Improving Resource Utilization for Compositional Scheduling using DPRM Interfaces

Jaewoo Lee   Linh T.X. Phan   Sanjian Chen   Oleg Sokolsky   Insup Lee

Department of Computer and Information Science, University of Pennsylvania

Email: {jaewoo, linhphan, sanjian, sokolsky, lee}@cis.upenn.edu

Abstract—The paper revisits the generation of interfaces for compositional real-time scheduling. Following an established line of research, we use periodic resource models in component interfaces to describe resource demand of the component. We identify a deficiency of existing interface generation algorithms that may require parameters of the resource model to be infeasibly small. We propose a new algorithm for interface generation that avoids this deficiency. We further demonstrate that resource utilization can be improved by using *dual-periodic resource model* (DPRM) interfaces that employ two periodic resource models to characterize the resource demand more precisely.

Index Terms—hierarchical real-time scheduling; periodic resource model; interface generation

I. Introduction

Component-based design has become the widely used technology for the construction of complex computer-based systems. Component technologies allow us to apply the divide-and-conquer approach to reduce design complexity. Components provide well-defined interfaces that abstract away implementation details and enable reuse of a component in different applications. Furthermore, many modern systems are developed through collaboration of many independent providers; in this case, components allow us to encapsulate intellectual properties.

Increasingly, real-time systems are also built using independently developed components. However, unlike conventional systems, real-time components need to satisfy timing and resource constraints and thus have to be allocated sufficient computational resources for this purpose. Schedulability analysis is employed to check that all timing constraints of an application containing multiple real-time tasks will be satisfied in the implementation. However, classical schedulability analysis algorithms are global; that is, they need to know all the tasks that comprise the system. This global nature of schedulability analysis greatly reduces the benefits of component-based development.

Compositional schedulability analysis techniques have been developed to allow component-based development to be used for systems where multiple independently developed components share a computational resource [1]. Interfaces of real-time components contain information about the resource needs of a component, and the system scheduler uses this information to allocate resources to components. Within a component, a separate component-level scheduler further allocates the resource to the component workload, which can contain real-time tasks or other subcomponents.

A common way to represent resource requirements in a component interface is to use a *resource model* [2]. Several resource models have been proposed in the literature, with the periodic resource model [1] being one of the most commonly used. A periodic resource model $\Gamma = (\Pi, \Theta)$ used as the interface of a component specifies that the component needs to be allocated at least Θ units of resource access in every Π time units. A necessary part of a resource model-based compositional schedulability analysis framework is an algorithm to calculate parameters of the resource model sufficient to make the component schedulable. It is also desirable to make such an algorithm optimal so that the component is not allocated unnecessary resources.

An optimal algorithm for the calculation of resource interfaces has been introduced in [3]. The algorithm computes periodic resource model (Π, Θ) that minimizes the *resource bandwidth* Θ/Π. While theoretically optimal, the algorithm cannot always be used in practice, because it calculates Θ as a rational number. Practically, Θ should be an integer multiple of the time slice used by the operating system, which may not be under the control of the application developer. We thus restrict the set of acceptable periodic resource models to have integer values of both Π and Θ. While scaling both Π and Θ by the same factor may yield an acceptable resource model with the same bandwidth, we remind the reader that Π cannot be made arbitrarily large, otherwise the component will become unschedulable due to the *blocking interval* of the resource model [1]. It is clear that an approximation of the optimal resource model with integer values introduces additional overhead into the scheduling framework. One of the goals of this paper is to quantify this overhead.

Furthermore, it is not sufficient to round up the value calculated by the existing algorithm. Consider the following example. Let the optimal resource model for a component be (1,0.54). Rounding the result up, we obtain the resource model (1,1). However, this may not be the minimum bandwidth that can be obtained with integer values, as a periodic resource model (4,3) may be able to schedule the component. We thus set out to develop a new algorithm to calculate an acceptable periodic resource model with the minimum bandwidth.

We then show that it is possible to characterize resource demand of a component even more precisely. We introduce a *dual-periodic resource model* (DPRM) interface, which contains two periodic resource models instead of one. It can be shown that if rational numbers are used in periodic models,

DPRM interfaces do not improve the total resource bandwidth [3]. However, when restricted to integer parameter values, we show that it is possible to reduce the overhead of the interface bandwidth by using DPRM. An extensive simulation study allows us to demonstrate the scale of the improvement.

Contributions. This paper makes three distinct contributions related to the use of periodic resource models in the interfaces of real-time components.

- We propose an efficient algorithm to calculate the minimum-bandwidth periodic resource model with integer parameter values.
- All algorithms for resource model calculation, including the one proposed here, rely on an upper bound on the value of the resource model period Π. In the literature, the upper bound is a parameter of the algorithm specified by the designer. In this paper, we derive a the theoretical upper bound for the period of the minimum-bandwidth resource model.
- Finally, we propose a new resource-demand interface, DPRM, and show that it allows us to reduce resource utilization compared to the minimum-bandwidth periodic resource model with integer parameters.

Related work. Since the first two-level hierarchical real-time scheduling framework introduced by Deng et al. [4] and its extension to multi-level hierarchical systems [2], several compositional analysis techniques have been proposed for such systems (see e.g., [1], [3], [5]). The majority of these techniques assume independent periodic task models – or their variations – for the components. However, these techniques have also recently been extended to analyze hierarchical systems with dependency, such as systems containing interacting tasks [6] and resource sharing [7]. Compositional analysis methods have also been investigated in the context of virtual machine (VM) environment [8], [9].

Most of the existing compositional analysis frameworks represent component interfaces using one of the two resource models: periodic [1] and explicit deadline periodic [5]. The advantages of these two resource models are that they can be directly transformed into real-time tasks, which are required by the upper-level scheduler, and their supply bound functions have regular structures that allow for optimal interface generation. All the existing algorithms, however, assume that the resource model take rational parameter values, which cannot always be used in practice. Further, these algorithms rely on a pre-specified bound on the resource period that is manually chosen by the designer, which cannot guarantee the optimality of the output interfaces.

Organization. The next section revisits the hierarchical scheduling framework. Section III-A presents a bound on the resource period and a revised interface generation algorithm using this bound, followed by a more efficient algorithm in Section IV. Section V proposes the DPRM interface that is able to reduce this overhead suffered by the periodic resource interface. Finally, we present our evaluation of our proposed techniques in Section VI before concluding the paper.

II. HIERARCHICAL SCHEDULING BACKGROUND

In a hierarchical scheduling framework, a system is composed of a set of real-time *components* that are scheduled in a tree-like manner as shown in Figure 1. Each component C in the system is defined by a tuple (W, Γ, A), where W is the component's workload, Γ is the resource interface of the component, and A is the scheduling policy that is used to schedule W. The workload W consists of either (i) a finite set of real-time tasks $\{T_1, T_2, \ldots, T_n\}$, if C is a leaf-component; or (ii) a finite set of subcomponents $\{C_1, C_2, \ldots, C_n\}$, otherwise. The resource interface Γ captures the minimum amount of resource that must be given to C to feasibly schedule the tasks/components in W. The compositional analysis of the system involves (1) computing the resource interface for each leaf-component from the resource demands of its tasks, and (2) subsequently, computing the resource interface for each non-leaf component from the interfaces of its subcomponents. We will focus on the former; the latter can be done using similar techniques as in [1].

In this paper, we assume that all tasks are periodic tasks with relative deadlines equal to periods. Each task T is defined by a period (deadline) p, a worst-case execution time e, with $p \geq e > 0$. The scheduling policy A is assumed to be Earliest Deadline First (EDF) and all our discussions pertain to EDF (without mentioning it explicitly). Note, however, the methods developed here can easily be extended to the RM (Rate Monotonic) by substituting the schedulability condition of EDF with that of RM.
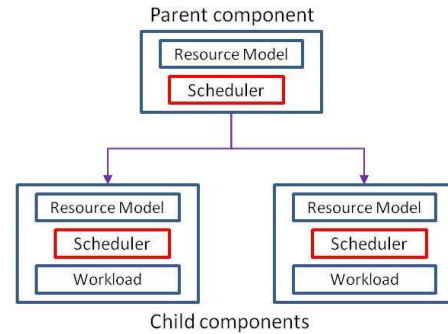

Fig. 1. A hierarchical scheduling system.

Schedulability condition. Given a workload W, the real-time resource requirement of W is characterized by a *demand bound function* (DBF) [10], denoted by $\mathsf{dbf}_W(t)$, which gives the maximum number of execution (resource) units required by the tasks/components of W in any time interval of length t for all $t \geq 0$. The DBF of a workload $W = \{T_1, T_2, \cdots, T_n\}$, with $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$, under EDF is [10]:

$$\forall\, t \geq 0, \quad \mathsf{dbf}_W(t) = \sum_{i=1}^{n} \left(\lfloor \frac{t}{p_i} \rfloor e_i \right)$$

Similarly, the minimum resource guaranteed by a resource model Γ is captured by a *supply bound function* (SBF) [1], written as $\mathsf{sbf}_\Gamma(t)$, which gives the minimum number of execution units provided by Γ in any time interval of length t for all $t \geq 0$. Lemma 1 states the schedulability condition

based on DBF and SBF [11]. In this lemma and the rest of the paper, LCM_W denotes the least common multiple (LCM) of all p_i where $1 \leq i \leq n$.

Lemma 1: Given a component $C = \big(W, \Gamma, EDF\big)$ with $W = \{T_1, T_2, \cdots, T_n\}$ and $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$. Then, C is schedulable (Γ can feasibly schedule W) iff

$$\forall t \ s.t. \ \ 0 \leq t \leq \mathsf{LCM}_W, \ \ \mathsf{sbf}_\Gamma(t) \geq \mathsf{dbf}_W(t). \quad (1)$$

Periodic Resource Model. A periodic resource model is defined by $\Gamma = (\Pi, \Theta)$ where Π is the resource period and Θ is the execution time guaranteed by Γ within every Π time units. The SBF of Γ is thus given by [1]:

$$\mathsf{sbf}_\Gamma(t) = \begin{cases} y\Theta + \max\big(0, \ t - x - y\Pi\big), & \text{if } t \geq \Pi - \Theta \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where $x = 2(\Pi - \Theta)$ and $y = \lfloor \frac{t - (\Pi - \Theta)}{\Pi} \rfloor$.

An important concept associated with the periodic resource model is *bandwidth*. Specifically, the bandwidth of $\Gamma = (\Pi, \Theta)$ is given by $\mathsf{bw}_\Gamma = \frac{\Pi}{\Theta}$. A resource model is *bandwidth-optimal* for a workload W iff its bandwidth is the smallest among that of any resource model that can feasibly schedule W. In this paper, our notion of optimality refers to bandwidth-optimality

Definition 1 (Bandwidth-Optimal): A periodic resource model $\Gamma = (\Pi, \Theta)$ is bandwidth-optimal for a given workload W iff $\mathsf{bw}_\Gamma \leq \mathsf{bw}_{\Gamma'}$ for all Γ' that can feasibly schedule W.

Computation of the optimal periodic resource model. Algorithm 1 outlines the conventional procedure for computing the optimal resource model of a given workload (see e.g., [1], [12]). In this algorithm, $\Pi_{\max}$ is a predefined upper bound on the resource period. The function $\mathsf{MinExec}(\Pi, \mathsf{dbf}_W, \mathsf{LCM}_W)$ (Line 3) computes the minimum Θ for a given Π such that $\Gamma = (\Pi, \Theta)$ can feasibly schedule W (c.f. Lemma 1).

Algorithm 1 the optimal periodic resource model computation
Input: $\Pi_{\max}$, and dbf_W and LCM_W of a workload W
Output: The minimum bandwidth periodic resource model Γ
1: $minBW = 1$
2: **for** $\Pi = 1$ to $\Pi_{\max}$ **do**
3: $\quad \Theta = \mathsf{MinExec}(\Pi, \mathsf{dbf}_W, \mathsf{LCM}_W)$
4: $\quad$ **if** $\frac{\Theta}{\Pi} < minBW$ **then**
5: $\quad\quad minBW = \frac{\Theta}{\Pi}$
6: $\quad\quad \Gamma = (\Pi, \Theta)$
7: $\quad$ **end if**
8: **end for**

In existing work, the maximum bound $\Pi_{\max}$ of the resource period used in Algorithm 1 is either not discussed (and thus, ultimately infinite) or manually chosen by the designer. While the former approach is infeasible, the latter does not guarantee optimality, as illustrated in the example below.

Example 1: Consider a workload $W = \{T_1, T_2\}$ with $T_1 = (51, 23)$ and $T_2 = (130, 70)$. Suppose $\Pi_{\max}$ is chosen to be 80 in Algorithm 1. Then, the output given by Algorithm 1 is $\Gamma = (1, 1)$. However, this resource model is *not* optimal because there exists a periodic resource model $\Gamma' = (97, 96)$, which can feasibly schedule W (c.f. Lemma 1 and Equation 2) and has a lower bandwidth than that of Γ (because $\frac{96}{97} < \frac{1}{1}$).

Since the optimality of Algorithm 1 depends on how large $\Pi_{\max}$ is, the value chosen for $\Pi_{\max}$ must guarantee that the algorithm always outputs a minimum bandwidth model. Simultaneously, $\Pi_{\max}$ should be as small as possible to limit the computational complexity. In the next section, we present our method for computing the bound $\Pi_{\max}$ theoretically and a revised version of Algorithm 1 that uses this bound.

III. BOUND ON OPTIMAL RESOURCE PERIOD AND A REVISED ALGORITHM

A. An upper bound on the resource period

We first define the preliminary results that serve as foundation for our computation. Observe that any SBF of a periodic resource model can be upper bounded by a linear function. We define the *upper supply bound function* (USBF) [5] of a resource model Γ to be the linear function with the smallest slope among all linear functions that upper bound sbf_Γ. The USBF of a periodic resource model $\Gamma = (\Pi, \Theta)$ is [5]:

$$\forall t \geq 0 : \ \mathsf{usbf}_\Gamma(t) = \max\Big(\frac{\Theta}{\Pi}\big(t - (\Pi - \Theta)\big), 0 \Big). \quad (3)$$

Lemma 2: Given a component $C = (W, \Gamma, EDF)$ where $W = \{T_1, T_2, \cdots, T_n\}$, $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$, and $\Gamma = (\Pi, \Theta)$. Then, C is schedulable only if

$$\forall t \ s.t. \ \ 0 \leq t \leq \mathsf{LCM}_W, \ \ \mathsf{usbf}_\Gamma(t) \geq \mathsf{dbf}_W(t). \quad (4)$$

The proof of Lemma 2 is available in our technical report [13]. Note that Lemma 2 is necessary and not sufficent condition. By abuse of notation, we refer to Equation 4 as the *USBF-schedulability condition* for W and we say that a model Γ can *potentially* schedule W iff it satisfies Equation 4.

Basic ideas. The upper bound on the resource period of the optimal periodic resource model for a given workload W can be derived based on dbf_W and its relationship with the USBFs of the resource models that can potentially schedule W. Intuitively, let M be the set of resource models that can potentially schedule W. Suppose $\Gamma_{\mathsf{opt}} = (\Pi_{\mathsf{opt}}, \Theta_{\mathsf{opt}})$ is the optimal resource model for W. Then,

$$\mathsf{bw}_{\Gamma_{\mathsf{opt}}} \geq B_{\min} \overset{\text{def}}{=} \min\big\{ \mathsf{bw}_\Gamma \mid \Gamma \in M \big\}.$$

To derive the bound on Π_{opt}, we will find all the possible resource models in M that have the minimum bandwidth equal to $B_{\min}$. Towards this, we vary the resource period Π and compute for each Π a (unique) resource model $\Gamma_\Pi = (\Pi, \Theta_\Pi)$ belonging to M that has the minimum bandwidth among all the resource models in M with the same resource period Π. Then, $B_{\min}$ can be computed by $B_{\min} = \min\big\{ \mathsf{bw}_{\Gamma_\Pi} \mid \Pi \in \mathbb{N} \big\}$. We will show that for all Π, the USBF of Γ_Π intersects dbf_W at exactly one special point – a *critical time point*. At the same time, Γ_Π has the largest bandwidth among all the resource models $\Gamma_{\Pi,t}$ with period Π that have their USBF intersecting dbf_W at critical time points t. In other words,

$$\mathsf{bw}_{\Gamma_\Pi} = \max_{\substack{t \in \mathsf{CrT}_W \\ \Gamma_{\Pi,t} = (\Pi, \Theta_{\Pi,t})}} \{\mathsf{bw}_{\Gamma_{\Pi,t}} \mid \mathsf{usbf}_{\Gamma_{\Pi,t}}(t) = \mathsf{dbf}_W(t)\},$$

where CrT_W is the set of all critical time points of W, which is determined based solely on the structure of dbf_W. Further, for given any Π and any $t \in \mathsf{CrT}_W$, we compute

the bandwidth of $\Gamma_{\Pi,t}$ directly from $\mathsf{dbf}_W(t)$, Π and t. From these values, we derive $B_{\min}$, which allows us to bound Π_{opt}.

Computation details. First, we define the set of critical time points of a workload W.

Definition 2: Given a workload $W = \{T_1, T_2, \cdots, T_n\}$ where $n \in \mathbb{N}$ and $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$. The set of critical time points of W is defined by

$$\mathsf{CrT}_W = \left\{ \operatorname*{argmax}_{0 < t < \mathsf{LCM}_W} \frac{\mathsf{dbf}_W(t)}{t - s} \mid s \in \{1, 2, \cdots, l\} \right\} \quad (5)$$

where $y = \min_{1 \leq i \leq n} p_i$ and $l = y - \mathsf{dbf}_W(y)$.

Example 2: Consider $W = \{(5, 1), (12, 2), (17, 1)\}$. The set of critical time points of W is $\mathsf{CrT}_W = \{5, 36\}$, which is shown in Figure 2.
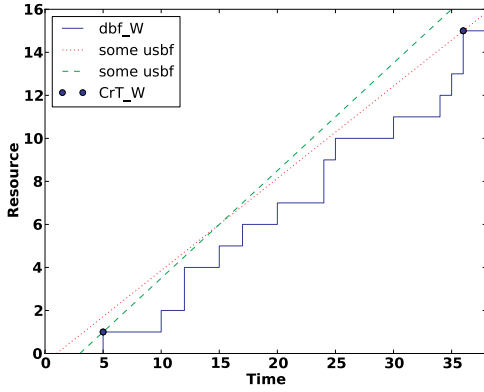

Fig. 2. Relationship between CrT_W and dbf_W.

Lemma 3: Given a component $C = (W, \Gamma, EDF)$ where $W = \{T_1, T_2, \cdots, T_n\}$, $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$, and $\Gamma = (\Pi, \Theta)$. Suppose Γ satisfies the USBF-schedulability condition for W. Then, for all $t \geq 0$, if $\mathsf{usbf}_\Gamma(t) = \mathsf{dbf}_W(t)$ then $t \in \mathsf{CrT}_W$.

Proof: We will prove the lemma by contradiction. Suppose there exists $t_0 \notin \mathsf{CrT}_W$ such that $\mathsf{usbf}_\Gamma(t_0) = \mathsf{dbf}_W(t_0)$. Let $s = \Pi - \Theta$. Then, by Definition 2,

$$\exists\, t' \in \mathsf{CrT}_W : \frac{\mathsf{dbf}_W(t')}{t' - s} > \frac{\mathsf{dbf}_W(t_0)}{t_0 - s} \quad (6)$$

On the other hand, $\mathsf{usbf}_\Gamma(t') \geq \mathsf{dbf}_W(t')$ implies

$$\frac{\Theta}{\Pi}(t' - (\Pi - \Theta)) \geq \mathsf{dbf}_W(t')$$
$$\Rightarrow \quad \frac{\mathsf{dbf}_W(t_0)}{t_0 - s}(t' - s) \geq \mathsf{dbf}_W(t')$$
$$\Rightarrow \quad \frac{\mathsf{dbf}_W(t_0)}{t_0 - s} \geq \frac{\mathsf{dbf}_W(t')}{t' - s}. \quad (7)$$

Since Equation 7 contradicts Equation 6, the lemma holds. ∎

For any given Π and any given $t \in \mathsf{CrT}_W$. Let $\Gamma_{\Pi,t}$ be the resource model with period Π such that its USBF intersects dbf_W at time point t. Then, $\Gamma_{\Pi,t}$ is unique and its bandwidth can be determined using Lemma 4. Due to space constraint, we refer the reader to [13] for the proof of the lemma.

Lemma 4: Given any $\Pi \in \mathbb{N}$ and any $t \in \mathsf{CrT}_W$. Let $\Gamma_{\Pi,t} = (\Pi, \Theta)$ be the periodic resource model such that

$\mathsf{usbf}_{\Gamma_{\Pi,t}}(t) = \mathsf{dbf}_W(t)$. Then, $\Theta = \mathsf{Exec}(\Pi, t, d_t)$ and $\mathsf{bw}_{\Gamma_{\Pi,t}} = \mathsf{BW}(\Pi, t, d_t)$, where $d_t = \mathsf{dbf}_W(t)$ and

$$\mathsf{Exec}(\Pi, t, d_t) \stackrel{\mathsf{def}}{=} \frac{\Pi - t + \sqrt{(\Pi - t)^2 + 4\Pi d_t}}{2},$$

$$\mathsf{BW}(\Pi, t, d_t) \stackrel{\mathsf{def}}{=} \frac{\Pi - t + \sqrt{(\Pi - t)^2 + 4\Pi d_t}}{2\Pi}.$$

The bound $\Pi_{\max}$ on the optimal period Θ_{opt} can now be computed based on CrT_W and a known resource model $\Gamma_c = (\Pi_c, \Theta_c)$ with $\Pi_c \leq \Pi$ that can feasibly schedule W.

Theorem 1: Given a workload $W = \{T_1, T_2, \cdots, T_n\}$ with $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$. Suppose $\Gamma_c = (\Pi_c, \Theta_c)$ is the current periodic resource model obtained at some intermediate execution step of Algorithm 1. Then, the optimal periodic resource model $\Gamma_{\mathsf{opt}} = (\Pi_{\mathsf{opt}}, \Theta_{\mathsf{opt}})$ for W satisfies

$$\Pi_c \leq \Pi_{\mathsf{opt}} \leq \mathsf{MaxResPeriod}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$$

where $\kappa = \frac{\Theta_c}{\Pi_c}$ and

$$\mathsf{MaxResPeriod}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W) \stackrel{\mathsf{def}}{=} \min_{t \in \mathsf{CrT}_W} \frac{\kappa t - \mathsf{dbf}_W(t)}{\kappa(1 - \kappa)}.$$

Proof: Since Algorithm 1 finds the optimal resource period in an increasing manner, $\Pi_{\mathsf{opt}} \geq \Pi_c$. Further,

$$\mathsf{bw}_{\Gamma_{\mathsf{opt}}} \leq \mathsf{bw}_{\Gamma_c} = \kappa. \quad (8)$$

Next, for any given $t \in \mathsf{CrT}_W$, let $\Gamma_{\Pi_{\mathsf{opt}},t} = (\Pi_{\mathsf{opt}}, \Theta_t)$ where $\Theta_t = \mathsf{Exec}(\Pi_{\mathsf{opt}}, t, d_t)$ and $d_t = \mathsf{dbf}_W(t)$. That is, the USBF of $\Gamma_{\Pi_{\mathsf{opt}},t}(t)$ intersects dbf_W at time point t. Let $\Gamma^*_{\mathsf{opt}} = (\Pi_{\mathsf{opt}}, \Theta^*_{\mathsf{opt}})$ be the resource model with the minimum bandwidth among all resource models with period Π_{opt} that can potentially schedule W. Then, its bandwidth must be at least equal to that of $\Gamma_{\Pi_{\mathsf{opt}},t}$ for all $t \in \mathsf{CrT}_W$ (otherwise, Γ^*_{opt} does not satisfy the USBF-schedulability condition). Thus,

$$\forall\, t \in \mathsf{CrT}_W : \mathsf{bw}_{\Gamma^*_{\mathsf{opt}}} \geq \mathsf{BW}(\Pi_{\mathsf{opt}}, t, d_t) \quad (9)$$

On the other hand, since Γ_{opt} can feasibly schedule W, its bandwidth must be at least equal to that of Γ^*_{opt}. That is,

$$\mathsf{bw}_{\Gamma_{\mathsf{opt}}} \geq \mathsf{bw}_{\Gamma^*_{\mathsf{opt}}} \quad (10)$$

Combine Equations 8, 9 and 10, we obtain: For all $t \in \mathsf{CrT}_W$, $\mathsf{BW}(\Pi_{\mathsf{opt}}, t, d_t) \leq \kappa$ is equivalent to

$$\Leftrightarrow \sqrt{(\Pi_{\mathsf{opt}} - t)^2 + 4\Pi_{\mathsf{opt}} d_t} \leq 2\Pi_{\mathsf{opt}}\kappa + t - \Pi_{\mathsf{opt}}$$
$$\Leftrightarrow (\Pi_{\mathsf{opt}} - t)^2 + 4\Pi_{\mathsf{opt}} d_t \leq \big((2\kappa - 1)\Pi_{\mathsf{opt}} + t\big)^2$$
$$\Leftrightarrow \Pi_{\mathsf{opt}} \leq \frac{\kappa t - d_t}{\kappa(1 - \kappa)} = \frac{\kappa t - \mathsf{dbf}_W(t)}{\kappa(1 - \kappa)}$$

The above can be rewritten as $\Pi_{\mathsf{opt}} \leq \min_{t \in \mathsf{CrT}_W} \frac{\kappa t - \mathsf{dbf}_W(t)}{\kappa(1 - \kappa)}$ or $\Pi_{\mathsf{opt}} \leq \mathsf{MaxResPeriod}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$. ∎

Example 3: Given a workload W with $\mathsf{CrT}_W = \{10\}$ and $\mathsf{dbf}_W(10) = 2$ as shown in Figure 3. Suppose that $\Gamma_c = (2, 1)$ is the current minimum bandwidth periodic resource model that can feasibly schedule W among all models with period $\Pi \leq 2$. In this case, $\kappa = 0.5$. The upper bound on the resource period is computed to be $\Pi_{\mathsf{opt}} \leq \mathsf{MaxResPeriod}(0.5, \mathsf{dbf}_W, \mathsf{LCM}_W) = \frac{0.5 \cdot 10 - 2}{0.5(1 - 0.5)} = 12$ by Theorem 1. The optimal periodic resource model for W is given by $\Gamma_{\mathsf{opt}} = (3, 1)$, which indeed satisfies Theorem 1.
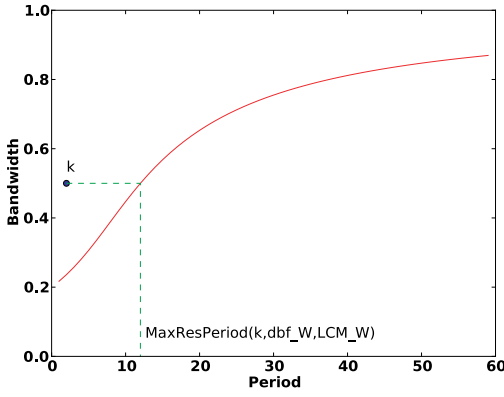
Fig. 3. The upper bound on the resource period in Example 3.

B. A revised algorithm using bound on resource period

Algorithm 2 gives an extension of Algorithm 1 by incorporating the upper bound on the resource periods $\mathsf{MaxResPeriod}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W))$ defined in Section III-A.

Algorithm 2 A revised algorithm using resource period bound.

Input: dbf_W, LCM_W for a workload W
Output: The optimal periodic resource model Γ for W
 1: **if** $\mathsf{dbf}_W(\mathsf{LCM}_W) \geq \mathsf{LCM}_W - 1$ **then**
 2:    $\Gamma = (1, 1)$
 3: **else**
 4:    $\Theta' = \mathsf{MinExec}(\mathsf{LCM}_W, \mathsf{dbf}_W, \mathsf{LCM}_W)$
 5:    $\kappa = \frac{\Theta'}{\mathsf{LCM}_W}$
 6:    $\Pi_{\max} = \mathsf{MaxResPeriod}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$
 7:    **for** $\Pi = 1$ to $\Pi_{\max}$ **do**
 8:       $\Theta = \mathsf{MinExec}(\Pi, \mathsf{dbf}_W, \mathsf{LCM}_W)$
 9:       **if** $\frac{\Theta}{\Pi} < \kappa$ **then**
10:          $\kappa = \frac{\Theta}{\Pi}$
11:          $\Gamma = (\Pi, \Theta)$
12:          $\Pi_{\max} = \min(\Pi_{\max}, \mathsf{MaxResPeriod}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W))$
13:       **end if**
14:    **end for**
15: **end if**

In Algorithm 2, Line 1-2 handles the special case $\mathsf{dbf}_W(\mathsf{LCM}_W) \geq \mathsf{LCM}_W - 1$, which has $\Gamma = (1, 1)$ as the minimum bandwidth resource model. This is because any resource model $\Gamma' = (\Pi, \Theta)$ that can feasibly schedule W must satisfy $2(\Pi - \Theta) \leq 1$ (due to $\mathsf{sbf}_{\Gamma'}(\mathsf{LCM}_W) \geq \mathsf{dbf}_W(\mathsf{LCM}_W) \geq \mathsf{LCM}_W - 1$) and hence $\Pi = \Theta$ (since $\Theta, \Pi \in \mathbb{N}$). In Line 4-5, Θ' denotes the minimum supply for $\Pi = \mathsf{LCM}_W$ and κ denotes the bandwidth of $(\mathsf{LCM}_W, \Theta')$. Since κ is not 1, we can find the initial $\Pi_{\max}$ in Line 6. The function $\mathsf{MinExec}(\Pi, \mathsf{dbf}_W, \mathsf{LCM}_W)$ (Line 4,8) is the same as in Algorithm 1. The function $\mathsf{MaxResPeriod}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$ in Line 6 and 12 computes the upper bound on the resource period as defined in Theorem 1. Finally, the minimum bandwidth acquired during algorithm execution is stored in κ and used to re-evaluate $\Pi_{\max}$ (Line 9-13).

Computation complexity. Observe that computing the set CrT_W has $\mathrm{O}\big(\min_{P_i \in W} P_i \cdot \mathsf{LCM}_W\big)$ time complexity, and $\Theta' = \mathsf{LCM}_W$ and $\Pi_{\max} = \min_{P_i \in W} P_i \cdot \mathsf{LCM}_W$ in the worse

case. Hence, the time complexity of Algorithm 2 is [13]: $\mathrm{O}\big(\min_{P_i \in W} P_i \cdot \mathsf{LCM}_W + \mathsf{LCM}_W \times (\min_{P_i \in W} P_i \cdot \mathsf{LCM}_W)\big)$, which is equal to $\mathrm{O}\big((\mathsf{LCM}_W)^2 \cdot \min_{P_i \in W} P_i\big)$.

IV. A NEW ALGORITHM FOR COMPUTING THE OPTIMAL PERIODIC RESOURCE MODEL

In this section, we present a new algorithm for computing the optimal resource model that is more efficient than the revised algorithm in the previous section. Observe that in searching for the optimal resource model for a workload W, Algorithm 2 iterates the resource period Π from 1 to the period bound $\Pi_{\max}$, which is computed using $\mathsf{MaxResPeriod}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$ and updated with respect to the minimum bandwidth κ obtained thus far. Since computing the resource execution time Θ for any given period Π has a constant time complexity, the algorithm's time complexity is proportional to the number of iterations of Π, which is $\mathsf{MaxResPeriod}(\kappa_0, \mathsf{dbf}_W, \mathsf{LCM}_W)$ in the worst case where $\kappa_0 = \mathsf{MinExec}(\mathsf{LCM}_W, \mathsf{dbf}_W, \mathsf{LCM}_W)/\mathsf{LCM}_W$. Since $\Theta \leq \Pi$, the upper bound on Θ will always be less than or equal to the upper bound on Π. Further, computing the resource period Π for any given Θ has the same time complexity as that of computing Θ from Π. As a result, we can reduce the search space by iterating Θ instead of Π.

Based on the above observation, Algorithm 3 gives a new procedure for computing the optimal resource model. We first explain the different steps involved in the algorithm and then present theoretical results supporting its correctness. Note that the result for the special case when $\mathsf{dbf}_W(\mathsf{LCM}_W) \geq \mathsf{LCM}_W - 1$ is $\Gamma = (1, 1)$ for the same reason as in Algorithm 2.

Algorithm 3 A new interface generation algorithm

Input: dbf_W, LCM_W for a workload W with $\mathsf{dbf}_W(\mathsf{LCM}_W) < \mathsf{LCM}_W - 1$
Output: The optimal periodic resource model Γ for W
 1: $\kappa = \mathsf{MinExec}(\mathsf{LCM}_W, \mathsf{dbf}_W, \mathsf{LCM}_W)/\mathsf{LCM}_W$
 2: $\Theta_{\max} = \mathsf{MaxResExec}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$
 3: **for** $\Theta = 1$ to $\Theta_{\max}$ **do**
 4:    $\Pi = \mathsf{MaxPeriod}(\Theta, \mathsf{dbf}_W, \mathsf{LCM}_W)$
 5:    **if** $\frac{\Theta}{\Pi} < \kappa$ **then**
 6:       $\kappa = \frac{\Theta}{\Pi}$
 7:       $\Gamma = (\Pi, \Theta)$
 8:       $\Theta_{\max} = \min(\Theta_{\max}, \mathsf{MaxResExec}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W))$
 9:    **end if**
10: **end for**

In Algorithm 3, the function $\mathsf{MinExec}(\Pi, \mathsf{dbf}_W, \mathsf{LCM}_W)$ (Line 1) is the same as in Algorithm 1. The variable κ (Line 1) indicates the bandwidth of $(\mathsf{LCM}_W, \mathsf{MinExec}(\mathsf{LCM}_W, \mathsf{dbf}_W, \mathsf{LCM}_W))$. The function $\mathsf{MaxResExec}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$ (Line 2 and 8) computes the upper bound of Θ based on Theorem 3. The initial value of $\Theta_{\max}$ is in Line 2. The function $\mathsf{MaxPeriod}(\Theta, \mathsf{dbf}_W, \mathsf{LCM}_W)$ (Line 4) computes – for any given Θ – an upper bound on the resource period Π of any resource model (Π, Θ) that can feasibly schedule W. The functions $\mathsf{MaxPeriod}(\Theta, \mathsf{dbf}_W, \mathsf{LCM}_W)$ and $\mathsf{MaxResExec}(\kappa,$

dbf_W, LCM_W) are computed as below.

Computation of $\mathsf{MaxPeriod}(\Theta, \mathsf{dbf}_W, \mathsf{LCM}_W)$**.** Theorem 2 gives the upper bound on the period Π of any resource model $\Gamma = (\Pi, \Theta)$ that can feasibly schedule W.

Theorem 2: Given a workload $W = \{T_1, T_2, \cdots, T_n\}$ where $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$. For any given Θ, the resource model $\Gamma = (\Pi, \Theta)$ can feasibly schedule W iff

$$\Pi \leq \min_{0 \leq t \leq \mathsf{LCM}_W} \mathsf{IntPeriod}(\Theta, t) \overset{\text{def}}{=} \mathsf{MaxPeriod}(\Theta, \mathsf{dbf}_W, \mathsf{LCM}_W)$$

where: $\mathsf{IntPeriod}(\Theta, t) \overset{\text{def}}{=} \begin{cases} \left\lfloor \frac{t+\Theta}{m-1} \right\rfloor, & \text{if } \mathsf{sbf}_\Gamma(t) \geq \mathsf{dbf}_W(t) \\ \left\lfloor \frac{t+\Theta}{m} \right\rfloor, & \text{otherwise} \end{cases}$

with $m = \left\lceil \frac{\mathsf{dbf}_W(t)+\Theta}{\Theta} \right\rceil$.

Before presenting the proof of Theorem 2, we state some notations. For any given Θ and any given t such that $1 \leq t \leq \mathsf{LCM}_W$, $\mathsf{Period}(\Theta, t)$ denotes a period value such that the resource model $R_{\Theta,t} = (\mathsf{Period}(\Theta, t), \Theta)$ satisfies $\mathsf{sbf}_{R_{\Theta,t}}(t) = \mathsf{dbf}_W(t)$. Then, the following corollary holds. Its proof can be found in [13].

Corollary 1: For all $\Theta \geq 0$, and all t s.t. $1 \leq t \leq \mathsf{LCM}_W$,

$$\left\lfloor \mathsf{Period}(\Theta, t) \right\rfloor = \mathsf{IntPeriod}(\Theta, t).$$

Proof of Theorem 2: ($\Rightarrow$) Recall the SBF of Γ defined in Equation 2. One can easily verify that for all Π_1, Π_2,

$$\Pi_1 \leq \Pi_2 \iff \mathsf{sbf}_{(\Pi_1,\Theta)}(t) \geq \mathsf{sbf}_{(\Pi_2,\Theta)}(t) \ \forall t \geq 0. \quad (11)$$

Suppose $\Gamma = (\Pi, \Theta)$ can feasibly schedule W, i.e.,

$$\forall \, 0 \leq t \leq \mathsf{LCM}_W : \mathsf{sbf}_\Gamma(t) \geq \mathsf{dbf}_W(t).$$

By definition, $\mathsf{dbf}_W(t) = \mathsf{sbf}_{R_{\Theta,t}}(t)$ where $R_{\Theta,t} = (\mathsf{Period}(\Theta, t), \Theta)$ for all $0 \leq t \leq \mathsf{LCM}_W$. Hence,

$$\forall \, 0 \leq t \leq \mathsf{LCM}_W : \mathsf{sbf}_\Gamma(t) \geq \mathsf{sbf}_{R_{\Theta,t}}(t). \quad (12)$$

Since Γ and $R_{\Theta,t}$ have the same execution time Θ, and due to Equation 11, Equation 12 is equivalent to

$$\forall \, 0 \leq t \leq \mathsf{LCM}_W : \Pi \leq \mathsf{Period}(\Theta, t).$$

Since $\Pi \in \mathbb{N}$, $\Pi \leq \mathsf{Period}(\Theta, t)$ is equivalent to $\Pi \leq \left\lfloor \mathsf{Period}(\Theta, t) \right\rfloor = \mathsf{IntPeriod}(\Theta, t)$ due to Corollary 1. Hence,

$$\Pi \leq \min_{0 \leq t \leq \mathsf{LCM}_W} \mathsf{IntPeriod}(\Theta, W, t)$$

($\Leftarrow$) Suppose $\Pi \leq \min_{0 \leq t \leq \mathsf{LCM}_W} \mathsf{IntPeriod}(\Theta, t)$. Then,

$$\Pi \leq \mathsf{IntPeriod}(\Theta, t) \leq \mathsf{Period}(\Theta, t), \ \forall \, 0 \leq t \leq \mathsf{LCM}_W.$$

Denote $\Pi_t = \mathsf{Period}(\Theta, t)$. Apply Equation 11, we have

$$\forall \, 0 \leq t \leq \mathsf{LCM}_W, \ \mathsf{sbf}_{(\Pi,\Theta)}(t) \geq \mathsf{sbf}_{(\Pi_t,\Theta)}(t).$$

Since $\mathsf{sbf}_{(\Pi_t,\Theta)} = \mathsf{dbf}_W(t)$ by the definition of Π_t, we imply $\mathsf{sbf}_{(\Pi,\Theta)}(t) \geq \mathsf{dbf}_W(t)$ for all $0 \leq t \leq \mathsf{LCM}_W$. In other words, $\Gamma = (\Pi, \Theta)$ can feasibly schedule W. ∎

Computation of $\mathsf{MaxResExec}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$**.** For any given current minimum bandwidth κ at some intermediate execution step of Algorithm 3, we can compute the upper bound $\mathsf{MaxResExec}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$ on the value Θ of the

optimal resource model $\Gamma_{\mathsf{opt}} = (\Pi_{\mathsf{opt}}, \Theta_{\mathsf{opt}})$ in a similar fashion as done in Section III-A.

Theorem 3: Given a workload $W = \{T_1, T_2, \cdots, T_n\}$ with $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$. Suppose $\Gamma_c = (\Pi_c, \Theta_c)$ is the current periodic resource model obtained at some intermediate execution step of Algorithm 3. Then, the optimal periodic resource model $\Gamma_{\mathsf{opt}} = (\Pi_{\mathsf{opt}}, \Theta_{\mathsf{opt}})$ for W satisfies $\Theta_c \leq \Theta_{\mathsf{opt}} \leq \mathsf{MaxResExec}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$ with $\kappa = \frac{\Theta_c}{\Pi_c}$,

$$\mathsf{MaxResExec}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W) \overset{\text{def}}{=} \min_{t \in \mathsf{CrT}_W} \frac{\kappa t - \mathsf{dbf}_W(t)}{1 - \kappa}.$$

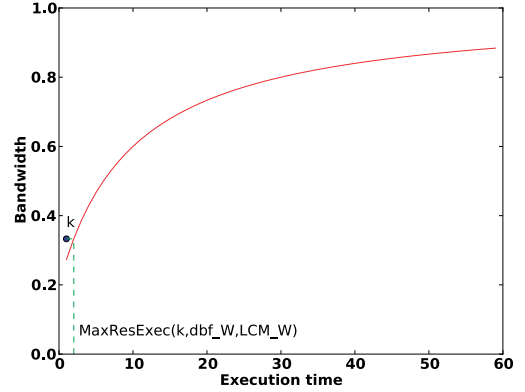*Proof:* The proof is establish based on similar arguments as that of Theorem 1. Its details can be found in [13]. ∎

Fig. 4. The upper bound on the resource execution units in Example 4.

Example 4: Given a workload W with $\mathsf{CrT}_W = \{10\}$ and $\mathsf{dbf}_W(10) = 2$. Suppose that $\Gamma_c = (3, 1)$ is the current minimum-bandwidth periodic resource model that can feasibly schedule W given by Algorithm 3. In this case, $\kappa = \frac{1}{3}$. The upper bound on the resource execution units is computed using Theorem 3 to be $\Theta_{\mathsf{opt}} \leq \mathsf{MaxResExec}(\frac{1}{3}, \mathsf{dbf}_W, \mathsf{LCM}_W) = \frac{\frac{1}{3}10 - 2}{1 - \frac{1}{3}} = 2$. As illustrated in Figure 4, the optimal resource model for W is $\Gamma_{\mathsf{opt}} = (3, 1)$, which satisfies Theorem 3.

Computation complexity compared to Algorithm 2. From Theorem 3 and 1, $\mathsf{MaxResExec}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W) = \kappa \mathsf{MaxResPeriod}(\kappa, \mathsf{dbf}_W, \mathsf{LCM}_W)$. Since $\kappa \leq \frac{\mathsf{LCM}_W - 1}{\mathsf{LCM}_W}$, one can hence verify that Algorithm 3 is at least $\frac{\mathsf{LCM}_W}{\mathsf{LCM}_W - 1}$ times faster than Algorithm 2 [13].

V. DUAL PERIODIC RESOURCE MODEL

A. Overhead of periodic resource interface with integer values

When assuming rational parameter values for resource interfaces, the periodic resource interface with period of 1 and execution time equal to the utilization of the workload always has the minimum bandwidth among that of all resource interfaces [3]. However, this optimality of periodic resource model is no longer achievable when it is restricted to have only integer parameters. As an example, consider a workload W composed of only one task $T = (5, 1)$. The ideal minimum bandwidth resource interface (i.e., with rational parameter values) given by Algorithm 1) is $(1, 0.2)$. Hence, the minimum bandwidth of W is 0.2. On the other hand, the minimum

bandwidth resource interface with integer parameter values for W (given by Algorithm 2) is $(3,1)$, which has a bandwidth of $\frac{1}{3}$. Thus, the minimum bandwidth periodic resource interface with integer parameter values incurs at least 66% overheads compared to the ideal one with rational parameter values. By the same reason, the new algorithm (Algorithm 3) also experiences similar bandwidth overhead.

The above overhead introduced by the integer constraints has prompted a need for new resource interfaces with integer parameters and their associated interface computation techniques that can achieve better resource utilization than the periodic resource interface do. In the coming sections, we present such an interface and its computation. Here, we discuss the computation for leaf-components only; the computation for non-leaf components can be established using a similar technique as in the case of periodic resource interface [1].

B. Dual periodic resource model (DPRM)

A dual periodic resource model (DPRM) interface is defined by $\Omega = (\Gamma_1, \Gamma_2)$ where Γ_1 and Γ_2 are periodic resource models. Semantically, each DPRM offers the same amount of resource as the total resource units given by the two resource models Γ_1 and Γ_2. Thus, its bandwidth is given by $\text{bw}_\Omega = \text{bw}_{\Gamma_1} + \text{bw}_{\Gamma_2}$. Its SBF and schedulability condition are given by Lemma 5 and 6, respectively, whose proofs are available in [13].

Lemma 5: The SBF of a DPRM $\Omega = (\Gamma_1, \Gamma_2)$ where $\Gamma_1 = (\Pi_1, \Theta_1)$ and $\Gamma_2 = (\Pi_2, \Theta_2)$ is given by:

$$\text{sbf}_\Omega(t) = \text{sbf}_{\Gamma_1}(t) + \text{sbf}_{\Gamma_2}(t), \ \forall t \geq 0. \tag{13}$$

Lemma 6: Given a component $C = (W, \Omega, EDF)$ where $W = \{T_1, T_2, \ldots, T_n\}$, $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$, and $\Omega = (\Gamma_1, \Gamma_2)$ is a DPRM interface. C is schedulable under Ω iff

$$\forall t \ s.t. \ 0 \leq t \leq \text{LCM}_W, \ \text{dbf}_W(t) \leq \text{sbf}_\Omega(t). \tag{14}$$

DPRM interface computation. Given a component with workload W that is scheduled under EDF, we would like to compute the optimal DPRM interface $\Omega = (\Gamma_1, \Gamma_2)$ for W (i.e., Ω has the minimum bandwidth among that of all DPRM interface $\Omega' = (\Gamma_1', \Gamma_2')$ that can feasibly schedule W). The basic idea is to iterate the period Π_1 and execution time Θ_1 of the first element Γ_1 as well as the periodic Π_2 of the second element Γ_2 of the DPRM interface $\Omega = (\Gamma_1, \Gamma_2)$. For each running Π_1, Θ_1 and Π_2, we compute the corresponding Θ_2 such that $\Omega = \big((\Pi_1, \Theta_1), (\Pi_2, \Theta_2)\big)$ has the minimum bandwidth among that of all DPRM interface $\Omega' = \big((\Pi_1, \Theta_1), (\Pi_2, \Theta_2')\big)$ that can feasibly schedule W. We then keep track of the interface Ω with the minimum bandwidth during our iteration.

Algorithm 4 shows the procedure for computing the minimum-bandwidth DPRM interface Ω. The functions MinExec(Π, dbf_W, LCM_W) and MaxResPeriod(κ, dbf_W) (Line 1 and 2, respectively) are the same as in Algorithm 3. The function MaxResExecDPRM(Π, dbf_W, LCM_W) (Line 4) gives an upper bound on the value of Θ_1. The function getResModel(dbf_W, LCM_W, Γ_1) (Line 7) gives the optimal

periodic resource model for the remaining resource demand of W after W has been served by the resource model Γ_1. These two new functions are computed as below.

Algorithm 4 DPRM interface computation

Input: dbf_W, LCM_W for a workload W with
$\qquad \text{dbf}_W(\text{LCM}_W) < \text{LCM}_W - 1$
Output: The optimal DPRM $\Omega = (\Gamma_1, \Gamma_2)$ for W
1: $\kappa = \text{MinExec}(\text{LCM}_W, \text{dbf}_W, \text{LCM}_W)/\text{LCM}_W$
2: $\Pi_1^{\max} = \text{MaxResPeriod}(\kappa, \text{dbf}_W, \text{LCM}_W)$
3: **for** $\Pi_1 = 1$ to $\Pi_1^{\max}$ **do**
4: $\quad \Theta_1^{\max} = \text{MaxResExecDPRM}(\Pi, \text{dbf}_W, \text{LCM}_W)$
5: $\quad$ **for** $\Theta_1 = 1$ to Θ_1^{max} **do**
6: $\qquad \Gamma_1 = (\Pi_1, \Theta_1)$
7: $\qquad \Gamma_2 = \text{getResModel}(\text{dbf}_W, \text{LCM}_W, \Gamma_1)$
8: $\qquad$ **if** $\text{bw}_{\Gamma_1} + \text{bw}\Gamma_2 < \kappa$ **then**
9: $\qquad\quad \kappa = \text{bw}_{\Gamma_1} + \text{bw}_{\Gamma_2}$
10: $\qquad\quad \Omega = (\Gamma_1, \Gamma_2)$
11: $\qquad\quad \Pi_1^{max} = \min(\Pi_1^{max}, \text{MaxResPeriod}(\kappa, \text{dbf}_W, \text{LCM}_W))$
12: $\qquad$ **end if**
13: $\quad$ **end for**
14: **end for**

Computation of MaxResExecDPRM(Π, dbf_W, LCM_W)**.** Given any Π_1, Θ_1 and Π_2, the execution time Θ_2 of Ω is determined such that the resource supplied by the periodic resource model (Π_2, Θ_2) must be at least equal to the remaining demand of the workload W after W has been serviced by (Π_1, Θ_1). Towards this, we define the *remaining demand bound function* (RDBF) as below.

Definition 3: Given a workload $W = \{T_1, T_2, ..., T_n\}$ with $1 \leq i \leq n$. The RDBF of W after being serviced by a resource model R, denoted by $\text{rdbf}_{W-R}(t)$, specifies the maximum number of remaining execution units required by W in any time interval of length t after W has been serviced by W.

One can easily verify that

$$\forall t \geq 0: \ \text{rdbf}_{W-R}(t) = \max\big(0, \text{dbf}_W(t) - \text{sbf}_R(t)\big). \tag{15}$$

Lemma 7 gives the schedulability condition for W under Ω. Its proof comes directly from Lemma 6 and Equation 15.

Lemma 7: Given a component $C = (W, \Omega, EDF)$ where $W = \{T_1, T_2, \ldots, T_n\}$, $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$, and $\Omega = (\Gamma_1, \Gamma_2)$ is a DPRM. Then, C is schedulable under Ω (Ω can feasibly schedule W) iff

$$\forall t \ s.t. 0 < t \leq \text{LCM}_W, \ \text{rdbf}_{W-\Gamma_1}(t) \leq \text{sbf}_{\Gamma_2}(t). \tag{16}$$

We define the LSBF (lower supply bound function) of a periodic resource model Γ to be the linear function with the smallest slope that lower bounds sbf_Γ, given by [1]:

$$\forall t \geq 0: \ \text{lsbf}_\Gamma(t) = \max\left(\frac{\Theta}{\Pi}\big(t - 2(\Pi - \Theta)\big), 0\right). \tag{17}$$

The maximum value of Θ_1 in the optimal DPRM interface $\Omega = (\Gamma_1, \Gamma_2)$ with $\Gamma_1 = (\Pi_1, \Theta_1)$ can now be computed using function MaxResExecDPRM(Π, dbf_W, LCM_W) defined in the following theorem.

Theorem 4: Given a workload $W = \{T_1, T_2, \cdots, T_n\}$, with $T_i = (p_i, e_i)$ for all $1 \leq i \leq n$. For any

given Π_1, the minimum bandwidth DPRM interface $\Omega = \big((\Pi_1, \Theta_1), (\Pi_2, \Theta_2)\big)$ for W satisfies $\Theta_1 \leq \Theta_1^{\max}$ where

$$\Theta_1^{\max} = \max_{t \in \mathsf{CrT}_W} \frac{(2\Pi_1 - t) + \sqrt{(2\Pi_1 - t)^2 + 8\Pi_1 d_t}}{4}.$$

Proof: The proof is similar to that of Theorem 1 except that it is based on LSBF instead of USBF. The details are available in [13]. ∎

Computation of getResModel($\mathrm{dbf}_W, \mathsf{LCM}_W, \Gamma_1$)**.** The function getResModel($\mathrm{dbf}_W, \mathsf{LCM}_W, \Gamma_1$) computes a period resource model Γ_2 such that $\Omega = (\Gamma_1, \Gamma_2)$ is the minimum bandwidth DPRM interface that can schedule W. This Γ_2 can be obtained as the output of Algorithm 3 on the inputs $\mathrm{rdbf}_{W-\Gamma_1}(t)$ and LCM_W. The correctness of this computation is justified in [13].

VI. Simulation

To evaluate our algorithms and DPRM interface, we ran simulations on 200 random workloads, each consisting of three tasks. Each task's period was randomly chosen in the range of 10-100 following the uniform distribution. Each task's execution time was uniformly distributed random number from 1 to the task's period. We constrained the workload utilization to be no more than 0.8.
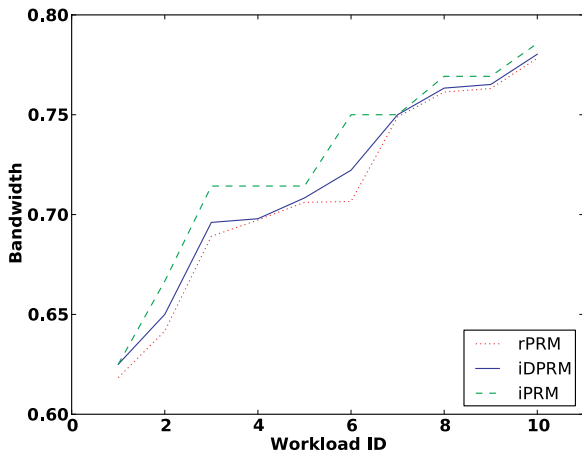

Fig. 5.   Comparison between Periodic Resource Model and DPRM.

For each generated workload, we computed its optimal periodic resource model (iPRM) and its optimal DPRM (iD-PRM) with integer parameters. We also computed the optimal periodic resource model with rational number (rPRM) [3].

Figure 5 shows the results of the first ten workloads. The X-axis is the workload identifier sorted by utilization whereas the Y-axis is the optimal bandwidth of the computed resource models. As shown in the figure, the iDPRM was always better than or as good as the iPRM: the iDPRM had smaller bandwidth than the iPRM did in 77% of the simulated workloads, with a bandwidth reduction of up to 12.5%. Further, with respect to the ideal bandwidth given by the rPRM, the iDPRM incurred only 1.25% bandwidth overhead in average whereas the iPRM suffered more than 2.56 times as much (3.22% overhead).

To evaluate the scalability of DPRM interface, we repeated the above experiment for larger workloads. Our simulation results showed that as the number of tasks increases, the above improvement of DPRM interface (over the periodic resource interface) also increases [13]. This is expected due to the corresponding increase in complexity of the DBF function of the workload, which can be more effectively captured by the DPRM interface.

VII. Conclusion

Traditional algorithms for computing the minimum-bandwidth resource model face two drawbacks: (i) they assume rational parameters for the resource model, which cannot always be used in practice, and (ii) the resource period is searched within a range specified by the designer, which cannot guarantee optimality. We have presented more efficient algorithms that tackle these drawbacks by considering integer parameters and a safe bound on the period. We further proposed the DPRM interface and an algorithm for computing the minimum bandwidth DPRM interface that is more accurate than the periodic resource interface when restricting the interface to have only integer parameters.

Our simulation results showed that the DPRM achieved a lower bandwidth than the periodic model did in 77% of the workloads. DPRM further reduced more than half the bandwidth overheads suffered by the periodic resource model. This advantage of DPRM interface over the periodic model interface was also shown to scale to the size of the workload.

References

[1] I. Shin and I. Lee, "Periodic resource model for compositional real-time guarantees," in *RTSS*, 2003, p. 2.
[2] A. K. Mok, X. A. Feng, and D. Chen, "Resource partition for real-time systems," in *RTAS*, 2001, p. 75.
[3] A. Easwaran, M. Anand, I. Lee, L. T. X. Phan, and O. Sokolsky, "Simulation relations, interface complexity, and resource optimality for real-time hierarchical systems," in *Workshop on Reconciling Performance with Predictability*, 2009.
[4] Z. Deng and J.-S. Liu, "Scheduling real-time applications in an open environment," in *RTSS*, 1997, pp. 308–319.
[5] A. Easwaran, M. Anand, and I. Lee, "Compositional analysis framework using EDP resource models," *RTSS*, pp. 129–138, 2007.
[6] S. Matic and T. A. Henzinger, "Trading end-to-end latency for composability," in *RTSS*, 2005, pp. 99–110.
[7] M. Behnam, I. Shin, T. Nolte, and M. Nolin, "SIRAP: a synchronization protocol for hierarchical resource sharing in real-time open systems," in *EMSOFT*, 2007, pp. 279–288.
[8] R. Kaiser, "Alternatives for scheduling virtual machines in real-time embedded systems," in *IIES*, 2008, pp. 5–10.
[9] T. Cucinotta, G. Anastasi, and L. Abeni, "Respecting temporal constraints in virtualised services," in *COMPSAC*, 2009, pp. 73–78.
[10] S. K. Baruah, L. E. Rosier, and R. R. Howell, "Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor," *Real-Time Syst.*, vol. 2, no. 4, pp. 301–324, 1990.
[11] I. Shin and I. Lee, "Compositional real-time scheduling framework with periodic model," *ACM Trans. Embed. Comput. Syst.*, vol. 7, no. 3, pp. 1–39, 2008.
[12] A. Easwaran, I. Lee, I. Shin, and O. Sokolsky, "Compositional schedulability analysis of hierarchical real-time systems," in *ISORC*, 2007, pp. 274–281.
[13] J. Lee, L. T. Phan, S. Chen, O. Sokolsky, and I. Lee, "Dual periodic resource model," *Tech Report MS-CIS-10-31*, University of Pennsylvania. [Online]. Available: http://repository.upenn.edu/cis_reports/940