{s\}}^{(a)} \triangleq 1 - (1 - \pi_i^{(a)}) \sum_{j \in \mathcal{H}_s} x_{i,j}^{(a)}$;
- adding the following set of constraints, replicated $\forall j \in \mathcal{H}$, $\forall \mathcal{B} \subset \mathcal{A}$, $\forall a \in \mathcal{A}$, $\forall s \in \mathcal{S}$:

$$\begin{aligned}
U_j - \sum_{b \in \mathcal{B}} \sum_{i \in \mathcal{A}^{(b)}} x_{i,j}^{(b)} u_i^{(b)} - \sum_{i \in \mathcal{A}^{(a)}} x_{i,j}^{(a)} u_i^{(a)} &\geq K \left(v_{\mathcal{B}}^j - 1 \right), \ \forall j, \ \mathcal{B} \\
U_j - \sum_{b \in \mathcal{B}} \sum_{i \in \mathcal{A}^{(b)}} x_{i,j}^{(b)} u_i^{(b)} - \sum_{i \in \mathcal{A}^{(a)}} x_{i,j}^{(a)} u_i^{(a)} &\leq K v_{\mathcal{B}}^j - \epsilon, \ \forall j, \ \mathcal{B} \\
B_s - \sum_{b \in \mathcal{B}, \, i \in \tilde{\mathcal{A}}^{(b)}} y_{i,s}^{(b)} b_i^{(b)} &\geq K \left(w_{\mathcal{B}}^s - 1 \right), \ \forall s, \ \mathcal{B} \\
B_s - \sum_{b \in \mathcal{B}, \, i \in \tilde{\mathcal{A}}^{(b)}} y_{i,s}^{(b)} b_i^{(b)} &\leq K w_{\mathcal{B}}^s - \epsilon, \ \forall s, \ \mathcal{B} \\
\sum_{I_h \in \mathcal{G}^{(a)}} \frac{tn_h}{f^{(a)} - s^{(a)}} \prod_{j \in \mathcal{H}} \sum_{\mathcal{B} \subset \mathcal{A}(I_h) \setminus \{a\}} v_{\mathcal{B} \cup \{a\}}^j & \cdot \\
\cdot \prod_{b \in \mathcal{B}} \prod_{i \in \mathcal{A}^{(b)}} \pi_{i,j}^{(b)} \prod_{b \in \mathcal{A}(I_h) \setminus \{a\} \setminus \mathcal{B}} \prod_{i \in \mathcal{A}^{(b)}} \overline{\pi_{i,j}^{(b)}} & \cdot \\
\cdot \prod_{s \in \mathcal{S}} \sum_{\mathcal{B} \subset \mathcal{A}(I_h) \setminus \{a\}} w_{\mathcal{B} \cup \{a\}}^s & \cdot \\
\cdot \prod_{b \in \mathcal{B}} \prod_{i \in \mathcal{A}^{(b)}} \pi_{i,\{s\}}^{(b)} \prod_{b \in \mathcal{A}(I_h) \setminus \{a\} \setminus \mathcal{B}} \prod_{i \in \mathcal{A}^{(b)}} \overline{\pi_{i,\{s\}}^{(b)}} &\geq \xi^{(a)}, \ \forall a
\end{aligned} \tag{19}$$

[6]Details are omitted for the sake of brevity, however they can be found in http://feanor.sssup.it/~tommaso/eng/papers-soca09.html.

where K is a sufficiently large constant, and ϵ is a sufficiently small one. The first two inequalities constrain the $\left\{ v_{\mathcal{B}}^j \right\}$ variables to the services allocation $\left\{ x_{i,j}^{(a)} \right\}$ variables, the following two inequalities constraint the $\{ w_{\mathcal{B}}^s \}$ variables to the subnet allocation $\left\{ y_{i,s}^{(a)} \right\}$ variables, and the last inequality constitutes the actual probabilistic availability constraint, derived from Equation 18, where the availability has been averaged over the grouping intervals in $\mathcal{G}^{(a)}$ ($f^{(a)} - s^{(a)}$ is the number of time-slots in the reserved time-span $I^{(a)}$ and tn_h is the number of time-slots grouped under group h).

However, it is clear that, with a maximum probability of $\xi^{(a)}$, the workflow is expected to not found the needed resources as available, when activated by the user, leading to the necessity to pay the penalty $P^{(a)}$ back to the customer. From the provider's perspective, considering a high number of applications, for each application $a \in \mathcal{A}_{new}$ admitted into the system ($x^{(a)} = 1$), the immediate gain $G^{(a)}$ should be discounted by the expected penalty due t SLA violations $\overline{\xi^{(a)}} P^{(a)}$, where $\overline{\xi^{(a)}} \triangleq 1 - \xi^{(a)}$, obtaining the following overall objective function, to replace the one in Equation 15:

$$\min_{x_{i,j}^{(a)}, y_{i,s}^{(a)}, d_i^{(a)}, b_i^{(a)}} \sum_{I_h \in \mathcal{G}} \sum_{j \in \mathcal{H}_{off}(I_h)} \zeta_j m_{j,h} - \sum_{a \in \mathcal{A}} x^{(a)} \left(G^{(a)} - \overline{\xi^{(a)}} P^{(a)} \right). \tag{20}$$

Note that, in order for the provider to consider acceptance of an application, a necessary condition is that the revenue is greater than the expected penalty $G^{(a)} > \overline{\xi^{(a)}} P^{(a)}$, and that, in case not all applications can be admitted, the ones leading to greater spreads between revenues and associated costs (immediate or expected) will be accepted.

4) Estimation of mean execution response time: Alternatively to the *quantile-based* response-time constraint as discussed in Section IV-B1, it is possible to provide a weaker guarantee relying on the *average* response-time constraint. Services of an application with such a type of guarantee would exploit all of the available bandwidth found on the host they have been allocated to, when the application is actually activated by the user. Also, in such case we rely on a reservation period assignment which is sufficiently small so as to let Equation 1 to be approximated as: $\rho_i^{(a)} = \frac{c_{i,j}^{(a)}}{u_i^{(a)}}$ (see [17] for details). Based on the analysis in [11], the utilization assigned to the service $\tau_i^{(a)}$of application $a \in \mathcal{A}$ can be considered as a discrete random variable $u_i^{(a)}$ with a finite number of possible values, and with a probability distribution that changes for each time-slice $I_h \in \mathcal{G}^{(a)}$.

$$\forall \mathcal{B} \subset \mathcal{A}(I_h) \setminus \{a\}, \ \Pr \left[U_{A,i}^{(a)} = \sum_{j \in \mathcal{H}} x_{i,j}^{(a)} U_j - \sum_{j \in \mathcal{H}} \sum_{b \in \mathcal{B}} x_{i,j}^{(b)} \frac{C_{i,j}^{(b)}}{d_i^{(b)}} \right] = P_{j,\mathcal{B}}(I_h),$$

where $P_{j,\mathcal{B}}(I_h)$ is defined in Equation 18. Thus, it is possible to estimate the service expected response-time $\rho_i^{(a)}$, conditioned to an activation of the workflow in any $t_k \in I_h$:

$$E\left[\rho_i^{(a)} \mid t_k \in I_h\right] = \sum_{\mathcal{B}\subset\mathcal{A}(I_h)\setminus\{a\}} P_{j,\,\mathcal{B}}(I_h) \frac{\sum_{j\in\mathcal{H}} x_{i,\,j}^{(a)} C_{i,\,j}^{(a)}}{\sum_{j\in\mathcal{H}} x_{i,\,j}^{(a)} U_j - \sum_{j\in\mathcal{H}} \sum_{b\in\mathcal{B}} x_{i,\,j}^{(b)} \frac{C_{i,\,j}^{(b)}}{d_i^{(b)}}} \tag{21}$$

It is also possible to estimate the expected value of the overall response-time during the time span $I^{(a)}$ as in:

$$E\left[\rho_i^{(a)}\right] = \sum_{I_h\in\mathcal{G}^{(a)}} \frac{tn_h}{f^{(a)} - s^{(a)}} E\left[\rho_i^{(a)} \mid t_k \in I_h\right] \tag{22}$$

Concluding, it is possible to formalize the constraint to be added to the problem in order to ensure a minimum average end-to-end response-time $\overline{R}^{(a)}$ (as defined in the SLA):

$$\sum_{i\in\mathcal{A}^{(a)}} \sum_{I_h\in\mathcal{G}^{(a)}} \frac{tn_h}{f^{(a)} - s^{(a)}} \sum_{\mathcal{B}\subset\mathcal{A}(I_h)\setminus\{a\}} P_{j,\,\mathcal{B}}(I_h)\cdot$$
$$\cdot \frac{\sum_{j\in\mathcal{H}} x_{i,\,j}^{(a)} C_{i,\,j}^{(a)}}{\sum_{j\in\mathcal{H}} x_{i,\,j}^{(a)} U_j - \sum_{j\in\mathcal{H}} \sum_{b\in\mathcal{B}} x_{i,\,j}^{(b)} \frac{C_{i,\,j}^{(b)}}{d_i^{(b)}}} \le \overline{R}^{(a)} \tag{23}$$

where the networking terms may be handled similarly.

C. Solving the Problem

In the previous subsections, the problem of optimum deployment of distributed, interactive, real-time workflows, providing proper probabilistic availability and service guarantees, and conforming to the resource provider business policy, has been formalised as a set of constraints and objective function, along with a set of interesting variants that may be used depending on the context. The formalised problems, both in the deterministic settings of Section IV-A, and in the probabilistic ones of Section IV-B, fall generally within the class of Mixed-Integer Geometric Programming (MIGP) optimization problems. One possible way of solving these problems is by recurring to the `yalmip` [14] package together with the `gpposy` [10] solver, both available for Matlab. Performance measurements of these solvers over the type of problems that have been introduced have not been gathered yet. However, in the future, the development of a custom solver is foreseen for integrating such algorithm into the software architecture of the IRMOS project.

V. Numerical Example

In this section, a simple example is sketched out in order to highlight the advantages of the proposed SLA model for the provisioning of probabilistic service guarantees. To this purpose, the allocation problem is made trivial, i.e., we consider a workflow application denoted as d ($\mathcal{A}_{new} = \{d\}$) that consists of only one service that has to be deployed necessarily on one of three given hosts . Also, only computing requirements are considered, whereas the networking requirements are neglected. A time horizon $\mathcal{T}$ of 2400 time-slots is considered, in which there are already three applications $\mathcal{A}_{old} = \{a, b, c\}$, allocated on each one of the three hosts j, with the parameters shown in Table I. Finally, we also assume that in the three hosts there is only one potential group of conflict I_h and that the number of

Table I
Execution Parameters

Application	c	d	U	π
a	10	100	0.1	0.050
b	60	120	0.5	0.002
c	35	100	0.35	0.015
d	30	120	0.25	0.030

Table II
Available utilization under different groups of overlapping reservations

$\mathcal{B}$	$P_{j,\,\mathcal{B}}(I_h)$	U_A
{d}	0.027511	1
{a,d}	0.001474545	0.9
{b,d}	5.6145E-5	0.5
{c,d}	4.26645E-4	0.65
{a,b,d}	2.955E-6	0.4
{a,c,d}	2.2455E-5	0.55
{b,c,d}	8.55E-7	0.15
{a,b,c,d}	4.5E-8	0.05

tn_h	$\Pr\left[U^{(d)} \le U_A\right]$	$E\left[\rho^{(d)}\right]$
200	0.92	110.295
300	0.88	105.442
400	0.84	100.589

time-slots under conflict tn_h has a different value depending on the host: {200, 300, 500}. According to the execution requirements of the pre-existing applications as shown in Table I, in order to maintain deterministic guarantees, the maximum utilization offered to the new application would be 0.05, thus d would not be admitted. Given that the client of the new application has defined some flexibility in his SLA, we can examine what the hosts have to offer under probabilistic guarantees.

By applying the grouping methodology in Section IV-A2, and the analysis in Section IV-B, we obtain 7 different subgroups of overlapping reservations $\mathcal{B} \subset \mathcal{A}$: $\{d\}$, $\{a, d\}$, $\{b, d\}$, $\{c, d\}$, $\{a, b, d\}$, $\{a, c, d\}$, $\{b, c, d\}$, $\{a, b, c, d\}$. For each $\mathcal{B}$ we can calculate $P_{j,\mathcal{B}}(I_h)$. For example, $P_{j,\,\{a,\,d\}}(I_h) = \pi^{(a)} \cdot (1 - \pi^{(b)}) \cdot (1 - \pi^{(c)}) \cdot \pi^{(d)} = 0.05 \cdot (1 - 0.002) \cdot (1 - 0.015) \cdot 0.03 \approxeq 0.00147$. Similarly, we obtain the probability of occurrence for all possible subgroups $\mathcal{B}$, which, along with the corresponding available utilization values U_A, are presented in Table II.

As it can be seen, only the last two possible combinations exhibit overallocation if d is accepted, and such combinations occur with a quite low probability, as evident from Table I. By using Equation 23 on the first host we obtain: $\Pr\left[U_{avail} \ge U^{(d)}\right] = \sum_{I_h\in\mathcal{G}^{(d)}} \Pr\left[U_{avail} \ge U^{(d)} \mid \text{d activated at } t_k \in I_h\right] \frac{tn_h}{f^{(d)} - s^{(d)}} = [200 \cdot (0.02751 + 0.00147 + 5.6145E^{-5} + 4.26645E^{-4} + 2.955E^{-6} + 2.2455E^{-5}) + 2200]/2400 \approxeq 0.92$. Also by using Equation 22 we can estimate the expected response time for the new application d on the same host: $E\left[\rho^{(d)}\right] = 200 \cdot [(0.02751 + 0.00147 + 5.6145E^{-5} + 4.26645E^{-4} + 2.955E^{-6} + 2.2455E^{-5}) \cdot {}^{30}/_{0.25} + 8.55E^{-7} \cdot {}^{30}/_{0.15} + 4.5E^{-8} \cdot {}^{30}/_{0.05}] + 2200 \cdot {}^{30}/_{0.25}/2400 \approxeq 110.295$.

So, d would receive the required utilization with a probability of 92%, whereas the average response time would be 110.295. By applying the same rules on each one of the three candidate hosts we obtain the values that are presented in Table II. If these estimated pairs of values match the SLA parameters (i.e., they satisfy Equations 19 and 23 respectively), then the related host may be considered as a candidate for hosting d.

VI. Conclusions

This paper addressed the problem of optimum allocation of distributed real-time workflows under probabilistic service guarantees. It presented a mathematical description of the problem that constitutes a probabilistic admission control test in which statistical knowledge of actual usage by the users is leveraged, in order to host more real-time tasks over the same physical node than it would be allowed by traditional deterministic real-time admission control tests. It also presented an SLA model that allows for proper trade-offs between the saturation levels of the provider's resources and the penalties to be paid to the clients in case of misbehaviour. Future work will focus on providing a strong assessment of the effectiveness of the proposed technique. To this direction a dedicated solver for solving the presented optimization problem will be developed and used in realistic provider's settings and scenarios.

References

[1] L. Abeni and G. Buttazzo. Integrating multimedia applications in hard real-time systems. In *RTSS '98: Proceedings of the IEEE Real-Time Systems Symposium*, page 4, Washington, DC, USA, 1998. IEEE Computer Society.

[2] S. K. Baruah, N. K. Cohen, C. G. Plaxton, and D.A. Varvel. Proportionate progress: A notion of fairness in resource allocation. *Algorithmica*, 15:600–625, 1994.

[3] Jon C. R. Bennett and H. Zhang. Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking*, 5(5):675–689, October 1997.

[4] Marco Caccamo, Giorgio C. Buttazzo, and Deepu C. Thomas. Efficient reclaiming in reservation-based real-time systems with variable execution times. *IEEE Trans. Comput.*, 54(2):198–213, 2005.

[5] Fabio Checconi, Tommaso Cucinotta, Dario Faggioli, and Giuseppe Lipari. Hierarchical multiprocessor CPU reservations for the linux kernel. In *Proceedings of the 5^{th} International Workshop on Operating Systems Platforms for Embedded Real-Time Applications (OSPERT 2009)*, Dublin, Ireland, June 2009.

[6] Tommaso Cucinotta, Gaetano Anastasi, and Luca Abeni. Real-time virtual machines. In *Proceedings of the 29^{th} IEEE Real-Time System Symposium (RTSS 2008) – Work in Progress Session*, Barcelona, December 2008.

[7] Tommaso Cucinotta, Gaetano Anastasi, and Luca Abeni. Respecting temporal constraints in virtualised services. In *Proceedings of the 2^{nd} IEEE International Workshop on Real-Time Service-Oriented Architecture and Applications (RTSOAA 2009)*, Seattle, Washington, July 2009.

[8] Abhijit Davare, Qi Zhu, Marco Di Natale, Claudio Pinello, Sri Kanajan, and Alberto Sangiovanni-Vincentelli. Period optimization for hard real-time diustributed automotive systems. In *Proc. of DAC'07*, San Diego, California, USA, June 2007.

[9] Neena R. Kaushik, Silvia M. Figueira, and Stephen A. Chiappari. Flexible time-windows for advance reservation scheduling. In *MASCOTS '06: Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation*, pages 218–225, Washington, DC, USA, 2006. IEEE Computer Society.

[10] K. Koh, S. Kim, A. Mutapcic, and S. Boyd. gpposy: a matlab solver for geometric programs in posynomial form. technical report. Technical report, Stanford University, May 2006.

[11] Kleopatra Konstanteli, Dimosthenis Kyriazis, Theodora Varvarigou, Tommaso Cucinotta, and Gaetano Anastasi. Real-time guarantees in flexible advance reservations. In *Proceedings of the 2^{nd} IEEE International Workshop on Real-Time Service-Oriented Architecture and Applications (RT-SOAA 2009)*, Seattle, Washington, July 2009.

[12] Karthik Lakshmanan and Raj Rajkumar. Distributed resource kernels: Os support for end-to-end resource isolation. In *RTAS '08: Proceedings of the 2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 195–204, Washington, DC, USA, 2008. IEEE Computer Society.

[13] Giuseppe Lipari, Gerardo Lamastra, and Luca Abeni. Task synchronization in reservation-based real-time systems. *IEEE Trans. Comput.*, 53(12):1591–1601, 2004.

[14] J. Löfberg. Yalmip: a toolbox for modeling and optimization in MATLAB. In *Proc. of the CACSD Conference*, Tapei, Taiwan, 2004.

[15] Anwar Mamat, Ying Lu, Jitender Deogun, and Steve Goddard. Real-time divisible load scheduling with advance reservation. In *ECRTS '08: Proceedings of the 2008 Euromicro Conference on Real-Time Systems*, pages 37–46, Washington, DC, USA, 2008. IEEE Computer Society.

[16] Clifford Mercer, Stefan Savage, and Hideyuki Tokuda. Processor capacity reserves for multimedia operating systems. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, 1994.

[17] Luigi Palopoli, Tommaso Cucinotta, Luca Marzario, and Giuseppe Lipari. AQuoSA — adaptive quality of service architecture. *Software – Practice and Experience*, 39(1):1–31, 2009.

[18] W. Smith, I. Foster, and V. Taylor. Scheduling with advanced reservations. In *Proceedings of the 14th International IEEE/ACM Parallel and Distributed Processing Symposium*, May 2000.

[19] Ion Stoica, Hussein Abdel-wahab, Kevin Jeffay, Sanjoy K. Baruah, Johannes E. Gehrke, and C. Greg Plaxton. A proportional share resource allocation algorithm for real-time, time-shared systems, 1996.