Uniform Multiprocessor Periodic Resource model

Progress Report and Ongoing Work

João Pedro Craveiro and José Rufino
Universidade de Lisboa, Faculdade de Ciências, LaSIGE — Lisbon, Portugal
jcraveiro@lasige.di.fc.ul.pt, ruf@di.fc.ul.pt

I. Introduction

Hierarchical scheduling is a current trend in embedded software, with applications ranging from multimedia applications [3] to safety-critical domains such as the civil aviation, aerospace and automotive industries [4]. The need for independent development and arbitrary number of levels are the main motivation and advantages of *compositional analysis*; compositionality is the property of a complex system that can be analysed by analysing some properties of its components (without knowing their internal structure or hierarchy) and the way they are composed. In this sense, a component comprises a workload, a scheduler, and a resource supply. Compositional analysis comprises three main points [5]. **(i) Local schedulability analysis** Analysing the schedulability of a component's workload upon its scheduler and resource supply. **(ii) Component abstraction** Obtaining the component's interface from its inner characteristics. **(iii) Interface composition** Transforming the set of interfaces abstracting the real-time requirements of individual subcomponents into an interface abstracting the requirement of scheduling them together according to a given *intercomponent scheduling* strategy.

Motivation and problem: Multiprocessors are entering the realm of embedded systems, namely safety-critical and hard real-time systems as those employed in the aerospace and the automotive industries. However, the multiprocessor capabilities are routinely not exploited, because of a lack of support in terms of verification and certification [6]. Due to these industries' prevalent use of and interest in TSP systems, compatibility between TSP and platforms with multiple processors, both identical and non-identical, is highly desired. The ARINC 653 specification, a standard for TSP systems in civil aviation and aerospace, shows limited support thereto [7]. To allow reusing the obtained results in a wider range of systems and applications, we approach this problem through compositional analysis of hierarchical scheduling frameworks (HSF), of which TSP systems are a special case.

Contributions: In [1], we presented the problem of compositional analysis of HSFs on uniform multiprocessors. In this fast abstract, we overview the progress made on the problem since then, and some ongoing work. For a detailed description with proofs, the reader is referred to [2]. This is the first work explicitly dealing with compositional analysis of HSFs on uniform multiprocessor platforms. Our contributions so far [2] are: **(i)** the uniform multiprocessor resource (UMPR) model to serve as a component interface for compositional analysis; **(ii)** a sufficient local schedulability test for sporadic task workloads using global EDF (gEDF) on the UMPR resource model; **(iii)** component abstraction guidelines to select the platform for the UMPR interface; We also briefly describe how we are dealing with the remaining open point — interface composition/intercomponent scheduling.

Related work

Compositional analysis: There is no previous literature explicitly dealing with compositional analysis of HSFs on uniform multiprocessor platforms. Approaches for identical multiprocessors include the multiprocessor periodic resource (MPR) model; an MPR interface (Π, Θ, m) abstracts the provision of Θ processing units over every period with Π time units length over a virtual platform consisting of m identical unit-speed processors [8]. The UMPR extends the MPR for uniform multiprocessors; we chose the MPR because of its simplicity and compositionality potential; other approaches present a less pessimistic approach, at the expense of less simple abstractions specifying the individual contribution of each processor in the virtual platform [9]–[11].

gEDF on dedicated uniform multiprocessors: Schedulability analysis of gEDF on uniform multiprocessors was introduced by Funk et al. [12]. Baruah & Goossens [13] provide a sufficient gEDF-schedulability test for constrained-deadline sporadic task sets of uniform multiprocessors; we extend their approach, since their analysis does not take into account that the platform may be at times partially or totally unavailable.

II. System model

Task model: A component $\mathcal{C}$ comprises a workload (task set) $\mathcal{T}$, with n constrained-deadline sporadic tasks $\tau_i \stackrel{\text{def}}{=} (T_i, C_i, D_i)$. We denote by $\delta_{\max}(\mathcal{T}) \stackrel{\text{def}}{=} \max_{\tau_i \in \mathcal{T}} \frac{C_i}{D_i}$ the maximum density among all tasks in $\mathcal{T}$. The *demand bound function* $\text{DBF}(\tau_i, t)$ gives an upper bound to the maximum cumulative execution requirement by jobs of sporadic task τ_i which have both their arrival and deadline times within any time interval with length t [13].

Platform and scheduling model: We assume a uniform multiprocessor platform with m processors, defined and represented as $\pi \stackrel{\text{def}}{=} \{s_i\}_{i=1}^{m}$, with $1.0 \geq s_i \geq s_{i+1} > 0.0$ for all $i < m$. Each s_i represents a processor's schedulable utilization; this value corresponds to the amount of processor capacity units it provides within one time unit; the total capacity of the platform is expressed as $S_m(\pi) \stackrel{\text{def}}{=} \sum_{i=1}^{m} s_i$. We also make use of the *lambda* parameter, $\lambda(\pi) \stackrel{\text{def}}{=} \max_{\ell=1}^{m-1} \frac{\sum_{j=\ell+1}^{m} s_j}{s_\ell}$, which and abstracts how close is π to an identical multiprocessor platform [12]. We also assume a work-conserving global EDF scheduling strategy with unrestricted migration; without loss of generality, we refer to it simply as gEDF.

This paper partly reports progress on the problem presented at RTSOPS 2012 [1], and summarizes results described in [2]. This work was partially supported by FCT/Égide (PESSOA programme), through the transnational cooperation project SAPIENT; by the EC, through project IST-FP7-STREP-288195 (KARYON); and by FCT, through project PTDC/EEI-SCR/3200/2012 (READAPT), multiannual funding to LaSIGE (UI 408), and Doctoral Grant SFRH/BD/60193/2009.

Resource model — UMPR: To solve the described problem, we propose the uniform multiprocessor periodic resource model, extending and generalizing the MPR model. An interface expressed with the *uniform multiprocessor periodic resource* (UMPR) model $\mathcal{U} \stackrel{\text{def}}{=} (\Pi, \Theta, \pi)$ specifies the provision of Θ units of resource over every period of length Π over a virtual uniform multiprocessor platform π. The supply bound function $\text{SBF}(\mathcal{U}, t)$ expresses the minimum resource supply $\mathcal{U}$ guarantees over *any* time interval of length t (see [2] for its definition).

III. Local schedulability test

We extend Baruah and Goossens's [13] "busy interval" analysis, taking into consideration that the processors available to the component at each instant may not be the fastest processors in the platform. Due to this, we introduce some pessimism regarding the upper bound on the number of jobs that carry in some execution into the "busy interval" (ν).

Theorem 1 (See proof in [2])**.** *Let $\nu = m - 1$. A component $\mathcal{C}$ comprising a constrained-deadline sporadic task set $\mathcal{T}$ is schedulable under gEDF using a UMPR interface $\mathcal{U} = (\Pi, \Theta, \pi)$, if for all tasks $\tau_k \in \mathcal{T}$ and all $A_k \geq 0$,*

$$\sum_{i=1}^{n} \text{DBF}(\tau_i, A_k + D_k) + (\nu + \lambda(\pi)) \cdot (A_k + D_k) \cdot \delta_{\max}(\mathcal{T}) \leq \text{SBF}(\mathcal{U}, A_k + D_k) \ . \tag{1}$$

IV. Component abstraction

For the (identical) multiprocessor resource model, the technique presented in [8] to generate the component interface for $\mathcal{C}$ consists of **(i)** assuming Π is specified by the system designer; **(ii)** computing the values of Θ and m so that $\mathcal{C}$ is schedulable with the least possible resource bandwidth (Θ/Π); for the computation of the schedulability to become tractable, the SBF is replaced by a linear lower bound. For the UMPR, component abstraction is not this simple, because the notions of numbers of processors and total capacity are no longer represented together as only m (and consequently the number of candidate virtual platforms for each component explodes). On the other hand, the available physical platform may impose restrictions on this. As such, we assume both period Π and platform π are specified (for instance, as a requirement presented to the component's developers by their contractor, which will in the end integrate the multiple components), and only Θ needs to be computed to guarantee schedulability. We nevertheless provide important analytically proven guidelines for the selection of platform π.

1) For the same number of processors and total capacity, UMPR interfaces with non-identical uniform multiprocessor platforms are better than those with identical multiprocessor platforms. This is coherent with previous findings in the literature [13].
2) UMPR interfaces with platforms providing the same total capacity with a lower number of faster processors are better than those with a greater number of slower processors.

Our experiments with randomly generated task sets are consistent with these guidelines.

V. Ongoing: Intercomponent scheduling and interface composition

We have proven experimentally [2] and analytically that, in the presence of uniform multiprocessor platforms, classical gEDF scheduling does not guarantee compositionality. We currently have the definition of a scheduler, umprEDF, that guarantees that the effective supply that each component receives is consistent with the interface derived for it. We are working on formally proving its properties, and on interface composition (deriving a global-level resource interface).

Acknowledgments. *The authors would like to thank I. Shin and A. Easwaran for being available to answer our doubts about specific aspects of their work, and the organizers and attendees of RTSOPS 2012 for the fruitful discussions there.*

References

[1] J. P. Craveiro and J. Rufino, "Heterogeneous multiprocessor compositional real-time scheduling," Fast Abstract in RTSOPS 2012, Jul. 2012.
[2] ——, "Compositional hierarchical scheduling frameworks on uniform multiprocessors," University of Lisbon, DI–FCUL, Tech. Rep. TR-2012-07, 2012, revised January 2013). [Online]. Available: http://hdl.handle.net/10455/6891
[3] L. Abeni and G. Buttazzo, "Integrating multimedia applications in hard real-time systems," in *RTSS '98*, Dec. 1998.
[4] J. Rufino, J. Craveiro, and P. Verissimo, "Architecting robustness and timeliness in a new generation of aerospace systems," in *Architecting Dependable Systems VII, LNCS 6420*, A. Casimiro, R. de Lemos, and C. Gacek, Eds. Springer, 2010, pp. 146–170.
[5] I. Shin and I. Lee, "Compositional real-time schedulability analysis," in *Handbook of Real-Time and Embedded Systems*, I. Lee, J. Y.-T. Leung, and S. H. Son, Eds. Chapman & Hall / CRC, 2007.
[6] J. Anderson, S. Baruah, and B. Brandenburg, "Multicore operating-system support for mixed criticality," in *CPS Week 2009 Workshop on Mixed Criticality*, San Francisco, CA, USA, Apr. 2009.
[7] J. Craveiro, J. Rufino, and F. Singhoff, "Architecture, mechanisms and scheduling analysis tool for multicore time- and space-partitioned systems," *ACM SIGBED Rev.*, vol. 8, no. 3, Sep. 2011, ECRTS '11 WiP.
[8] I. Shin, A. Easwaran, and I. Lee, "Hierarchical scheduling framework for virtual clustering of multiprocessors," in *ECRTS '08*, Jul. 2008.
[9] E. Bini, G. Buttazzo, and M. Bertogna, "The multi supply function abstraction for multiprocessors," in *RTCSA '09*, Aug. 2009.
[10] E. Bini, M. Bertogna, and S. Baruah, "Virtual multiprocessor platforms: Specification and use," in *RTSS '09*, Dec. 2009.
[11] A. Burmyakov, E. Bini, and E. Tovar, "The Generalized Multiprocessor Periodic Resource interface model for hierarchical multiprocessor scheduling," in *RTNS '12*, Nov. 2012.
[12] S. Funk, J. Goossens, and S. Baruah, "On-line scheduling on uniform multiprocessors," in *RTSS '01*, Dec. 2001.
[13] S. Baruah and J. Goossens, "The EDF scheduling of sporadic task systems on uniform multiprocessors," in *RTSS '08*, Nov./Dec. 2008.