Hierarchical Scheduling

8

Jin Hyun Kim, Deepak Gangadharan, Kyong Hoon Kim, Insik Shin, and Insup Lee

Contents

J. H. Kim
Department of Information and Communication Engineering, Gyeongsang University, Jinju, Gyeongnam, South Korea
e-mail: jinhyun@seas.upenn.edu; jin.kim@gnu.ac.kr

D. Gangadharan · I. Lee (✉)
Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA
e-mail: deepakg@seas.upenn.edu; lee@cis.upenn.edu

K. H. Kim
Department of Informatics, Gyeongsang National University, Jinju, Gyeongnam, South Korea
e-mail: khkim@gnu.ac.kr; kyongkim@seas.upenn.edu

I. Shin
Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Yuseong-gu, South Korea
e-mail: insik.shin@gmail.com

Y.-C. Tian, D. C. Levy (eds.), *Handbook of Real-Time Computing*,
https://doi.org/10.1007/978-981-287-251-7_6

Abstract

This chapter presents two hierarchical scheduling analysis paradigms for a uniprocessor hierarchical scheduling system: compositional framework (CF) and real-time calculus (RTC). Each paradigm uses different techniques for resource models and schedulability analysis: CF uses supply-bound functions (sbf) and demand-bound functions (dbf), whereas RTC computes a lower-bound of service curve that satisfies the demand of a given workload. This chapter describes both CF and RTC approaches and various schedulability analysis techniques for hierarchical scheduling systems. These techniques are described based on bounded delay resource model, periodic resource model, explicit deadline periodic model, and arrival and service curves. Finally, this chapter also describes optimality results and compares CF and RTC approaches.

1 Introduction

The increasing complexity of real-time systems such as Cyber-Physical Systems (CPS) demands scalable design and analysis methods to assure various properties of the systems. Large-scale complex real-time systems are developed using components that are integrated horizontally and vertically. Such component-based approach can be effective in coping with scalability based on the notion of compositionality, which is one of the most fundamental principles (e.g., dividend conquer) used in computer and network engineering to deal with large-scale systems. To address one of the scalability challenges of complex real-time systems, researchers have been developing the concept of compositional real-time scheduling theory to support compositional schedulability analysis. The compositional framework developed by Shin and Lee (2003) is based on the demand-supply interfaces that hide implementation details, capturing only essential timing and resource requirements of components. Here, the component computational demand can be abstracted to a scheduling demand interface such that schedulability analysis of composed components simplifies to checking if there are sufficient resource supplies to meet the demands of the components. The salient aspect of the work is that global timing properties can be established by combining local timing properties of subsystems.

This compositional reasoning approach has resulted in design-analysis tools (e.g., (Chadli et al. 2016; Phan et al. 2011) to tackle the growing complexity of real-time systems: the analysis of a complex system can be achieved by first defining interfaces for its subsystems, which capture their resource and timing requirements while hiding their complex internal details, and then reasoning about the composition of their interfaces. This approach can also used hierarchically by reducing each subsystem to a simple interface, with significant reduction in the complexity of the system analysis at higher layers of integration.

A hierarchical scheduling system (HSS) comprises one or more components in a scheduling hierarchy. Each component recursively encapsulates one or more child components and has its own scheduling algorithm. Resources are supplied from a single parent component to multiple child components. A child component exposes its resource requirements through an interface to its enclosing-level component, called parent component. A parent component executes one or more resource-supply tasks that individually allocate resources to their child components. The resource demand of a component is satisfied by a resource supply of its parent component, meaning that all tasks in the component are schedulable, i.e., no task enclosed by the component misses its deadline. The compositional framework has been developed to support various ways for building and analyzing schedulable components hierarchically. It is based on interface models that capture the resource demand and supply of a component.

The rest of this chapter is organized as follows: Sect. 2 formally defines the HSS and presents properties to be analyzed in the compositional framework. Sect. 3 describes the compositional framework analysis techniques based on various resource models. In Sect. 4, we present RTC-based compositional analysis techniques. Sect. 5 compares the compositional framework and RTC-based techniques.

2 Uniprocessor Hierarchical Scheduling Systems

This section presents the notion of a hierarchical scheduling system (HSS) and then describes steps to construct a schedulable HSS. We also present analysis techniques that are necessary for constructing a schedulable HSS. An HSS consists of a set of components, each of which is a scheduling system. Each component uses its own scheduling algorithm to run a set of tasks. The scheduling system of an HSS differs from the classical scheduling system in that it uses the notion of a hierarchical resource share between higher and lower-level components in a vertical way. In the classical non-hierarchical scheduling system, a resource is shared horizontally by a set of tasks on the same level. In an HSS, a resource is shared vertically by different level of system, so a component, except the root component, can run only while its parent component is running.

A component $M = (W, A)$ consists of a workload W, a set of tasks, and a scheduling algorithm A. The interface I of a component abstracts the resource demand of all workload tasks in a component. Its purpose is to capture the demand of a component and represent the demand, while hiding task-level details, to its parent component.

A scheduling unit S is defined by a triple (W, R, A) with workload W, resource supply R, and scheduling algorithm A. As we will see later in this chapter, there are many resource modeling paradigms for describing resource supply. There are three resource modeling paradigms considered in this chapter: Bounded-Delay Resource (BDR) model, Periodic Resource Model (PRM), and Explicit Deadline Periodic (EDP) resource model. We use X for resource modeling paradigm. A resource supply task R is thus an instance of X with concrete parameters of the resource

Table 1 Symbol definitions

Symbol	Definition
A	Scheduling algorithm
$\tau_i = (C_i, T_i, D_i)$	Explicit deadline task i with the C_i worst-case execution time, T_i period, and D_i deadline
$\tau_i = (C_i, T_i)$	Implicit deadline task i with the C_i worst-case execution time and T_i period. It is assumed that $C_i \le T_i$
$W = \{\tau_i\}$	Workload with a set of tasks τ_i. It is assumed that there are finite number of tasks
$M = (W, A)$	Component of a hierarchical scheduling system with W scheduled by A
$X = \{\text{BDR}, \text{PRM}, \text{EDP}\}$	The set of resource modeling paradigms considered in this chapter
$S = (W, R, A)$	Scheduling unit, where R is a resource supply represented using the resource modeling paradigm X
BDR, $R = (\alpha, \Delta)$	Bounded-Delay Resource model (BDR) R with α availability factor and Δ partition delay
PRM, $R = (\Theta, \Pi)$	Periodic Resource Model (PRM) R with the Θ worst-case execution time and Π period
EDP, $R = (\Theta, \Pi, \Delta)$	Explicit Deadline Periodic (EDP) resource model R with the Θ worst-case execution time, Π period, and Δ deadline
I	Component interface
$\text{sbf}_X(R, t)$	Supply bound function of the resource supply R in model X
$\text{dbf}_A(W, t)$	Demand bound function of the demand of W under scheduling algorithm A
U_W	Utilization of workload W

modeling paradigm X. A resource modeling paradigm is also used to represent the interface I of a component.

A scheduling unit $S = (W, R, A)$ is said to be schedulable if no task of W scheduled by A under the resource supply of R misses the deadline. Note that this paper assumes that the workload W and the resource supply task R are not synchronized. In other words, whenever the resource is provided, the workload needs to use it; Otherwise, the provided resource is wasted. Table 1 lists the symbols used in this chapter.

Figure 1 presents an example of an HSS, where each component consists of periodic tasks that are scheduled according to the scheduling algorithm of the component. The interface of the component captures the resource requirements of its tasks. For instance, I_1 of M_1 represents the resource requirements of τ_3, τ_4, and τ_5 when they are scheduled by RM. The parent component M_0 converts the interface I_1 to a resource supply task τ_1 such that when M_0 schedule τ_1 for execution, it is sufficient to meet the resource demanded by I_1. The interface I_2 and the task τ_2 are similarly related.

We say that a component is schedulable if all tasks in the component never miss deadlines. In HSS, the schedulability of components is achieved recursively starting from the root component. If the parent component M_0 can schedule τ_1 and τ_2 under

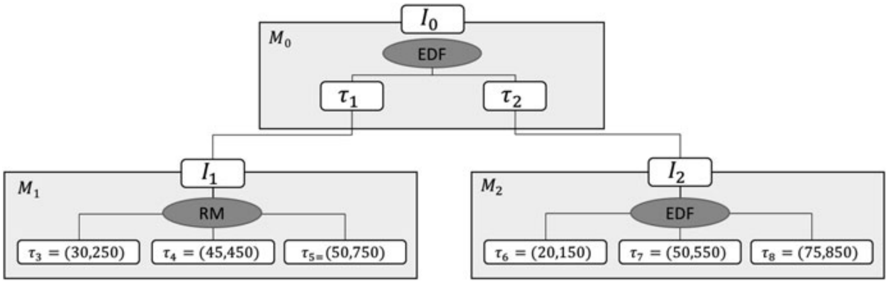


Fig. 1 The running example of a hierarchical scheduling system

EDF scheduling algorithm, the child component M_1 whose resource is supplied by τ_1 successfully schedules τ_3, τ_4, and τ_5 using RM. Similarly, M_2 is schedulable if τ_2 is schedulable. Consequently, the schedulability of the whole HSS in Fig. 8 is reduced to checking the schedulability of M_0.

Our schedulability analysis is based on the demand bound and supply bound functions and Eq. 1. For a component (W, A), the demand-bound function $\mathrm{dbf}_A(W, t)$ computes the maximum resource demand by W under A for a time interval t. For a resource supply task R, the supply bound function $\mathrm{sbf}_X(R, t)$ computes the minimum possible resource supply that X provides during a time interval t. Using these two functions, a scheduling unit is proven schedulable by showing that the minimum possible resource supply satisfies the maximum possible resource demand over all possible time as follows:

$$\forall t \geq 0 \ \mathrm{dbf}_A\left(W, t\right) \leq \mathrm{sbf}_X\left(R, t\right) \tag{1}$$

The effectiveness of HSS in terms of resource utilization depends on resource model paradigms. This chapter explains various compositional analysis techniques, which are based on three resource models, Bounded-Delay Resource Model (BDR), Periodic Resource Model (PRM), and Explicit-Deadline Periodic resource model (EDP).

3 Compositional Framework for HSS

This section explains the compositional framework in terms of the demand abstraction and resource supply patterns. Since the resource demand of a component depends on a scheduling algorithm, we first explain the demand bound functions for EDF and RM. We then explain how to derive resource supply based on resource modeling paradigms.

3.1 Demand Bound Functions for EDF and RM

The notion of demand bound function (dbf) is commonly used to compute the maximum resource demand for a set of tasks executing under a scheduling algorithm. For an implicit-deadline periodic task set $W = \{(C_i, T_i)\}$ under EDF and a time interval t, the demand-bound function $\mathrm{dbf}_{\mathrm{EDF}}(W, t)$ computes the worst-case resource demand of W (Baruah et al. 1990) as follows:

$$\mathrm{dbf}_{\mathrm{EDF}}\,(W, t) = \sum_{(C_i, T_i)\in W} \left\lfloor \frac{t}{T_i} \right\rfloor \cdot C_i \tag{2}$$

where $\mathrm{dbf}_{\mathrm{EDF}}(W, t)$ computes cumulative resources demanded by all tasks of W during the time interval t.

For an explicit-deadline periodic task set $W = \{(C_i, T_i, D_i)\}$, the demand-bound function $\mathrm{dbf}_{\mathrm{EDF}}(W, t)$ is defined in Baruah et al. (1990) as follows:

$$\mathrm{dbf}_{\mathrm{EDF}}\,(W, t) = \sum_{\tau_i\in W} \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor \cdot C_i \tag{3}$$

For a given implicit-deadline periodic task set $W = \{(C_i, T_i)\}$ under RM and a time interval t, the demand-bound function $\mathrm{dbf}_{\mathrm{RM}}(W, t, i)$ computes the worst-case demand of task i that includes the cumulative resource demands of higher priority tasks (Lehoczky et al. 1989). It is defined as follows:

$$\mathrm{dbf}_{\mathrm{RM}}\,(W, t,\ \ i) = C_i + \sum_{(C_k, T_k)\in H\ P_W(i)} \left\lceil \frac{t}{T_k} \right\rceil .C_k \tag{4}$$

where $\mathrm{HP}_W(i)$ denotes tasks whose priorities in W are higher than task i's priority.

3.2 Interface and Resource Supply Task in Compositional Framework

In the literature (Easwaran et al. 2007; Mok et al. 2001; Shin and Lee 2003), a resource model is used to denote a resource provided with a certain supply pattern to a sub-component running within a component. Another notion is the interface of a component for resource requirements of a component that needs to be met by its resource-supplying parent component. In some papers, the interface is also used as a resource supplier for a child component, which make distinctions vague.

To avoid this confusion, this chapter was organized and written to clearly distinguish between the component interface and the resource supply task. Here, a component interface is used to capture the resource requirement of a component, while a resource supply task denotes a task of the parent component that provides resources to a child component. In HSS, a component can contain also other components in addition to its tasks. In principle, an interface of a sub-component or child component is obtained by computing the collective resource requirements of tasks in a workload. Then, the resource supply task to service the child component is obtained based on the interface.

There are two approaches for deriving an interface of a component depending on resource modeling paradigms. The first approach is to derive an interface with consideration of the resource modeling paradigm being used. Given a component $M = (W, A)$ and a resource modeling paradigm X, a resource supply task R is derived so that the scheduling unit $S = (W, R, A)$ is schedulable. Here, we can use R as the interface (i.e., $I = R$) of the component. This approach is used with the BDR and PRM resource modeling paradigms in Sects. 3.3.3 and 3.4.3, respectively.

The second approach is to derive an interface without consideration of the resource modeling paradigm being used. For a given component $M = (W, A)$, an interface I is generated so that I combines all the real-time requirements of tasks in W executed using the scheduling algorithm A. Then, with consideration of the resource modeling paradigm, the parent component finds a resource supply task R that can satisfy the interface I. The approach is used with the EDP resource modeling paradigm as described in Sect. 3.5.3. In the EDP resource modeling paradigm, the interface and the resource supply task are separately computed so that the interface and the corresponding resource supply task can be different (i.e., $I \neq R$).

This chapter was written to clarify these difference as much as possible and also try to unify notations used in the different papers. For this reason, we have slightly modified the definitions and theorem statements to be consistent with notations used in this chapter.

3.3 Compositional Framework Based on Bounded Delay Resource Model

Traditional real-time system models and scheduling problems (Liu and Layland 1973) assume that the CPU resources are dedicated exclusively to execute a set of tasks. This assumption, however, implies that if a system consists of multiple components, the tasks of all the components need to be considered together for schedulability analysis. In order to preserve components during schedulability analysis, the notion of real-time virtual resource was introduced in Mok et al. (2001), called *static resource partition* (SRP) model which assigns a static temporal resource partitioning to each component. Here, each component has its own set of tasks and is scheduled by its own scheduler as if it has exclusive access to the physical resource. Then, the second level scheduler co-ordinates the sharing of the resource by the components. This enables separation of concerns (Mok et al. 2001), i.e., scheduling at the resource partition level and scheduling at the task level can be considered separately.

Since the SRP model fixes the resource partitioning statically, the second-level (or partition-level) scheduler has little flexibility in resource allocation other than shifting the partition. In order to overcome this inflexibility, Mok et al. (2001) generalize the SRP model in a way that the time intervals of a resource partition are not explicitly specified. They provide the bounded-delay resource (BDR) model, where each component accesses a virtual resource that operates at a fraction of the rate at which the physical resource operates. If infinite time slicing is possible, then each component has exclusive access to the virtual resource with uniform rate and zero delay bound. In practice, this is not achievable because small delay bounds incur large partition scheduling overheads and therefore delay bounds should not be smaller than what is necessary to accommodate jitter. The BDR model bounds this type of uncertainty with the delay bound D, which is the maximum time a component may have to wait to get access to the physical resource over any time interval. This means that if an event e occurs x time units from another event e', assuming that the virtual resource operates at uniform rate, then e and e' will be at most $x + D$ time units apart in real time.

3.3.1 Static Resource Partition Model

Mok et al. (2001) introduce the notion of resource partition that assigns time intervals to partitions.

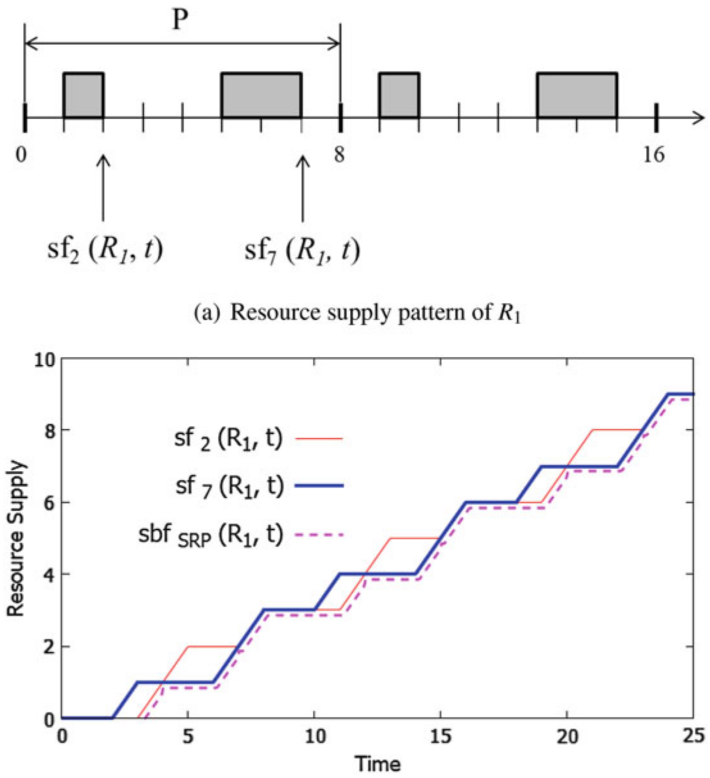
Definition 1 Mok et al. (2001) A static resource partition (SRP) is a tuple (Γ, P), where P is the partition period and Γ is a sequence of N time pairs $\{(S_1, E_1), (S_2, E_2), \ \ldots, \ (S_N, E_N)\}$ that satisfies $(0 \leq S_1 < E_1 < S_2 < E_2 < \ \ldots \ < S_N < E_N \leq P$ for some $N \geq 1$. $(S_1, S_2, \ \ldots, S_N)$ and $(E_1, E_2, \ \ldots, E_N)$ are start times and end times, respectively, for access to the physical resource. The physical resource is available to the component only during the intervals $(S_i + j \times P, E_i + j \times P)$, where $1 \leq i \leq N, \ \ j \geq 0$.

The following example illustrates the concept of resource partition.

Example 1 A SRP, $R_1 = (\{(1, 2), \ (5, 7)\}, \ 8)$, is a resource partition with the period equal to 8 and this partition has access to the physical resource from 1 to 2 time units and from 5 to 7 time units. The resource-supply pattern is repeated every 8 time units. Figure 2a illustrates this partition.

Since a SRP, $R = (\Gamma, P)$, repeats the same resource supply pattern Γ every P time units, the resource supply function of R starting from t_0 to t, denoted as $\mathrm{sf}_{t_0}(R, t)$, is determined by the pattern. For example, consider the supply function starting from time 2 (shown in Fig. 2a). After the delay of 3 time units, the resource is supplied for 2 time units. Then, after 2 time-unit delay, the resource is supplied for one time unit. Since this pattern is repeated, $\mathrm{sf}_2(R_1, t)$ can be computed, which is plotted as shown in Fig. 2b. Similarly, $\mathrm{sf}_7(R_1, t)$ is drawn in Fig. 2b.

For each resource supply interval $(S_i, \ E_i)$ of the SRP, the worst-case supply occurs when the supply starts at the end of the previous resource partition. The supply bound function (In Mok et al. (2001), it is called *the least supply function* and denoted as $S^*(t)$) is derived from the minimum of all supply functions as follows:



(a) Resource supply pattern of R_1

(b) sf and sbf functions of R_1

Fig. 2 An example SRP $R_1 = (\{(1,2),\ (5,7)\},\ 8)$

$$\text{sbf}_{\text{SRP}}\,(R,t) = \min_{i=1}^{N}\left\{\text{sf}_{E_i}\,(R,t)\right\} \tag{5}$$

In Example 1, the SRP model R_1 contains two resource intervals whose supply functions are shown in Fig. 2a. Thus, the supply bound function of R_1 is the minimum between two supply functions, which is drawn as a dashed line in Fig. 2b.

3.3.2 Bounded Delay Resource Model

Mok et al. (2001) abstracts the SRP model with two parameters: *availability factor* and *delay* and calls it *Bounded Delay Resource* (BDR) model. For a given SRP model (Γ, P), the availability factor or resource supply rate to a component is defined by the resource amount per the resource period P. For instance, the availability factor of Example 1 is given by $\frac{3}{8}$. The delay parameter indicates the smallest time interval after which such availability factor is guaranteed by the SRP model.

Definition 2 Mok et al. (2001) The availability factor (rate) of a SRP model R is $\alpha(R) = \left(\sum_{i=1}^{n} (E_i - S_i)\right)/P$.

Definition 3 Mok et al. (2001) The partition delay Δ of a SRP model R is the smallest δ so that for any t_0 and t, $(t_0 \geq 0, t \geq 0)$, $(t - \delta) \times \alpha(R) \leq \mathrm{sf}_{t_0}(R, t) \leq (t + \delta) \times \alpha(R)$.

Definition 4 Mok et al. (2001) If h is the execution rate of a resource supply on which the SRP model R is implemented, then the normalized execution of the SRP model R is an allocation of resource time to R at a uniform, uninterrupted rate of $(\alpha(R) \times h)$.

The partition delay is a measure of the largest deviation of a partition in any time interval with respect to the resource supply being behind or ahead of its normalized execution. The bounded delay resource model is defined as follows.

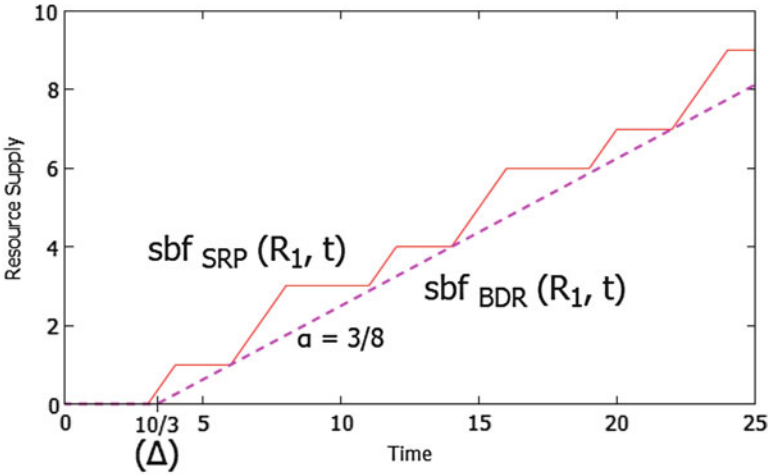
Definition 5 Mok et al. (2001) A bounded-delay resource (BDR) model is a tuple (α, Δ), where α is the rate of the partition and Δ is the partition delay.

Then, the supply bound function of the BDR model $R = (\alpha, \Delta)$ can be defined as follows:

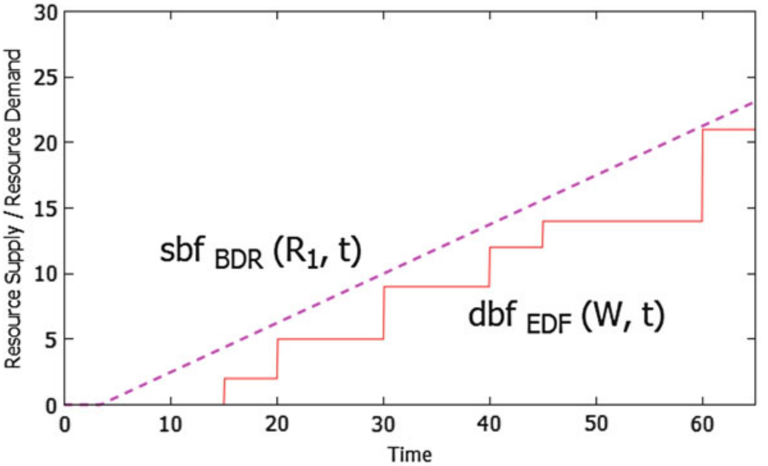
$$\mathrm{sbf}_{\mathrm{BDR}}(R, t) = \begin{cases} \alpha\,(t - \Delta) \text{ if } t \geq \Delta \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{6}$$

The BDR model of the SRP R_1 Example 1 is $\left(\frac{3}{8},\ \frac{10}{3}\right)$. As shown in Fig. 3a, the partition delay $\left(\Delta = \frac{10}{3}\right)$ can be obtained from the closest line of rate $\frac{3}{8}$ to the supply bound function $\mathrm{sbf}_{\mathrm{SRP}}(R_1, t)$. For example, the smallest delay occurs when the line passes the point (6, 1) so that Δ is equal to $\frac{10}{3}$ by solving $1 = \frac{3}{8}(6 - \Delta)$ from Eq. 6. The BDR model $\left(\frac{3}{8},\ \frac{10}{3}\right)$ implies that the component is supplied with a resource at the rate of $\frac{3}{8}$ after the delay time $\frac{10}{3}$.

Example 2 Consider $M = (W, \mathrm{EDF})$ with $W = \{(2,15,15),\ (3,20,20),\ (2,30,30)\}$ and EDF scheduling algorithm. The resource is supplied with the BDR model $R_1 = \left(\frac{3}{8},\ \frac{10}{3}\right)$ implemented by the SRP model in Fig. 2a. Figure 3b shows the supply bound function of the BDR model and the demand bound function of the workload under EDF. As shown in Fig. 3b, the demand of the workload is satisfied by (i.e., less than or equal to) the supply bound of the BDR model R_1, which means the schedulability of the component is guaranteed by the resource supply provided by the BDR model.



(a) BDR model $(\frac{3}{8}, \frac{10}{3})$ of SRP of Example 1.1



(b) sbf and dbf analysis of Example 1.2

Fig. 3 An example of the bounded delay model

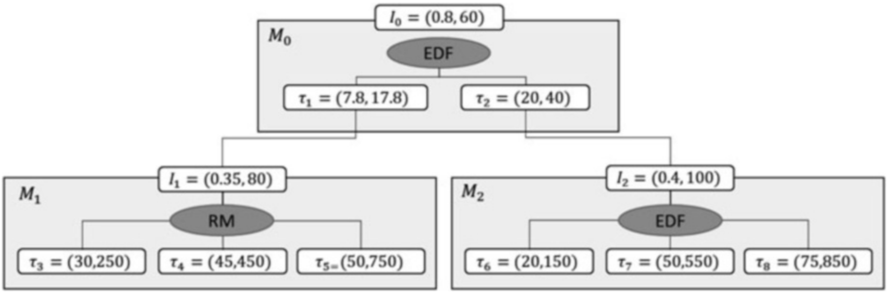


Fig. 4 The running example of a hierarchical scheduling system under BDR model

3.3.3 Hierarchical Partition Scheduling

A resource partition can also reside inside another partition, thereby forming a hierarchy of partitions. The schedulability of a hierarchical partition based on the BDR model is derived by the following theorem.

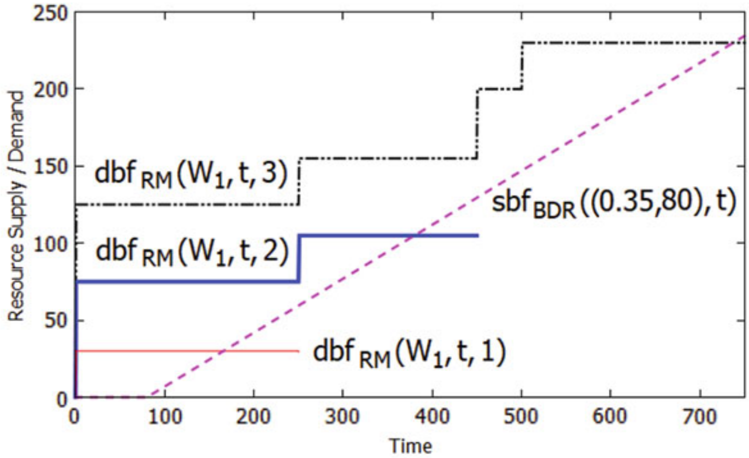
Theorem 1 Feng and Mok (2002) *In the* BDR *model, a resource partition group* $\{R_i(\alpha_i, \Delta_i)\}(1 < i \leq n)$ *is schedulable on a partition* $R\,(\alpha, \Delta)$ if $\sum_{i=1}^{n}\alpha_i \leq \alpha$ *and* $\Delta_i > \Delta$ *for all i.*

The above theorem states that for a group of partitions to be schedulable inside a parent partition, (i) the sum of the availability factors of the partitions in the group must be no greater than the availability factor of the parent partition and (ii) the bounded delay of each partition in the group must be greater than the partition delay of the parent.

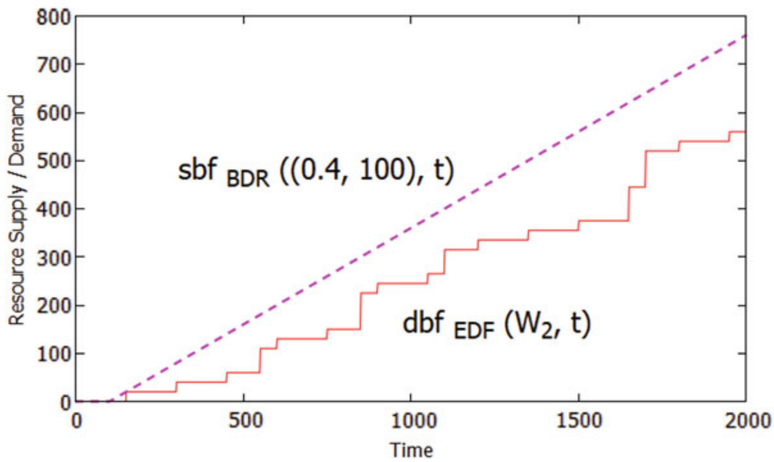
For example, Fig. 4 shows how the BDR model is used for hierarchical scheduling of the example in Fig. 1. As shown in Fig. 5a (Note that $\mathrm{dbf_{RM}}(W, t, i)$ for RM is drawn only from 0 to T_i in the figure), for each task τ_i in the component M_1, there exists $t \leq T_i$ such that $\mathrm{dbf_{RM}}(W, t, \;\; i) \leq \mathrm{sbf_{BDR}}((0.35{,}80), t)$, which implies the guarantee of the schedulability by the BDR model $R = (0.35{,}80)$ under RM algorithm. Similarly, Fig. 5b shows that the demand bound function is no greater than the supply bound function of the BDR model (0.4, 100) under EDF algorithm. Thus, we can use (0.35, 80) and (0.4, 100) as the BDR interface models of two components, denoted as I_1 and I_2, respectively.

In addition, Theorem 1 implies that these two BDR interfaces are also schedulable by the upper-layer component BDR interface model $I_0 = (0.8{,}60)$ because the sum of two availability factors is less than that of I_0, $(0.35 + 0.4 < 0.8)$ and the delay is less than each delay ($60 < 80$ and $60 < 100$).

Furthermore, an algorithm to transform schedulable resource partitions into schedulable tasks in an upper-layer component is provided in Feng and Mok (2002). First, Theorem 2 defines the transformation of a sub-partition on the normalized execution of a given resource partition, where the resource rate is shared among all sub-components. Then, the normalized partition is converted into a resource-supply task (It is called virtual task in Mok et al. (2001)) using half-half algorithm



(a) sbf and dbf of M_1 in Fig. 1.4



(b) sbf and dbf of M_2 in Fig. 1.4

Fig. 5 Supply and demand bound analysis of M_1 and M_2 in Fig. 4

(Mok et al. 2001) in which the resource supply task satisfies both partition delay Δ and availability factor α for a given BDR model (α, Δ).

Theorem 2 Feng and Mok (2002) *Given a set of* BDR *resource supplies* $\{R_i(\alpha_i, \Delta_i)\}(1 < i \leq n)$ *to be scheduled on a* BDR *resource supply of* $R(\alpha, \Delta)$*. Let* S_n *denote a scheduler of scheduling* $R_i^{'}(\alpha_i/\alpha, \Delta_i - \Delta)$ *on a dedicated resource with capacity of the same as the normalized execution of* R*. Also let* S_p *denote the virtual time* S_n *scheduler of scheduling* R_i *on* R*. Then* S_p *is valid if* S_n *is valid.*

Theorem 3 Mok et al. (2001) *(Half-half algorithm) A resource supply task* $\tau_k = \left(\alpha \times \frac{\Delta}{2(1-\alpha)}, \frac{\Delta}{2(1-\alpha)}\right)$ *can support the resource supply of the* BDR *model* (α, Δ).

Example 3 Let us consider the example in Fig. 4. Two schedulable subcomponents of BDR resource interfaces $I_1 = (0.35{,}80)$ and $I_2 = (0.4{,}100)$ for the given BDR component interface $I_0 = (0.8{,}60)$ are transformed to $I_1' = (0.35/0.8, 80 - 60) = (0.4375, 20)$ and $I_2' = (0.4/0.8, 100 - 60) = (0.5, 40)$, respectively, by Theorem 2. The half-half algorithm converts a component interface (α, Δ) into a virtual periodic task $(\alpha \times \Delta/2(1-\alpha), \Delta/2(1-\alpha))$. Thus, two transformed interfaces I_1' and I_2' are converted into resource supply tasks $\tau_1 = (7.8{,}17.8)$ and $\tau_2 = (20, 40)$, respectively. Because the utilization of two resource supply tasks is 0.938 on the normalized execution of the BDR model, which is less than one, they are schedulable by EDF on a partition $I_0 = (0.8{,}60)$. Figure 6 shows four resource supply and demand bound functions related to component M_2. The demand bound function of τ_2 is delayed to 60 and slower to the rate of 0.8 due to the bounded-delay model of $I_0 = (0.8{,}60)$, which is drawn as a thick line in Fig. 6. This demand bound function is no greater than the supply bound function of the BDR model $I_2 = (0.4{,}100)$ which satisfies the requirement of demand of M_2. Therefore, the resource supply task $\tau_2 = (20{,}40)$ in the upper-layer component M_0 meets the schedulability of M_2 in the BDR model.

3.4 Compositional Framework Based on Periodic Resource Model

The periodic task model (Liu and Layland 1973) has been commonly used to describe periodic real-time jobs. A periodic task model is defined by $\tau = (C, T)$, which assumes an implicit deadline, which is the same as its period. The

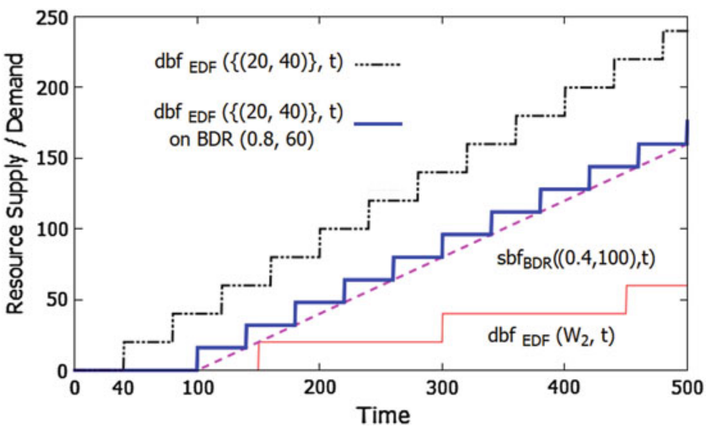


Fig. 6 Relation among the normalized executed resource supply and the demand of M_2

compositional framework based on periodic resource model adopts the periodic task model to capture the resource demand of a component and the resource supply to a component.

3.4.1 Periodic Resource Model

A periodic resource model (PRM) paradigm is represented by (Θ, Π) where Π is a period and Θ is a resource budget available every period Π. It is used to describe a resource supply to a component. It can also be used to describe the resource demand of tasks within a component. Here, it is called the interface of the component. For example, the periodic interface $I = (\Theta, \Pi)$ of a component represents that the collective resource demand of the tasks within the component is Θ time units every Π period. The difference between the periodic resource supply task and the periodic interface model is that the resource supply task is in terms of supply and the interface model is in terms of demand. The schedulability analysis of a component is based on supply and demand by checking if its resource supply is sufficient for its resource demand.

Figure 7 shows the worst-case resource supply of PRM, $R = (2, 5)$. Note that the resource supplies during the second and third periods are delayed by $\Psi = 6$ (i. e., $2 \times (\Pi - \Theta) = 2 \times (5 - 2)$) since Θ could be supplied at any time within the period Π. Based on this observation (Shin and Lee 2008), defines the supply bound function $\mathrm{sbf}_{\mathrm{PRM}}(R, t)$ to compute the minimum resource supply of PRM $R = (\Theta, \Pi)$ where $0 < \Theta \leq \Pi$ during a given time interval t as follows:

$$\mathrm{sbf}_{\mathrm{PRM}}(R, t) = \begin{cases} t - (k + 1) \cdot (\Pi - \Theta) & \text{if } t \in [(k+1) \cdot \Pi - 2 \cdot \Theta,\ (k+1) \cdot \Pi - \Theta]) \\ (k - 1) \cdot \Theta & \text{otherwise} \end{cases} \tag{7}$$

where $k = \max\left(\lceil (t - (\Pi - \Theta))/\Pi \rceil, 1\right)$, which computes the maximum number of periods within t. The first case computes the total resource supply where the last period overlapping t contribute to the resource supply. The second case computes

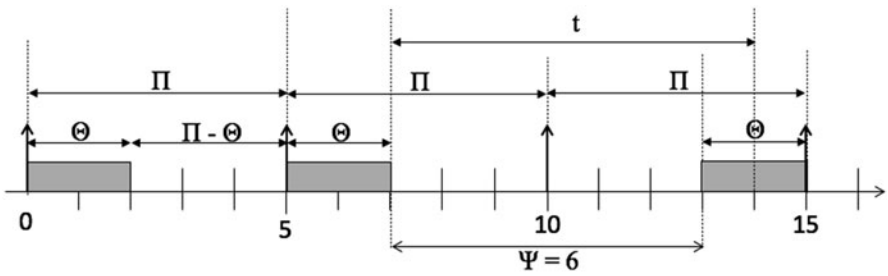


Fig. 7 A resource supply of PRM, $R = (\Theta, \Pi)$

the total resource supply where the last period does not contribute to the resource supply.

The supply bound function $\mathrm{sbf}_{\mathrm{PRM}}(R, t)$ is lower bounded by the linear function $\mathrm{lsbf}_{\mathrm{PRM}}(R, t)$ (Shin and Lee 2008), which is defined as follows:

$$\mathrm{lsbf}_{\mathrm{PRM}}(R, t) = \begin{cases} \frac{\Theta}{\Pi}(t - 2(\Pi - \Theta)) & \text{if } t \geq 2 \cdot (\Pi - \Theta) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

In the following section, we explain the schedulability conditions using $\mathrm{sbf}_{\mathrm{PRM}}$, $\mathrm{dbf}_{\mathrm{EDF}}$ and $\mathrm{dbf}_{\mathrm{RM}}$.

3.4.2 Schedulability Conditions for PRM

For a scheduling unit $S = (W, R, A)$, where resource supply is represented using PRM (Shin and Lee 2008), defines the schedulability conditions for an HSS scheduling unit under EDF and RM, respectively, as follows

Theorem 4 Shin and Lee (2008) *Scheduling unit $S = (W, R, EDF)$ is schedulable, where $W = \{\tau_i | \tau_i = (C_i, T_i)\}$ and $R = (\Theta, \Pi)$*, iff

$$\forall t \ s.t. 0 < t \leq \mathrm{LCM}_W, \quad \mathrm{dbf}_{\mathrm{EDF}}(W, t) \leq \mathrm{sbf}_{\mathrm{PRM}}(R, t) \tag{9}$$

where LCM_W is the least common multiple of T_i for all $\tau_i \in W$.

Theorem 5 Shin and Lee (2008) *Scheduling unit $S = (W, R, \mathrm{RM})$ is schedulable, where $W = \{\tau_i | \tau_i = (C_i, T_i)\}$ and $R = (\Theta, \Pi)$* iff

$$\forall \tau_i \in W, \quad \exists t_i \in [0, \quad T_i] \ \mathrm{dbf}_{\mathrm{RM}}(W, t_i, \quad i) \leq \mathrm{sbf}_{\mathrm{PRM}}(R, t_i) \tag{10}$$

The schedulability conditions for PRM above are used to check the component with respect to both a resource supply task R and I.

Example 4 The component M_1 in Fig. 8 executes a workload under RM. It can be shown schedulable by using Eq. 10 as follow: Fig. 9a shows that the supply graph crosses the demand graph of $\mathrm{dbf}_{\mathrm{RM}}(W, t, 3)$ at around 150 time units, which means that the resource supply task $\tau_1 = (36{,}100)$ supplies sufficient resource to complete τ_3 before the end of its period 250. τ_4 and τ_5 are also satisfied by with supply of $I = (36{,}100)$ before the end of their individual periods. The component M_1 is thus shown to be schedulable. Using Eq. 9, the schedulability of M_2 under EDF can be shown by checking the supply provided by $\tau_2 = (55{,}120)$ is greater than the collective demand of M_2's workload until LCM_W .

3.4.3 PRM Interface Generation: Periodic Capacity Bounds of PRM

In this section, we explain computation techniques for generating an interface for a component. Basically, the interface of components abstracts resource requirements

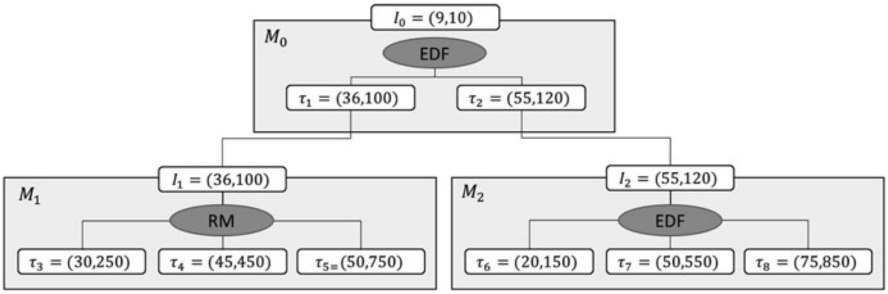


Fig. 8 The running example of a hierarchical scheduling system using the PRM model

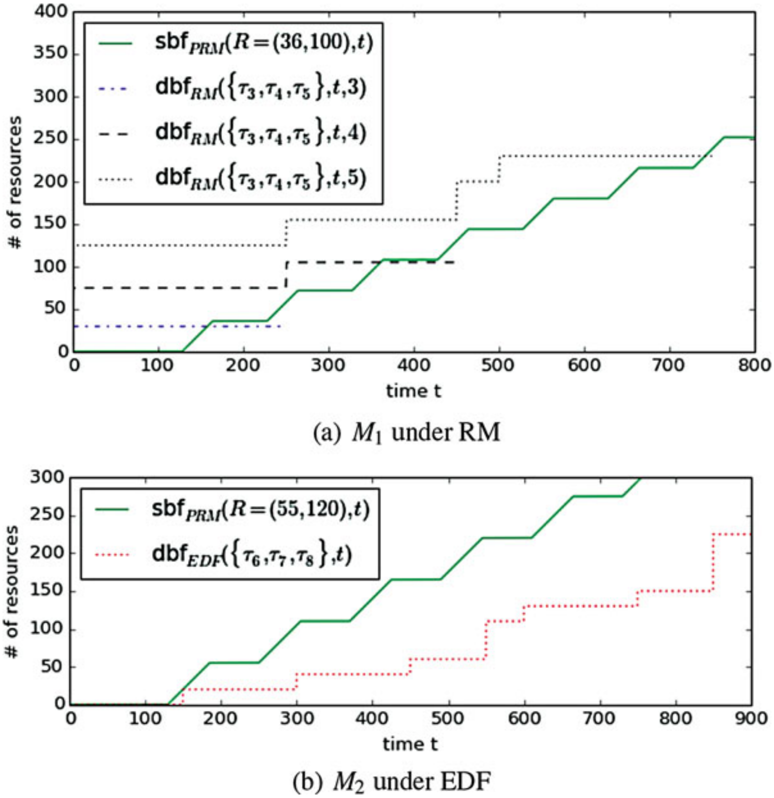


Fig. 9 Schedulability analysis by using $\mathrm{sbf}_{\mathrm{PRM}}$ and dbf_A

of individual tasks into a collective representation. In other words, the supply of the interface is also proved to satisfy the demand of the workload.

The supply of an interface $I = (\Theta, \Pi)$ is a resource supplier in terms of its workload, thus I can be found in such a way that for a given W, the a scheduling unit $S = (W, I, A)$ replacing R with I is found such that the minimum bandwidth of I is the minimum and W is schedulable by Theorems 4 or 5 according to A. We assume that Π of I is given with the workload W of a component. Then, the problem of computing the interface I for a W and Π is to find the minimum resource of Θ that satisfies the workload W.

The schedulability conditions of Theorems 4 and 5 depends, respectively, on LCM_W and the period of workload tasks. Hence (Shin and Lee 2003), introduces interface computation techniques that do not depend on the length of LCM_W or period of workload tasks. They introduced the notion of the periodic capacity bound of a workload $\mathrm{PCB}_W(\Pi, A)$, which bounds the maximum demand bound of W under A for a period Π. In other words, for a workload W, $\mathrm{PCB}_W(\Pi, A)$ computes a bandwidth of Θ/Π that satisfies W. The interface I is selected such that Θ/Π of $\mathrm{PCB}_W(\Pi, A)$ is the minimum. As a result, Θ and Π of the minimum Θ/Π from $\mathrm{PCB}_W(\Pi, A)$ is set to I.

For a period Π, a periodic capacity bound $\mathrm{PCB}_W(\Pi, A)$ is defined such that a scheduling unit $S = (W, I = (\Theta, \ \Pi), A)$ is schedulable if

$$\mathrm{PCB}_W\ (\Pi,\ A) \leq \frac{\Theta}{\Pi} \tag{11}$$

Using $\mathrm{PCB}_W(\Pi, A)$, it can be determined whether or not an interface $I = (\Theta, \Pi)$ can satisfy the resource demand of W under A. In the following, the interface of component under EDF and RM is derived from Eq. 11.

First, the optimal (minimal) periodic capacity bound $\mathrm{PCB}^*_W\ (\Pi, \mathrm{EDF})$ for a given Π is defined as follows:

$$\mathrm{PCB}^*_W\ (\Pi, \mathrm{EDF}) = \frac{\Theta^*}{\Pi} \tag{12}$$

where Θ^* is the smallest possible Θ such that

$$\forall 0 < \mathrm{t} \leq \mathrm{LCM}_W, \quad \mathrm{dbf}_{\mathrm{EDF}}\ (W, t) \leq \mathrm{sbf}_{\mathrm{PRM}}\ (I, t) \tag{13}$$

Then, a scheduling unit $S = (W, I = (\Theta, \ \Pi), \mathrm{EDF})$ is schedulable if and only if $\mathrm{PCB}^*_W\ (\Pi, \mathrm{EDF}) \leq \Theta/\Pi$.

Second, a function to compute a periodic capacity bound is derived as follows. For a workload W and a period Π of interface $I = (\Theta, \ \Pi)$, I satisfies W if

$$\mathrm{dbf}_{\mathrm{EDF}}\ (W, t) \leq \mathrm{lsbf}_{\mathrm{PRM}}\ (I, t) = \frac{\Pi}{\Theta}\ (t - 2 \cdot \Pi + 2 \cdot \Theta) \leq \mathrm{sbf}_{\mathrm{PRM}}\ (I, t) \tag{14}$$

Notice that $\mathrm{lsbf}_{\mathrm{PRM}}(I, t)$ is introduced to Eq. 13. From Eq. 14, the following inequality can be obtained:

$$\Theta \geq \frac{\sqrt{(t - 2 \cdot \Pi)^2 + 8 \cdot \Pi \cdot \mathrm{dbf}_{\mathrm{EDF}}\ (W, t)} - (t - 2 \cdot \Pi)}{4} \tag{15}$$

Let Θ^+ be the smallest possible Θ satisfying Eq. 15. Then, $S=(W, I=(\Theta^+, \Pi)$, EDF) is schedulable.

For a given workload W under EDF and a period Π, $I = (\Theta^+, \Pi)$ can be obtained using $\mathrm{PCB}_W(\Pi, \mathrm{EDF})$ as follows:

Theorem 6 Shin and Lee (2003) *For a given periodic workload W under* EDF *a periodic capacity bound* $\mathrm{PCB}_W(\Pi, \mathrm{EDF})$ *computes the utilization bound of interface* $I = (\Theta, \Pi)$ *that schedules the workload W. It is defined by*

$$\mathrm{PCB}_W(\Pi, \mathrm{EDF}) = \frac{\Theta^+}{\Pi}, \text{ where}$$

$$\Theta^+ = \max_{0<t\leq \mathrm{LCM}_W}\left(\frac{\sqrt{(t-2\cdot\Pi)^2 + 8\cdot\Pi\cdot \mathrm{dbf}_{\mathrm{EDF}}(W,t)} - (t-2\cdot\Pi)}{4}\right) \tag{16}$$

Again, Θ^+ is a lower-bound of possible Θ satisfying Eq. 15. $\mathrm{PCB}_W(\Pi, \mathrm{EDF}) = \Theta^+/\Pi$ is the smallest resource bound that satisfies W. Consequently, $I = (\Theta^+, \Pi)$ is the interface of the minimum bandwidth that satisfies W under EDF.

Similarly, $\mathrm{PCB}_W(\Pi, \mathrm{RM}) = \Theta^+/\Pi$ is the smallest bandwidth that satisfies W under RM and computed as follows:

Theorem 7 Shin and Lee (2003) *For a given periodic workload W under* RM, *a periodic capacity bound* $\mathrm{PCB}_W(\Pi, \mathrm{RM})$ *computes the utilization bound of interface* $I = (\Pi, \Theta)$ *that schedules the workload W. It is defined by*

$$\mathrm{PCB}_W(\Pi, \mathrm{RM}) = \frac{\Theta^+}{\Pi}, \text{ where}$$

$$\Theta^+ = \max_{\forall T_i \in W}\left(\frac{-\left(T_i - 2\cdot\Pi + \sqrt{(T_i - 2\cdot\Pi)^2 + 8\cdot\Pi\cdot J_i}\right)}{4}\right), \text{ where}$$

$$J_i = C_i + \sum_{\tau_k \in \mathrm{HP}_{W(i)}} \left\lceil \frac{T_i}{T_k} \right\rceil \cdot C_k$$

As mentioned in Sect. 3.2, the resource supply R in the PRM compositional framework is obtained just by setting $R = I$[12, 13].

3.4.4 Workload Utilization Bounds of PRM Under EDF and RM

To deal the LCM_W complexity in computing the demand and supply bound functions of PRM, PRM (Shin and Lee 2008) proposes ways to use the utilization bound of a workload to check schedulability in constant time (The complement concept of the workload utilization bound is an interface or resource model utilization bound which denote the maximum amount of resources available by an interface or a resource supply task. This chapter explains only the workload utilization bound. The interface utilization bound is referred to Sect. 6 of Shin and Lee (2008)).

Definition 6 (Utilization Bound) For a given scheduling unit $S = (W, R, A)$, we define the *utilization bound*, $\cup B$, as a value in [0,1] such that the scheduling unit is schedulable if the utilization $\cup_W$ of the workload W is no greater than the utilization bound $\cup B$.

The schedulability test for a scheduling unit can be performed by using the following inequality:

$$\cup_W \leq \cup B_{\mathrm{PRM},A}\,(W,\,R) \tag{18}$$

where $\cup_W$ is the utilization of a workload W and $\cup B_{\mathrm{PRM},\,A}(W, R)$ is a minimum resource utilization available by the resource supply that depends on the PRM and scheduling algorithm A for W.

Let $T_{\min}$ be the smallest period of tasks in workload W. A periodic multiple relationship denoted by $K_A(T_{\min}, R)$ indicates how many times of a periodic resource model $R = (\Theta,\ \Pi)$ can provide W with the whole periodic allocation of Θ during $T_{\min}$ for A equals RM and EDF, respectively. For RM,

$$K_{\mathrm{RM}}\,(T_{\min},\,R) = \max\,\{k\,|\text{k is an integer },\,((k+1)\cdot\Pi) - \Theta\, < T_{\min}\} \tag{19}$$

For EDF,

$$K_{\mathrm{EDF}}\,(T_{\min},\,R) = \max\left\{k\,|\text{k is an integer },\,((k+1)\cdot\Pi) - \Theta - \frac{k\cdot\Theta}{k+2} < T_{\min}\right\} \tag{20}$$

Theorem 8 (Utilization Bound Under EDF)

For a given scheduling unit $S = (W, R,\ \mathrm{EDF})$, *where* $W = \{\tau_i | \tau_i = (C_i,\ T_i)\}$, $R = (\Theta, \Pi)$, *and the utilization bound* $\cup B_{\mathrm{PRM,\,EDF}}(W, R)$ *is defined by*

$$\cup B_{\mathrm{PRM,EDF}}\,(W,\,R) = \frac{k\cdot U_{\mathrm{PRM}}}{k + 2\cdot(1 - U_{\mathrm{PRM}})} \tag{21}$$

where $k = K_{\mathrm{EDF}}(T_{\min}, \mathrm{PRM}(\Pi, \Theta))$, $\cup_{\mathrm{PRM}} = \frac{\Theta}{\Pi}$, *and* $T_{\min}$ *is the minimum period of tasks in* W.

Example 5 Consider a scheduling unit $S = (W, R, A)$, where $R = (4, 10)$ and $A = \mathrm{EDF}$. The resource capacity $\cup_R$ of R is 0.4. Suppose that the smallest task period $T_{\min}$ of the workload W is greater than 50, then k is 4 by the definition of $K_{\mathrm{EDP}}(T_{\min}, R)$. According to Theorem 8, EDF utilization bound of $\cup B_{\mathrm{PRM,\,EDF}}(W, R)$ is 0.32. If $\cup_W \leq 0.32$, the scheduling unit S is schedulable.

Theorem 9 (Utilization Bound Under RM) *For a given scheduling unit* $S = (W, R, \mathrm{RM})$, *where* $W = \{\tau_i | \tau_i = (C_i,\ \ T_i)\}$, $R = (\Theta, \Pi)$, *and* $T_i \geq (2 \cdot \Pi) - \Theta$, *and* $1 \leq i \leq n$, *the utilization bound* $\cup B_{\mathrm{PRM,\,RM}}(W, R)$ *is defined by*

$$\cup B_{\mathrm{PRM,RM}}\,(W,\,R) = \cup_{\mathrm{PRM}} \cdot n \cdot \left[\left(\frac{2 \cdot k + 2 \cdot (1 - \cup_{\mathrm{PRM}})}{k + 2 \cdot (1 - \cup_{\mathrm{PRM}})} \right)^{1/n} - 1 \right] \tag{22}$$

where $k{=}K_{\mathrm{RM}}\,(T_{\min},\, R)\,,\, \cup_{\mathrm{PRM}}{=}\frac{\Theta}{\Pi}$, *and* $T_{\min}$ *is the minimum period of tasks in* W.

3.5 Compositional Framework Based on Explicit Deadline Periodic Resource Model

The Explicit Deadline Periodic (EDP) resource model is a resource modeling paradigm that extends the periodic resource model with explicit deadline. Since an EDP resource model is restricted by the deadline potentially shorter than the period, the interval where no resource is supplied between consecutive resource periods is shorter than that of PRM, as shown in in Figs. 10 and 11. Consequently, an EDP resource model can be more effective in supplying resource to a child component than PRM. In EDP setting, the underlying scheduling algorithms considered in the literature are EDF, RM, and Deadline Monotonic (DM). In DM, the task of a shorter deadline has higher priority than the one of a longer deadline.

An EDP resource model is represented by (Θ, Π, Δ), which means the Θ time units of resource is guaranteed to be supplied by the deadline Δ every Π time units. From Easwaran et al. (2007), we define the supply bound function $\mathrm{sbf}_{\mathrm{EDP}}$ for $R = (\Theta,\ \ \Pi, \Delta)$ and any time interval t as follows:

$$\mathrm{sbf}_{\mathrm{EDP}}\,(R, t) = \begin{cases} y \cdot \Theta + \max\{0, t - x - y \cdot \Pi\}\,, & t \geq \Delta - \Theta \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

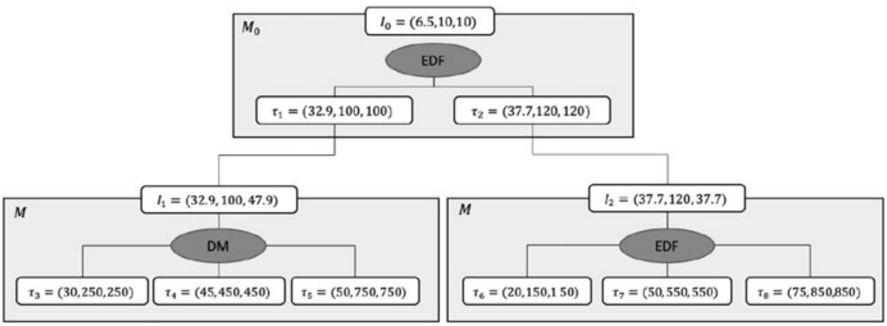


Fig. 10 The running example of a hierarchical scheduling system under EDP resource model

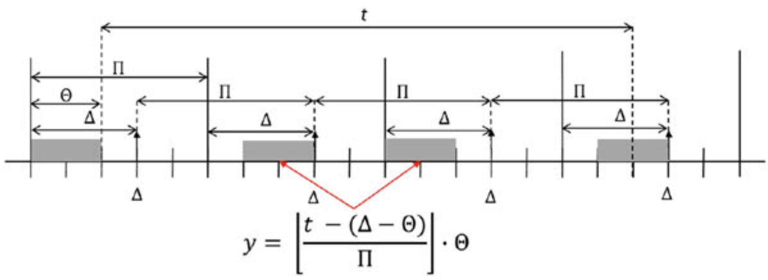


Fig. 11 A resource supply of a EDP resource model

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

Figure 10 shows an HSS under EDP resource model, extending the example of Fig. 8 with deadlines. Note that the interfaces I_1 and I_2 have different deadlines from their associated resources models, τ_1 and τ_2, in the parent component M_0. Sect. 3.5.4 will discuss the reason in detail.

3.5.1 Schedulability Conditions for EDP

The schedulability conditions for EDP components are different from those of PRM components as follows: First, the supply bound function of EDP component uses (Θ, Π, Δ) for R in Eq. 23. Second, the schedulability conditions of EDP component use $\tau = (C_i,\ T_i, D_i)$ for workload tasks. Third, the searching space to check $\mathrm{dbf}_A(W,t) \le \mathrm{sbf}_{\mathrm{EDP}}(R,t)$ is bound by D_i of tasks.

The schedulability conditions for an HSS scheduling unit under EDF and RM are defined as follows:

Theorem 10 Easwaran et al. (2007) *Scheduling unit* $S = (W, R, \mathrm{EDF})$ *is schedulable, where* $W = \{\tau_i | \tau_i = (C_i, T_i, D_i),\ \ 0 < i \le n\}$, *and* $R = (\Theta, \Pi, \Delta)$, iff

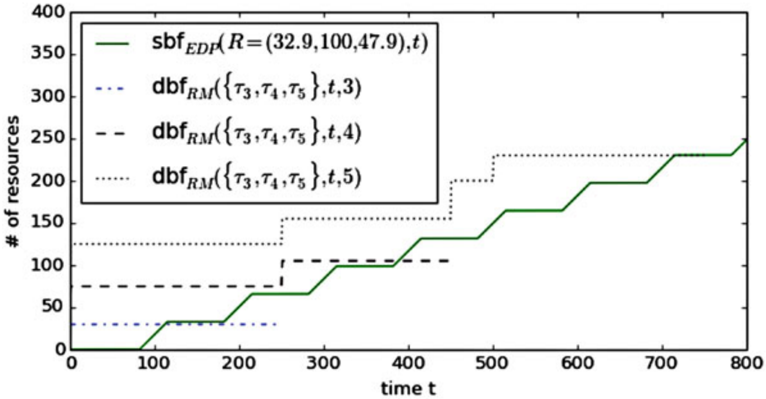
$$\forall t\ s.t. 0 < t \le \mathrm{LCM}_W + \max_{i=1}^{n} D_i,\ \ \mathrm{dbf}_{\mathrm{EDF}}(W,t) \le \mathrm{sbf}_{\mathrm{EDP}}(R,t) \tag{24}$$

where LCM_W *is the least common multiple of* T_i *for all* $\tau_i \in W$.

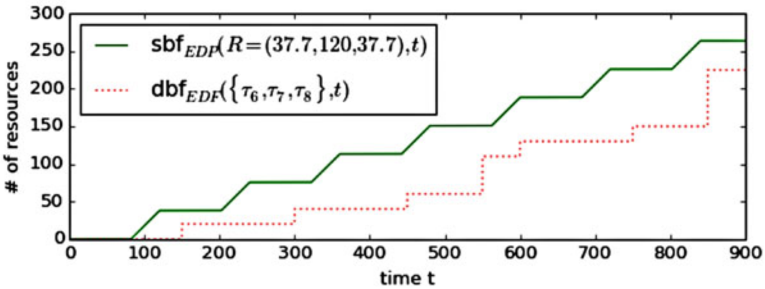
Theorem 11 Easwaran et al. (2007) *Scheduling unit* $S = (W, R, \mathrm{DM})$ *is schedulable, where* $W = \{\tau_i | \tau_i = (C_i,\ \ T_i, D_i),\ \ 0 < i \le n\}$, *and* $R = (\Theta, \Pi, \Delta)$, iff

$$\forall T_i \in W,\ \ \exists t_i \in [0, D_i]\ \ \mathrm{dbf}_{\mathrm{RM}}(W, t_i, i) \le \mathrm{sbf}_{\mathrm{EDP}}(R, t_i) \tag{25}$$

Figure 12 shows the demands of M_1 and M_2 in Fig. 10 represented by $\mathrm{dbf}_{\mathrm{RM}}$ and $\mathrm{dbf}_{\mathrm{EDF}}$ and the supplies represented by $\mathrm{sbf}_{\mathrm{EDP}}(R,t)$. Figure 12a shows that the supply of R = (32.9,100,47.9) provides sufficient resources to individual resource requirement of the tasks τ_3, τ_4, and, τ_5 under DM before their individual



(a) M_1 under DM



(b) M_2 under EDF

Fig. 12 Schedulability analysis by using sbf_{PRM} and dbf_A

deadlines. Hence, the component M_1 is shown to be schedulable by the supply of $R = (32.9,100,47.9)$. Figure 12b shows that the supply of $R = (37.7,120,37.7)$ provides sufficient resources for tasks, τ_6, τ_7, and τ_8, until 900 time units. It can also be shown that M_2 is schedulable by checking if the supply is sufficient for those tasks until LCM_W of the tasks, 28,050 time units.

3.5.2 Optimality Properties in EDP

Compared to PRM, the interface of EDP resource model can require less resources to schedule a component by restricting the maximum starvation of resource supply with its deadline. From Easwaran et al. (2007), we have two notions of optimality, which are used in EDP interface generation.

Definition 7 (Bandwidth Optimal) Easwaran et al. (2007) For a given component M and a period Π, an EDP interface $I = (\Theta, \ \Pi, \Delta)$ that schedules M is *bandwidth optimal* iff the bandwidth of the interface is the minimum of all EDP interfaces that can schedule M.

Definition 8 (Bandwidth Deadline Optimal Interface) Easwaran et al. (2007) An EDP $I = (\Theta, \Pi, \Delta)$ is a bandwidth-deadline optimal interface that schedules M iff (1) the I is bandwidth optimal for M, and (2) $\Delta > \Delta^{'}$ for all $I^{'} = (\Theta, \Pi, \Delta^{'})$ such that I'_{EDP} is bandwidth optimal for M.

Definition 7 defines the minimal bandwidth of EDP interface that schedules M without accounting for the deadline. Definition 8 defines the minimal bandwidth of EDP resource model that schedules M accounting for the deadline, so that the EDP resource model has the minimum bandwidth and the largest deadline among interfaces that schedule M with the same Θ and Π. Using these two properties, an EDP interface can be generated such that the bandwidth of the interface is minimal and the deadline is the longest out of possible EDP interfaces.

3.5.3 EDP Interface Generation

As described in Sects. 3.3.3 and 3.4.3, for BDR and PRM, the interface of a component is generated by first computing a sufficient resource supply for the component's workload and then use the resource supply parameters as the interface. In contrast, for EDP the interface of a component is first generated by computing the dbf of the component. Then, the resource supply task for the component is generated from the interface and using the properties of EDP.

For a component M and interface period Π, the bandwidth-deadline optimal interface of M can be computed by the following two steps: (i) find a bandwidth optimal interface that schedules M with the assumption of $\Theta = \Delta$ and (ii) find a bandwidth-deadline optimal interface by gradually increasing the deadline as long as the schedulability is preserved. At each step, ensure the schedulability using the schedulability conditions Eqs. 24 and 25.

More specifically, the interface of a component under EDF (Easwaran et al. 2007) is generated as follows:

Step 1 Set Δ to Θ. Check Eq. 24 over all interval length (LCM_W) and search for the minimum $\Theta(=\Theta^m)$ over the interval length that always schedules M i.e., find a bandwidth-optimal interface for M.

Step 2 Set Θ to Θ^m. Check Eq. 24 over all interval length (LCM_W) and search for the maximum $\Delta(=\Delta^m)$ over the interval length. Then, $I^m = (\Theta, \ \Pi^m, \Delta^m)$ is the bandwidth-deadline optimal interface for M under EDF.

The interface of a component under DM (Easwaran et al. 2007) is generated as follows:

Step 1 Set Δ to Θ. Check Eq. 25 for each task up to its deadline and search for the minimum $\Theta \left(= \Theta_i^m\right)$ over the interval length. Let $\Theta^m = \max_i \ \Theta_i^m$

Step 2 Set Θ to Θ^m Check Eq. 25 for each task up to its deadline and search for the maximum $\Delta \left(= \Delta_i^m\right)$ over the interval length. Let $\Delta_m = \min_i \ \Delta_i^m$ Then, $I^m = (\Theta^m, \ \Pi, \Delta^m)$ is the bandwidth-deadline optimal interface for M under DM.

The difference of the interface generation between EDF and DM is that the schedulability condition for the component under DM requires to check the schedulability of each task up to its deadline.

The two interfaces I_1 and I_2 in Fig. 10 are computed using these algorithms and thus optimal. A more efficient computation algorithm for generating EDP interfaces can be found in Easwaran et al. (2007).

3.5.4 Exact Transformation of Interfaces in EDP

This section discusses how to generate a resource model that can supply resource needed from the demand interface of a component. Suppose that there is the workload of a component schedulable by $I = (5,2,5)$. The parameter values of the interface I is directly used for the associated resource supply task $\tau = (5,2,5)$. Then, τ demands resources to its upper-level component, as shown in Fig. 13, and a supply of τ's parent component satisfying τ should provide more resources than I should actually provide. On the other hand, $\tau' = (5,2,8)$ requires the exact amount of resources from its parent component that satisfies I supply amount of resources from the parent component that satisfies $I = (5,2,5)$. Based on this observation, the exact transformation of an interface model into an interface task under EDF and DM serviced by EDP interfaces are defined in Easwaran et al. (2007).

Let $\mathcal{T}_A$ be a function that transforms an EDP interface into an interface task τ under scheduling algorithm A. Then, the transformation function $\mathcal{T}_{\mathrm{EDF}}$ for a given EDP interface is defined as follows:

Definition 9 Easwaran et al. (2007) For a given EDP interface $I = (\Theta, \Pi, \Delta)$, the transformation function is defined by $\mathcal{T}_{\mathrm{EDF}}(I) = (\Theta, \Pi, \Pi + \Delta - \Theta)$.

The interface task generated by $\mathcal{T}_{\mathrm{EDF}}$ is demand-supply optimal for I in that it requires the least resources among all interface tasks that satisfy I (Easwaran et al. 2007).

The transformation function $\mathcal{T}_{\mathrm{DM}}$ returns a demand-supply optimal interface task τ under DM for an EDP interface I. Let $I' = (\Theta', \Pi', \Delta')$ be the interface of a component that encloses the resource supply task τ that would be built from EDP interface I.

Definition 10 Easwaran et al. (2007) For an EDP interface $I = (\Theta, \Pi, \Delta)$ and a period Π', the bandwidth-optimal resource supply $R^* = (\Theta^*, \Pi', \Theta^*)$ under DM

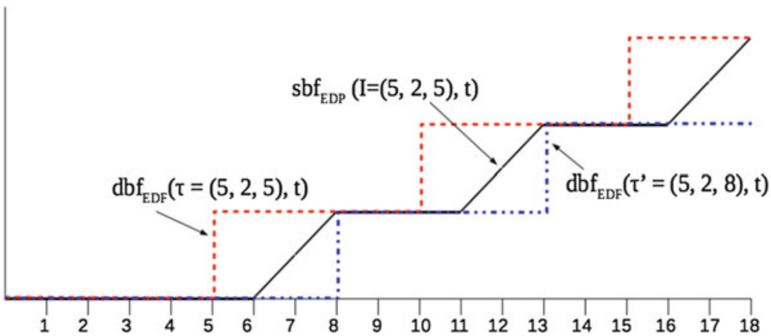


Fig. 13 Resource supplies of $\mathrm{sbf}_{\mathrm{EDP}(5,2,5)}$ and its resource supply tasks

is such that

$$\Theta^* = \begin{cases} \Theta + \frac{k \cdot \Theta}{\Delta} & \Pi' = \Pi + k, k \geq 0 \\ \frac{\Theta}{k} & \Pi' = \frac{\Pi}{k}, k \geq 2 \\ \Theta_C & \Pi' = \frac{\Pi}{k} + \beta, k \geq 2 \end{cases} \tag{26}$$

where $\beta \in \left(0, \frac{\Pi}{k \cdot (k=1)}\right)$ and k is an integer. If there exits $n \in \left(\frac{\Delta - \Theta}{k \cdot \beta}, \frac{\Delta - \Theta}{k \cdot \beta} + \Pi' + 1\right)$ such that $n \cdot k \cdot \beta - (\Delta - \Theta) = l \cdot \Pi^{'} + \Upsilon$ and n is the smallest integer satisfying $\Upsilon \geq \frac{n \cdot \Theta + \Upsilon}{n \cdot k - l}$, then $\Theta_C = \frac{n \cdot \Theta}{n \cdot k - (l+1)}$. Otherwise, $\Theta_C = \frac{\Theta}{k} + \frac{\beta \cdot \Theta}{\Pi}$.

Based on the bandwidth optimal resource supply $R^* = (\Theta^*, \Pi^{'}, \Theta^*)$, the transformation function $\mathcal{T}_{\mathrm{DM}}$ returns an interface task under DM is defined as follows:

Definition 11 Easwaran et al. (2007) For a given EDP interface $I = (\Theta, \Pi, \Delta)$ and period Π', the transformation function $\mathcal{T}_{\mathrm{DM}}\left(I, \Pi'\right) = \left(\Theta^*, \Pi', \Pi'\right)$.

Figure 10 shows the hierarchical scheduling system that updates the running example of Fig. 8 with the EDP resource model. Notice that the interface I_1 and I_2 have deadlines different from their interface tasks' deadlines.

4 Real-Time Calculus for HSS

There are several system level performance analysis techniques in literature, which are simulation based approaches, formal approaches, or a hybrid of the two approaches. Real-Time Calculus (RTC) (Chakraborty et al. 2003; Thiele et al. 2000) is a deterministic formal performance analysis technique, which provides worst-case bounds for the system parameters analyzed. Real-Time Calculus has its theoretical roots in Network Calculus (Le Boudec and Thiran 2001), which is used to analyze timing properties of application flows. The analysis based on Network Calculus utilizes *cumulative function* of event arrivals and number of resources provided on a node to determine the delays incurred in processing an event stream. In contrast, RTC uses *interval bound functions* to capture event arrivals as well as available resources. These interval bound functions for event arrivals and available resources are termed *arrival curves* and *service curves*, respectively. The cumulative function and interval bound function for a given event stream are illustrated in Fig. 14. The event stream arrival is shown in Fig. 14a, where the x-axis is the time t. In Fig. 14b, the cumulative function $A(t)$ of the event arrival is shown for any time t, which shows how many events arrived from zero to time t. The interval bound function shown in Fig. 14c gives the upper ($\alpha^u(\Delta)$) and lower bound ($\alpha^l(\Delta)$) on the number of event arrivals in any time interval Δ. For each Δ, a sliding window of width Δ is traversed over the entire event stream to find the maximum and minimum number of events that arrived in any time interval Δ.

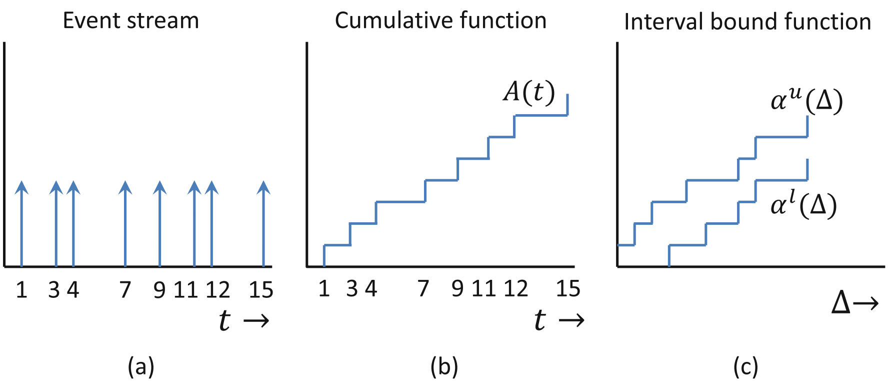


Fig. 14 Illustration of cumulative and interval bound functions

4.1 Workload and Service Model

The workload model captures the event arrivals that need to be processed on a computation node. This workload can be captured in terms of the number of events that arrive in a time interval or the number of processor cycles required to process the events in a time interval. In RTC, the workload is captured in terms of the *arrival curve*, which is defined below.

Definition 12 (Arrival Curve) Let $A(t)$ denote the number of events that arrive in time interval $(0, t)$. Then, the incoming workload is bounded by the arrival curve $\alpha = [\alpha^u, \ \alpha^l]$ iff for all arrival patterns $A(t)$:

$$\alpha^l \left(\Delta\right) \leq A \left(t + \Delta\right) - A(t) \leq \alpha^u \left(\Delta\right) \tag{27}$$

for all $\Delta \geq 0$. In other words, $\alpha^u(\Delta)$ and $\alpha^l(\Delta)$ give the maximum and minimum number of events that can arrive over any interval of length Δ across the entire stream.

The above definition of arrival curve is more specifically termed the *event-based arrival curve*. The arrival curve can also be characterized in terms of the processor time required to execute the task events in a time interval. This is referred to as the *resource-based arrival curve*. If the worst-case execution time (WCET) and best-case execution time (BCET) of processing an event are $e_{\max}$ and $e_{\min}$, respectively, then the resource-based arrival curve $\left(\overline{\alpha}^u, \overline{\alpha}^l\right)$ can be derived from the event-based arrival curve using the transformations: $\overline{\alpha}^u = \alpha^u \times e_{\max}$ and $\overline{\alpha}^l = \alpha^l \times e_{\min}$.

The service model captures the number of units of a resource provided in a given time interval Δ using *service curve*. This *resource-based service curve* is defined below.

Definition 13 (Service Curve) Let $C(t)$ denote the number of processing or communication units available from a resource in time interval (0, t). Then, $\overline{\beta} = \left[\overline{\beta}^u, \overline{\beta}^l\right]$ is a service curve of the processor iff for all service patterns $C(t)$:

$$\overline{\beta}^l(\Delta) \leq C(t + \Delta) - C(t) \leq \overline{\beta}^u(\Delta) \tag{28}$$

for all $\Delta \geq 0$. In other words, $\overline{\beta}^u(\Delta)$ and $\overline{\beta}^l(\Delta)$ denote the upper and lower bounds on the processing/communication units available from a resource over any interval of time Δ across the entire event stream.

The service curve can also be characterized in terms of the number of task events processed in a time interval, which is referred to as the *event-based service curve*. The event-based service curve can be derived from the resource-based service curve using the transformations: $\beta^u = \overline{\beta}^u / e_{\min}$ and $\beta^l = \overline{\beta}^l / e_{\max}$.

4.2 Schedulability Analysis

Given a set of tasks $\{\tau_i\}$ executed on a uniprocessor, the minimum service required to ensure schedulability of jobs of each task depends on the scheduling algorithm employed. For a periodic task $\tau_i = (C_i, \ T_i, D_i)$ (or sporadic task), the upper bound on arrival curve for each task can be obtained using the task's real time parameters, such as period (or minimum inter arrival distance) and WCET as follows

$$\overline{\alpha}_i^u(\Delta) = \left\lceil \frac{\Delta}{T_i} \right\rceil \times C_i \tag{29}$$

EDF schedulability : In order for the task to be schedulable, i.e., for execution of each job of the task to finish before the deadline D_i, the service required in case of EDF scheduling is given by $\overline{\beta}_i^l(\Delta) \geq \overline{\alpha}_i^u(\Delta - D_i)$. The following theorem presents the lower bound of service required for the schedulability of a set of tasks under EDF (Stoimenov et al. 2009).

Theorem 12

Under EDF *scheduling, the lower bound of service required for schedulability of tasks in the set τ_i with deadline D_i is given by* $\overline{\beta}^l(\Delta) = \sum_{\forall i} \overline{\alpha}_i^u(\Delta - D_i)$.

Fixed Priority Schedulability: Under FP scheduling with pre-emption, within any interval Δ, the lower priority tasks get the remaining service that is left over after servicing the tasks of higher priority. The lower bound of service required by task τ_i alone is given by $\overline{\beta}_i^{l*}(\Delta) = \overline{\alpha}_i^u(\Delta - D_i)$. The following theorem presents the lower bound of service required by task τ_i in the presence of a lower priority task τ_{i+1} (Chakraborty et al. 2006).

Theorem 13 *If task τ_i has higher priority than task τ_{i+1}, the lower bound of service curve required by τ_i is given by*

$$\forall\Delta \geq 0 : \overline{\beta}_i^l(\Delta) = \max\left\{\mathrm{Serv}_{\min}\left(\overline{\beta}_{i+1}^l(\Delta), \overline{\alpha}_i^u(\Delta)\right), \overline{\beta}_i^{l*}(\Delta)\right\} \tag{30}$$

where $\mathrm{Serv}_{\min}\left(\overline{\beta}_{i+1}^l(\Delta), \overline{\alpha}_i^u(\Delta)\right) = \overline{\beta}_{i+1}^l(\Delta - \lambda) + \overline{\alpha}_i^u(\Delta - \lambda)$ *and* $\lambda = \sup\left\{\varepsilon \left| \overline{\beta}_{i+1}^l(\Delta - \varepsilon) = \overline{\beta}_{i+1}^l(\Delta)\right.\right\}$.

The function $\mathrm{Serv}_{\min}\left(\overline{\beta}_{i+1}^l(\Delta), \overline{\alpha}_i^u(\Delta)\right)$ is the smallest service curve required by a task with arrival curve $\overline{\alpha}_i^u(\Delta)$ such that the remaining service curve after processing the task is at least $\overline{\beta}_{i+1}^l(\Delta)$. Let us assume a set of m tasks τ_i with the priority decreasing with increasing value of i. Then, the lower bound of service required for schedulability of task τ_m is given by $\overline{\beta}_m^l = \overline{\alpha}_m^u(\Delta - D_m)$. The lower bound of service for task τ_{m-1} $\left(\overline{\beta}_{m-1}^l(\Delta)\right)$ can be obtained using Theorem 13. This procedure can be repeated and the schedulability of the task set is ensured if the total service provided is greater than or equal to $\overline{\beta}_1^l(\Delta)$, which is the service requirement of task τ_1. The idea behind the computation of lower bound on service for the task set under fixed priority preemptive scheduling is that each task must have just enough service available in every time interval Δ to ensure its schedulability. This means that the lower bound of service available to a task must be such that the task is schedulable and the remaining service left after processing the task is just enough to ensure schedulability of all the lower priority tasks. Therefore, the lower bound of service available to the highest priority task must be such that the schedulability of all lower priority tasks is ensured along with its schedulability.

4.3 Compositional Analysis for Hierarchical Scheduling Systems

In this section, we present the steps of how compositional analysis is performed for hierarchical scheduling systems using RTC. Principally, the interface-based technique used by RTC is similar to the one used by compositional approaches using PRM and EDP. However, RTC uses the lower bound of service curve $\left(\overline{\beta}^l\right)$ in the interface of a component, which is different from the interface parameters used by compositional approaches using PRM and EDP. Once the interface parameter $\overline{\beta}^l$ is obtained for each component, the schedulability of the component can be verified by ensuring that the next higher level component in the hierarchy satisfies the required interface parameter. We explain this procedure using the hierarchical scheduling system shown in Fig. 8.

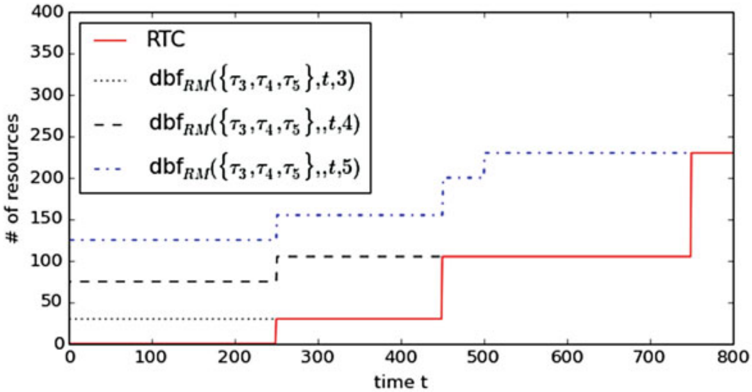
In Fig. 8, there are two levels in the hierarchy. Levell includes the leaf nodes with components M_1 and M_2. Component M_1 encapsulates tasks $\tau_{3,}$ τ_4 and τ_5 scheduled using RM, whereas component M_2 includes tasks $\tau_{6,}$ τ_7 and τ_8 scheduled using EDF. For component M_2, we can derive the lower bound of service required considering the three tasks that it includes using Theorem 12. Let us denote the resultant lower bound of service for component M_2 as $\overline{\beta}^l_{M_2}$. Similarly, the lower bound of service for component M_1 $\left(\overline{\beta}^l_{M_1}\right)$ can be obtained using Theorem 13. The interface parameters of M_1 and M_2 are $\overline{\beta}^l_{M_1}$ and $\overline{\beta}^l_{M_2}$ respectively. The lower bound of service provided by component M_0 at level 0 (or root level) must ensure that the interface requirement of components M_1 and M_2 are satisfied. The interface parameters $\overline{\beta}^l_{M_1}$ and $\overline{\beta}^l_{M_2}$ can be considered as the workload for component M_0.

In order to compute the interface of component M_0 given the period for the interfaces of components M_1 and M_2, the arrival curves $\overline{\alpha}^u_{M_1}$ and $\overline{\alpha}^u_{M_2}$ are calculated by finding the least execution times for the interface periods such that $\overline{\alpha}^u_{M_1}$ in component M_0 (given by Eq. 29) is greater than or equal to the interface parameter $\overline{\beta}^l_{M_1}$ and $\overline{\alpha}^u_{M_2}$ is greater than or equal to the interface parameter $\overline{\beta}^l_{M_2}$. This ensures that the arrival curves corresponding to components M_1 and M_2 upper bound the interface parameters $\overline{\beta}^l_{M_1}$ and $\overline{\beta}^l_{M_1}$, respectively. With the arrival curves of the child components and their interface periods, the interface parameter of M_0 $\left(\overline{\beta}^l_{M_0}\right)$ can be computed using Theorem 12.

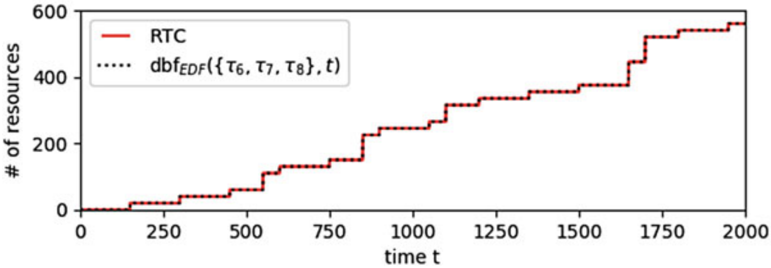
5 Comparison of Compositional and RTC Frameworks

Compositional Scheduling and RTC based analysis techniques propose distinct approaches to hierarchical scheduling analysis. Conceptually, the computed interface and the technique to compute the interface are different in the compositional framework when compared to the RTC framework. For hierarchical scheduling analysis, the compositional framework derives an interface parameter (Θ, Π) (for PRM) or (Θ, Π, Δ) (for EDP) for each component in the hierarchy, which denotes how much budget (Θ) the component requires in every period (Π) time units. The interface parameter is derived using the dbf and sbf functions, which are time interval based functions. The dbf value for a set of tasks is a function of the workload of the tasks and the scheduling policy used. The sbf value is a function of the resource model used, i.e., PRM or EDP.

On the contrary, the interface parameter that is derived in RTC framework for schedulability of a component is the required lower bound on service curve (β^l). The interface parameter β^l is derived from the workload of the tasks given by the arrival curve α and the scheduling policy. Therefore, one thing significant to note is that RTC framework inherently does not use a resource model to derive β^l and sbf from compositional framework cannot be compared with any quantity in RTC framework unless we derive a service curve that takes the resource model into consideration.



(a) M_1 under RM



(b) M_2 under EDF

Fig. 15 Comparison between β^l and dbf for M_1 and M_2

But, it tums out that the lower bound on service curve considering the resource model would be exactly the same as the sbf function. Further, conceptually β^l is analogous to the dbf used in compositional framework as both are derived from the workload and the scheduling policy used. Hence, we present couple of plots to show how β^l compares to dbf for the workload in components M_1 and M_2 from Fig. 8. Figure 15 shows two comparisons between the service curves of β^l and the resource demands of dbf for M_1 and M_2 from Fig. 8. Note that Fig. 15b shows that β^l and dbf for M_2 exactly overlap each other, showing that the interface β^l of RTC computes the exact upper-bound of dbf for M_2 under EDF. Figure 15a shows the comparison between β^l and dbf for M_1 under RM (Rate Monotonic). dbf individually computes each demand of the tasks of M_1 under RM while β^l collectively computes the lower-bound of service curve satisfying the same workload. However, the lower-bound service curve satisfies the schedulability condition of Eq. 10 that requires the demand of every task in the workload is satisfied at least once up to the deadlines of individual tasks. According to this observation, the service curve in β^l can be used as dbf so that it is checked against a given supply bound function to check the schedulability of HSS components. Thus, if the service curve can compute more optimal resource requirements than dbf, it is possible to find a more optimal hierarchical scheduling system using the service curve of RTC than the classical compositional framework.

6 Summary

This chapter presents the compositional framework depending on resource modeling paradigms (namely BDR, PRM, EDP) and the RTC-based techniques independent from resource models for hierarchical scheduling analysis. The EDP resource model extends PRM with deadlines such that a workload is satisfied with less resource allocations when compared to PRM. The RTC-based technique computes a lower-bound of service curve that satisfies a given workload. The lower-bound of service curve of RTC is analogous to demand-bound functions of the compositional framework, and thus, we posit that it can be used together with supply bound functions to find more resource-optimal hierarchical scheduling systems.

References

S.K. Baruah, L.E. Rosier, R.R. Howell, Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor. Real-Time Syst. **2**(4), 301–324 (1990)

M. Chadli, J.H. Kim, A. Legay, L.M. Traonouez, S. Naujokat, B. Steffen, K.G. Larsen, A model-based framework for the specification and analysis of hierarchical scheduling systems, in *International workshop on formal methods for industrial critical systems* (Springer, 2016), Cham, pp. 133–141

S. Chakraborty, S. Künzli, L. Thiele, A general framework for analysing system properties in platform-based embedded system designs, in *Proceedings of Design Automation and Test in Europe (DATE)* (2003) pp. 190–195

S. Chakraborty, Y. Liu, N. Stoimenov, L. Thiele, E. Wandeler, Interface-based rate analysis of embedded systems, in *Proceedings of 27th IEEE Real-Time Systems Symposium (RTSS)* (2006). pp. 25–34

A. Easwaran, M. Anand, I. Lee, Compositional analysis framework using edp resource models, in *Real-Time Systems Symposium*. RTSS 2007. 28th IEEE International (IEEE, 2007). pp. 129–138

X. Feng, A.K. Mok, A model of hierarchical real-time virtual resources, in *Proceedings of 23rd IEEE Real-Time Systems Symposium (RTSS)* (2002). pp. 26–35

J.Y. Le Boudec, P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*, vol 2050 (Springer Science & Business Media, Berlin/Heidelberg, 2001)

J. Lehoczky, L. Sha, Y. Ding, The rate monotonic scheduling algorithm: exact characterization and average case behavior, in *Real Time Systems Symposium, 1989, Proceedings* (IEEE, 1989). pp. 166–171

C.L. Liu, J.W. Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment. J. ACM **20**(1), 46–61 (1973). https://doi.org/10.1145/321738.321743

A.K. Mok, X. Feng, D. Chen, Resource partition for real-time systems, in *Proceeding of 7th IEEE Real-Time Technology and Applications Symposium (RTAS)* (2001). pp. 75–84

L.T. Phan, J. Lee, A. Easwaran, V. Ramaswamy, S. Chen, I. Lee, O. Sokolsky, CARTS: a tool for compositional analysis of real-time systems. ACM SIGBED Rev. **8**(1), 62–63 (2011)

I. Shin, I. Lee, Periodic resource model for compositional real-time guarantees, in *RTSS* (2003). pp. 2–13

I. Shin, I. Lee, Compositional real-time scheduling framework with periodic model. ACM Trans. Embed. Comput. Syst. (TECS) **7**(3), 30 (2008)

N. Stoimenov, S. Perathoner, L. Thiele, Reliable mode changes in real-time systems with fixed priority or edf scheduling, in *Proceedings of Design, Automation and Test in Europe (DATE)* (2009). pp. 99–104

L. Thiele, S. Chakraborty, M. Naedele, Real-time calculus for scheduling hard real-time systems, in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)* (2000). pp. 101–104