Tech Report: Scheduling Periodic Tasks on Multiple Periodic Resources

Xiayu Hua, Zheng Li, Hao Wu, Shangping Ren
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616, USA
{xhua, zli80, hwu28}@hawk.iit.edu, ren@iit.edu

***Abstract*—In this paper, we study the problem of scheduling periodic tasks on multiple periodic resources. We take two step approach by first integrating multiple periodic resources to an equivalent single periodic resource so that existing real-time scheduling theorems on single periodic resource can be applied. Second, we extend the schedulability tests of periodic tasks on a single periodic resource in continuous time domain given in [1] to discrete and finite time domain so that the schedulability tests can be applied in practice. We further empirically study the performance of periodic resource integration. Experiment results reveal the following interesting behaviors: 1) increasing the number of periodic resources does not necessarily increase the integrated resource's capacity; 2) integrating smaller capacity periodic resources has higher capacity increase than integrating larger capacity periodic resources; and 3) integrating periodic resources that have the same capacity results in the most capacity increase in the integrated resource.**

I. Introduction

In real-time community, for a long time, the focus of scheduling problems has been on *dedicated* resources, the resources that are constantly available to tasks. However, as virtualization technology develops, resources, especially virtual resources, are not dedicated resource any more. The virtual resources are often modeled as periodic resources which are only available for certain amount of time within a given period [2], [3], [4], [5], [6].

The study of periodic resources can be traced back to the time when the concept of time-division sharing is proposed. Time-division sharing is one of the most important technology and also one of the most efficient way to accommodate multiple applications on a single resource [2], [7]. With time-division sharing scheduling algorithms, such as round-robin and server-based scheduling algorithms, resources manifested to applications are periodic resources. The purpose of these algorithms is to partition a dedicated resource to serve multiple applications, rather than how to schedule a task on a periodic resource. Research on the schedulability issue of multiple tasks on a single periodic resource started in late 90 [2] and has recently draw more attention in the community [1], [3], [4], [8], [9], [10], [11]. However, until now, there has not been much, if any, work in the literature dealing with the issue of scheduling a periodic task set on multiple periodic resources when none of the individual periodic resource has large enough capacity to support the given task set.

In this paper, we study the problem of scheduling periodic task set on multiple periodic resources. To address the issue, we take two steps, first study how multiple periodic resources can be integrated into an equivalent single periodic resource so that the existing real-time scheduling theorems on a single periodic resource can be applied. Second, we extend one of the existing schedulability test for a single periodic resource proposed by Lee [1] from continuous time domain to discrete and finite time domain so that the test can be of practice use. We further use extensive simulation to observe the resource capacity increase patterns when periodic resources of different capacities are integrated.

The rest of paper is organized as follows: Section II discusses the related work. Section III defines terms and models that the paper is based on and also formulates the problem to be addressed in the paper. The equivalent transformation of multiple periodic resources into a single periodic resource is discussed in Section IV. In Section V, we extend Lee's schedulability test [1] of multiple tasks on a single resource from continuous time domain to discrete and finite time domain. Section VI empirically studies the performance of integrating periodic resources with different capacities. Finally, we conclude in Section VII.

II. Related work

Real-time scheduling problem has been studied extensively for half century. Most of the researches on the problem are focusing on dedicated resources over past decades. As the computer technology develops, single resource nowadays can provide thousands of more computational capabilities compared to decades ago. Hence, the real-time scheduling problem is extended from how to schedule a group of tasks onto a dedicated resource to how to schedule multiple groups of tasks onto a dedicated resource. One of the most intuitive way to schedule multiple groups of tasks onto a single dedicated resource is to split the dedicated resource into several partitions. Then schedule each group of tasks to its corresponding partition. This is the essential concept of periodic resource [2].

Although the periodic resource model hasn't been formally defined until later 90s [2], the insight of periodic resource can be traced back as early as the concept of time-division sharing was proposed. Time-division is one of the most efficient way to distribute resources to tasks [2], [7]. Server-based scheduling mechanism is one of the most popular time-division mechanisms to accommodate multiple groups of tasks and provide some level of schedulability guarantees [12], and extensive research work [13], [14], [15] developed different mechanisms to improve tasks' response time.

Server-based mechanisms can be treated as strategies to adjust the resource partition to meet tasks' specific demands. Considering the resource partition is given and cannot be adjusted, Shirero *et al.* [2] first defined periodic resources and proposed a real-time round robin scheduling algorithm in 1999. They also introduced the concept of *resource regularity*. Based on resource regularity, they proposed schedulability bounds for periodic tasks. A.K. Mok *et al.* [3], [16] then extended Shirero's work and proposed a more comprehensive schedulability analysis for periodic resources under earliest deadline first (EDF) and rate monotonic (RM) scheduling algorithms. However, both Shirero and Mok's periodic resource model had constraints that either the resource pattern or the resource regularity should be given. By removing the constraints, Shin *et al.* then extended the periodic resource model to a more general case and provided a complete schedulability analysis under such model and gave the schedulability bounds for both EDF and RM accordingly [1], [10].

However, both Shirero's original periodic resource model and Shin's extended model were only trying to solve the real-time scheduling problems on single periodic resource. Instead, in this paper we investigate the problem of how to integrate multiple periodic resources into an equivalent single periodic resource so that the existing theorems for single periodic resource can be applied.

III. System Models and Problem Formulation

A. Terms and Definitions

Task Set Model Γ

A task set contains a set of periodic tasks, i.e., $\Gamma = \{\tau_1, ..., \tau_n\}$ with $\tau_i = (p_i, e_i)$, where p_i and e_i are the task's period and worst case execution time, respectively.

Resource Model R

The resource we consider is a periodic resource, represented as $R = (\pi, \theta)$, where π is a period and θ is the total amount of available time durations ($0 < \theta < \pi$) within each period.

Capacity of Resource $C(R)$

The capability of a periodic resource $R(\pi, \theta)$ is defined as $C(R) = \frac{\theta}{\pi}$.

Periodic Resource Pattern(P_R)

Given a periodic resource $R(\pi, \theta)$, its pattern P_R defines its time availability within its period, i.e., $P_R(t) = 1$ indicates that the resource is available at time t, or it is not available at time t if $P_R(t) = 0$.

Fixed Pattern Periodic Resource

If a periodic resource is of fixed pattern, then $P_R(t) = P_R(t + k \times \pi)$, where $k \in \mathbb{N}^+$.

Continues Periodic Resource

Given fixed pattern periodic resource $R = (\pi, \theta)$, it is a continues periodic resources if and only if $\forall i \in [0, \theta - 1], k \in \mathbb{N}^+, P_R(i + k \times \pi) = 1$.

Synchronized Periodic Resource

Two periodic resources are synchronized if the start points of the two periods are the same.

Equivalent Single Resource Transformation of Periodic Resource Set ($I(\Psi)$)

For a set of periodic resource $\Psi = \{R_1, R_2, ..., R_n\}$, its equivalent single periodic resource transformation is defined as $I(\Psi)$ with $\forall t \in [0, LCM_{1 \le i \le n}(\pi_i)], P_{I(\Psi)}(t) = P_{R_1}(t) \vee P_{R_2}(t) \vee ... \vee P_{R_n}(t)$.

B. Problem Formulation

Given a set of periodic resources $\Psi = \{R_1, R_2, ..., R_n\}$ with $R_i = (\pi_i, \theta_i)$ and a set of periodic tasks $\Gamma = \{\tau_1, \tau_2, ..., \tau_m\}$, decide schedulability test if the task set is schedulable on the resource set with EDF and RM, respectively.

We take two steps to address the problem, first, transform the periodic resource set into an equivalent single periodic resource (Section IV), and second apply extended single periodic resource schedulability test (Section V) to decide whether the given task set is schedulable on the given resources.

It is worth pointing out that the discrete time system is assumed, i.e., $\forall i p_i, e_i, \pi_i, \theta_i \in \mathbb{N}$. We also assume that tasks can not be executed parallelly, i.e., at each time instance t, tasks can only execute on one resource. Further more, we assume there is no overhead for task migration between different resources.

IV. Integration of Periodic Resources

Based on the definition of equivalent single periodic resource of a periodic resource set $\Psi = \{R_1, R_2, ..., R_n\}$, i.e., $P_{I(\Psi)} = P_{R_1}(t) \vee P_{R_2}(t) \vee ... \vee P_{R_n}(t)$, we can develop an algorithm to calculate $I(\Psi)$ as illustrated in Algorithm 1.

ALGORITHM 1: RES_INT ($\Psi = \{R_1, R_2, ..., R_n\}$)

Calculate the hyperperiod of Ψ as $\lambda = \prod_{i=1}^{n} \pi_i$
$P_{I(\Psi)}(t) = 0, \forall t \in \{0, 1, ..., \lambda - 1\}$
$t = 0$
while $t < \lambda$ **do**
  for $i = 1$ to n **do**
    $P_{I(\Psi)}(t) = P_{I(\Psi)}(t) \vee P_{R_i}(t)$
  end
  $t = t + 1$
end
return $P_{I(\Psi)}$

With Algorithm 1, we can obtain the equivalent single resource of a periodic resource set. However, Algorithm 1 needs to check whether each time unit is available or not within the hyperperiod λ, which is time consuming. In the following, we first give the upper and lower bound analysis of the equivalent single periodic resource's capacity, and then propose closed-form formula to directly calculate the equivalent single periodic resource if the periodic resources satisfy certain conditions.

Lemma 1: Given a set of periodic resources $\Psi = \{R_1, R_2, ..., R_n\}$, then the capacity of its equivalent single periodic resource, i.e., $C(I(\Psi))$, satisfies the following condition

$$\max\{C(R_i), 1 \leq i \leq n\} \leq C(I(\Psi)) \leq \min\{1, \sum_{i=1}^{n} C(R_i)\}$$

□

Proof: By definition, $P_{I(\Psi)}(t) = P_{R_1}(t) \vee P_{R_2}(t) \vee ... \vee P_{R_n}(t)$, hence, we have $P_{I(\Psi)}(t) \geq \max\{P_{R_i}(t), 1 \leq i \leq n\}$, therefore, $\max\{C(R_i), 1 \leq i \leq n\} \leq C(I(\Psi))$.

If the available time slots of each periodic resource are not overlapped with each other, then the capacity $C(I(\Psi))$ is the sum of each individual periodic resource, i.e., $C(I(\Psi)) = \sum_{i=1}^{n} C(R_i)$. However, if the available time slots of any two periodic resources are overlapped, then $C(I(\Psi)) < \sum_{i=1}^{n} C(R_i)$. In addition, the resource capacity cannot exceed 1, so we have $C(I(\Psi)) \leq \min(1, \sum_{i=1}^{n} C(R_i))$. ■

It is worth pointing out that both the upper and lower bounds are tight bounds. For instance, with $R_1(3, 2)$ and $R_2(3, 1)$ with $P_{R_1}(0) = P_{R_1}(1) = 1$, $P_{R_1}(2) = 0$ and $P_{R_2}(0) = 0$, $P_{R_2}(1) = P_{R_2}(2) = 1$. Then for the equivalent single resource $I(R_1, R_2)$, $C(I(R_1, R_2)) = C(R_1)$. Similarly, with $R_1(3, 2)$ and $R_2(3, 1)$ and $P_{R_1}(0) = P_{R_1}(1) = 1$, $P_{R_1}(2) = 0$ and $P_{R_2}(0) = P_{R_2}(1) = 0$, $P_{R_2}(2) = 1$. Then for the equivalent single resource $I(R_1, R_2)$, we have $C(I(R_2, R_2)) = C(R_1) + C(R_2)$.

When the periods of periodic resources are mutually prime, we can derive the following Lemmas to simplify the calculation of periodic resource integration.

We first consider a simple case where one of the periodic resource has only 1 time slot available in each period.

Lemma 2: Given two fixed pattern periodic resource $R_1 = (\pi_1, \theta_1)$ and $R_2 = (\pi_2, 1)$. If π_1 and π_2 are mutually prime, then $I(R_1, R_2) = (\pi', \theta')$, where $\pi' = \pi_1 \cdot \pi_2$ and $\theta' = \pi_2 \cdot \theta_1 + \pi_1 - \theta_1$.

Proof: Suppose the pattern of R_2 is $P_{R_2}(k + i \cdot \pi_2) = 1$ where i is the number of period. That is, the kth slot of every period π_2 is available. We first assume the following condition hold and will prove it later:

- if $i_1 \neq i_2 \wedge i_1, i_2 \in \{0, 1, ..., \pi_1 - 1\}$, then

$$((k + i_1 \cdot \pi_2) \bmod \pi_1) \neq ((k + i_2 \cdot \pi_2) \bmod \pi_1)$$

The condition indicates that $\forall k \in \{0, 1, ..., \pi_1 - 1\}$, the kth time slot of resource R_1's period π_1 is overlapped with the available time slot of R_2 once within their hyperperiod $\pi_1 \cdot \pi_2$.

As within the hyperperiod $\pi_1 \cdot \pi_2$, $\pi_2 \cdot \theta_1$ available time slots are provided by R_1, $\pi_1 \cdot 1$ available time slots are provided by R_2, while θ_1 time units are overlapped, hence, we have $\theta' = \theta_1 \cdot \pi_2 + \pi_1 \cdot 1 - \theta_1 \cdot 1$.

Now we prove that the above assumption holds.

Suppose that:

$$\exists i_1, i_2 : (k + i_1 \cdot \pi_2) \bmod \pi_1) = ((k + i_2 \cdot \pi_2) \bmod \pi_1)$$

then we have:

$$(k + i_1 \cdot \pi_2) \bmod \pi_1 - (k + i_2 \cdot \pi_2) \bmod \pi_1 = 0$$

which implies

$$(k + i_1 \cdot \pi_2 - (k + i_2 \cdot \pi_2)) \bmod \pi_1 = 0$$

hence, we have $(i_1 - i_2)\pi_2 \bmod \pi_1 = 0$.

Since $i_1 \neq i_2$ and $i_1, i_2 \in \{0, 1, ..., \pi_1 - 1\}$, then π_1 and π_2 are not mutually prime, which contradicts our assumption that π_1 and π_2 are mutually prime. Therefore, our assumption holds. ■

Lemma 2 gives the formulation to calculate the integration of two periodic resources when one of them only has one time slot available within each period, next, we remove this assumption and generalize the analysis to the integration of two fixed pattern periodic resources with arbitrary capacities.

Lemma 3: Given two fixed pattern periodic resources $R_1(\pi_1, \theta_1)$ and $R_2(\pi_2, \theta_2)$, if π_1 and π_2 are mutually prime, then $I(R_1, R_2) = (\pi', \theta')$ where $\pi' = \pi_1 \cdot \pi_2$ and $\theta' = \theta_1 \cdot \pi_2 + \theta_2 \cdot \pi_1 - \theta_1 \cdot \theta_2$.

Proof: According to Lemma 2, if R_2 has only one time slot available with each period, then R_1 and R_2 have θ_1 available time slots are overlapped within each hyperperiod $\pi_1 \cdot \pi_2$. Then if R_2 has θ_2 available time slots within each period, applying the similar proof of Lemma 2, we have $\theta_1 \cdot \theta_2$ available time slots are overlapped within each hyperperiod $\pi_1 \cdot \pi_2$. Since the total available time slots of R_1 and R_2 are $\theta_1 \cdot \pi_2 + \theta_2 \cdot \pi_1$, hence, we have $\theta' = \theta_1 \cdot \pi_2 + \theta_2 \cdot \pi_1 - \theta_1 \cdot \theta_2$. ■

Lemma 3 integrates two periodic resource, the following lemma generalize the calculation to multiple resources.

Lemma 4: Given a periodic resource set $\{R_1, R_2, ..., R_n\}$ with $R_i = (\pi_i, \theta_i)$, if all the periods of the periodic resources in this set are mutually prime with each other, then $I(R_1, R_2, ..., R_n) = (\pi'_n, \theta'_n)$, where

$$\theta'_n = \begin{cases} \theta_1 \cdot \pi_2 + \theta_2 \cdot \pi_1 - \theta_1 \cdot \theta_2 & \texttt{if } n = 2 \\ \theta'_{n-1} \cdot \pi_n + \theta_n \cdot \pi'_{n-1} - \theta_n \cdot \theta'_{n-1} & \texttt{if } n > 2 \end{cases}$$

where $\pi'_n = \prod_{i=1}^{n} \pi_i$.

Proof: With Lemma 3, we can obtain the equivalent single periodic resource of R_1 and R_2, i.e., $I(R_1, R_2) = (\pi_1 \cdot \pi_2, \theta_1 \cdot \pi_2 + \theta_2 \cdot \pi_1 - \theta_1 \cdot \theta_2)$. By treating $I(R_1, R_2)$ as a single resource, applying Lemma 3 to integrate resource R_3, we have $I(R_1, R_2, R_3) = (\prod_{i=1}^{3}, \theta'_2 \cdot \pi_3 + \theta_n \cdot \pi'_2 - \theta_3 \cdot \theta'_2)$. Follow the procedures, repeating the above steps until the resource R_n finishes the proof. ■

The previous lemmas have a constraint that resource periods are mutually prime. We now remove the constraint and discuss a general case for integrating two synchronized and continues periodic resources.

Lemma 5: Given two synchronized and continues periodic resource $R_1(\pi_1, \theta_1)$ and $R_2(\pi_2, \theta_2)$. Without loss of generality, we assume $\pi_1 \geq \pi_2$ and their hyperperiod be π'. Let $tail_i = (i + 1)\pi_1 \text{mod} \pi_2, i \in [0, \pi'/\pi_1 - 1]$ and $k_i = \lfloor \frac{\pi_1 - \theta_1 - tail_i}{\pi_2} \rfloor$, then integrated resource capacity within a hyperperiod is:

$$\theta' = \sum_{i=1}^{\pi'/pi_1} \theta_1 + \max\{\min\{\theta_2, tail_i\} - \max\{\theta_1 - (\pi_1 - tail_i), 0\}, 0\} + \max\{k_i\theta_2 + \max\{0, \pi_1 - tail_i - (k_i + 1) + \theta_2 - \theta_1\}, 0\} \quad (1)$$

Due to page limit, the proof of the lemma is given in Appendix.

V. Schedulability Analysis on Single Periodic Resource

For self-containment, we first introduce the existing work about schedulability test on single periodic resource given in [1].

A. Existing Schedulability Tests on Single Periodic Resource

Given a periodic resource R, the resource supply bound function $sbf_R(t)$ represents the minimum available time that resource R can guarantee to supply within any time interval of length t. It can be calculated as below [17]:

$$sbf_R(t) = \begin{cases} t - (k+1)(\pi - \theta) \ \texttt{if} t \in [(k+1)\pi - 2\theta, (k+1)\pi - \theta] \\ (k-1)\pi \ \texttt{otherwise} \end{cases} \quad (2)$$

where $k = \max(\lceil (t - (\pi - \theta))/\pi \rceil, 1)$.

In order to meet timing constraints of task set Γ, a demand bound functions $dbf_{\text{EDF}}(\Gamma, t, \tau_i)$ and $dbf_{\text{RM}}(\Gamma, t)$ under EDF and RM scheduling policies, respectively, represents the maximum possible resource demand that $\tau_i (\tau_i \in \Gamma)$ may required within time interval of length t in order to meet the timing constraint. They can be calculated as below [17]:

$$dbf_{EDF}(\Gamma, t) = \sum_{\tau_i \in \Gamma} \left\lfloor \frac{t}{p_i} \right\rfloor e_i \quad (3)$$

$$dbf_{RM}(\Gamma, t, \tau_i) = e_i + \sum_{\tau_k \in HP(\tau_i)} \left\lceil \frac{t}{p_k} \right\rceil e_k \quad (4)$$

where $HP(\tau_i)$ is the set of higher-priority tasks than $\tau_i \in \Gamma$.

With the resource demand bound functions and resource supply bound functions, the schedulability test under EDF and RM scheduling policies can be represented as follows:

Lemma 6: (Theorem 4.1 in [1]) Given $\Gamma = \{\tau_1, ..., \tau_n\}$ and a periodic resource is R, the task set Γ is schedulable under EDF if and only if

$$\forall t : 0 < t \leq LCM_\Gamma : \ dbf_{EDF}(\Gamma, t) \leq sbf_R(t) \quad (5)$$

where LCM_Γ is the least common multiple of p_i of all $\tau_i \in \Gamma$.

Lemma 7: (Theorem 4.2 in [1]) Given $\Gamma = \{\tau_1, ..., \tau_n\}$ and a periodic resource R, the task set is schedulable under RM if and only if

$$\forall \tau_i \in \Gamma, \ \exists t_i \in [0, p_i] : \ dbf_{RM}(\Gamma, t_i, \tau_i) \leq sbf_R(t_i) \quad (6)$$

It is not difficult to see that both the schedulability tests are in continuous time domain, hence, they are only of theoretic value and can not be directly applied as schedulability tests in practice.

B. Discrete Time Domain Schedulability Tests

In this subsection, we give corresponding discrete time domain schedulability tests.

1) Schedulability Test under EDF:

Lemma 8: Given $\Gamma = \{\tau_1, ..., \tau_n\}$ and a periodic resource R, the task set Γ is schedulable under EDF if and only if :

$$\forall t \in \Omega : dbf_{EDF}(\Gamma, t) \leq sbf_R(t) \quad (7)$$

where Ω is a time point set defined in formula (8) and sorted in ascending order:

$$\Omega = \bigcup_{j=1}^{n} \Phi_j, \Phi_j = \{1, 2, ..., \text{LCM}_\Gamma / p_j\} \quad (8)$$

where LCM_Γ is the least common multiplier of all p_i with $\tau_i \in \Gamma$.

Proof: We prove this Lemma by proving formula (7) $\leftrightarrow$ (6).

We first prove formula (7) $\rightarrow$ (6) and we prove by contradiction.

Suppose that formula (7) $\rightarrow$ (6) does not hold, i.e.,

$$\forall t \in \Omega, dbf_{EDF}(\Gamma, t) \leq sbf_R(t)$$

but,

$$\exists t', 0 < t' < LCM_\Gamma, dbf_{EDF}(\Gamma, t') > sbf_R(t')$$

Without loss of generality, we further assume $t_i < t' < t_{i+1}$, where t_i and t_{i+1} are consecutive points in Ω.

As both demand and supply bound functions are non-decreasing functions [17], hence, we have

$$sbf_{EDF}(t') \geq sbf_{EDF}(t_i) \quad (9)$$

and,

$$dbf_{EDF}(\Gamma, t') \geq dbf_{EDF}(\Gamma, t_i) \quad (10)$$

Based on the assumptions, we have:

$$dbf_{EDF}(\Gamma, t_i) \leq sbf_{EDF}(t_i) \quad (11)$$

and,

$$dbf_{EDF}(\Gamma, t') > sbf_{EDF}(t') \quad (12)$$

From formula (9) and formula (11), we have

$$sbf_{EDF}(t') \geq sbf_{EDF}(t_i) \geq dbf_{EDF}(\Gamma, t_i) \quad (13)$$

Since demand bound function $dbf_{EDF}(\Gamma, t)$ is a step function and the value is only changed at $t \in \Omega$, we have

$$dbf_{EDF}(\Gamma, t') = dbf_{EDF}(\Gamma, t_i) \quad (14)$$

From formula (13) and formula (14), we have

$$sbf_{EDF}(t') \geq dbf_{EDF}(\Gamma, t') \tag{15}$$

which contradicts the assumption of formula (11).

Now we prove formula (6) $\rightarrow$ (7). This proof is straightforward. Since $\Omega \subseteq [0, LCM_\Gamma]$, formula (6) $\rightarrow$ (7) holds. ∎

2) Schedulability Test under RM:

Lemma 9: Given a task set $\Gamma = \{\tau_1, ..., \tau_n\}$ and a periodic resource R, the task set is schedulable under RM if and only if

$$\forall \tau_i \in \Gamma, \exists t_0 \in \Omega_i : dbf_{RM}(\Gamma, t_0, \tau_i) \leq sbf_R(t_0) \tag{16}$$

Ω_i is a time point set defined in formula (17):

$$\Omega_i = \bigcup_{\forall \tau_k \in \text{HP}(\tau_i)} \Phi_{\tau_k}, \Phi_{\tau_k} = \{1, 2, ..., \lfloor (p_i/p_k) \rfloor\} \tag{17}$$

where $\texttt{HP}(\tau_i)$ is a task set which contains the tasks has higher priority than τ_i under RM scheduling policy.

Proof: We prove this Lemma by proving formula (7) $\leftrightarrow$ (16).

We first prove formula (7) $\rightarrow$ (16).

According to the definition of demand bound function $dbf_{RM}(\Gamma, t, \tau_i)$, i.e. formula (4), it is a staircase function and only rises at the time instants $t^* + \epsilon$, where $t^* \in \Omega_i$ and ϵ is very small and could be closed to 0. In addition, the supply bound function $sbf_R(t)$ is non-decreasing, hence, we have:

$$\begin{aligned}\exists t_0 \in \Omega_i :& dbf_{RM}(\Gamma, t_0, \tau_i) - sbf_R(t_0) \\ &= \min\{dbf_{RM}(\Gamma, t, \tau_i) - sbf_R(t) | 0 \leq t \leq p_i\}\end{aligned} \tag{18}$$

Hence, if

$$\exists t_i \in [0, p_i] : \quad dbf_{RM}(\Gamma, t_i, \tau_i) \leq sbf_R(t_i)$$

then

$$dbf_{RM}(\Gamma, t_0, \tau_i) - sbf_R(t_0) \leq dbf_{RM}(\Gamma, t_i, \tau_i) - sbf_R(t_i) \leq 0$$

therefore, we have:

$$\exists t_0 \in \Omega_i : dbf_{RM}(\Gamma, t_0, \tau_i) \leq sbf_R(t_0)$$

We now prove formula (16) $\rightarrow$ (7).

This proof is straightforward, since $\Omega_i \subseteq [0, p_i]$, formula (16) $\rightarrow$ (7) holds.

∎

VI. Experiments

In this section, we empirically study the behaviors of periodic resources integration.

The first experiment illustrates how the capacity of integrated periodic resource varies under different periodic resource set. In order to observe the variation of integrated periodic resource, we integrate two periodic resources (R_1, R_2) under different scenarios. For each periodic resource, we set nine different capacities ranging from 0.1 to 0.9. Hence, we have total 81 combinations for two periodic resources under different capacities. For each combination, we randomly generate 100 pair of periodic resources which do not have fixed patterns. Fig. 1 shows the variations of average integrated periodic resource capacity. X-axis represents the capacity of periodic resource R_2 and Y-axis represents the capacity of the integrated periodic resource. Each curve represents the variation of the capacity of integrated periodic resource that integrated by a fixed capacity periodic resource R_1 and R_2 which has changing capacity.

As indicated by Fig. 1, in general, the integrated resource has larger the capacity than each of composing periodic resource. However, integration of periodic resources with different capacities have different impact on the capacity of integrated periodic resource. For instance, two periodic resources with both capacities equal to 0.1 produce a integrated periodic resource with capacity 0.19, which is almost equal to the summation of the two individual resources. However, two periodic resources with both capacities equal to 0.9 produce a integrated periodic resource with capacity 0.99. About 90% of the periodic resource's capacity is wasted. This observation brings an interesting question: what type of period resources shall be integrated together to result in the most benefit?

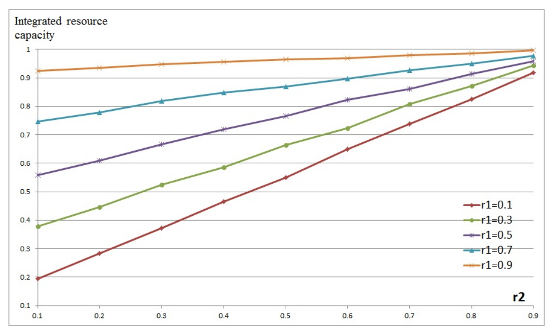

Fig. 1. The capacity of the integrated resource

To get an insight about the question, we first need to understand what is the benefit. We first define "benefit" as capacity increase ratio ρ as:

$$\rho(I(R_1, R_2) = \frac{C(I(R_1, R_2)) - \max(C(R_1), C(R_2))}{\max(C(R_1), C(R_2))} \tag{19}$$

Fig. 2 depicts the integrated periodic resources' capacity increase ratio. There is a very interesting observation that if one periodic resource integrates with another periodic resource that has the same capacity, the integrated periodic resource always has the highest capacity increase ratio. Furthermore, the closer the two periodic resources' capacities are, the higher the increase ratio.

From the previous observations, it is not difficult to see that the integration causes waste of periodic resources' capacities. Hence, we are also interested in knowing how integration causes waste. We measure waste as the overhead of an integration, which is defined as:

$$O(I(R_1, R_2)) = \frac{(C(R_1) + C(R_2)) - C(I(R_1, R_2))}{C(R_1) + C(R_2)} \tag{20}$$

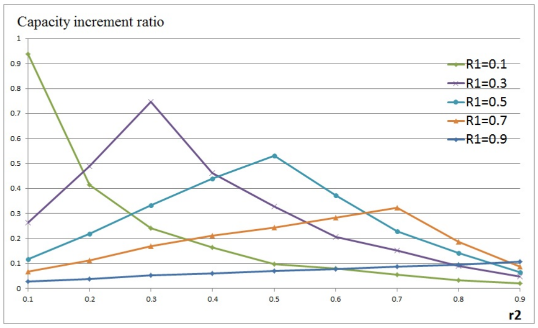

Fig. 2. The capacity increase ratio of the integrated resource

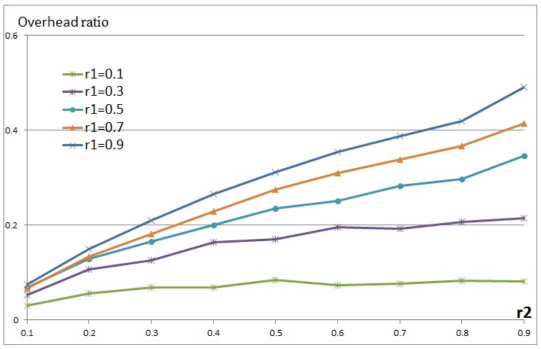

Fig. 3. Overhead of the integrated resource

Fig. 3 depicts the overhead. It clearly indicates that integrating small capacity periodic resource has lower waste ratio compared with resources with large capacities. Hence, we should avoid integrating large capacity periodic resources.

Till now, our experiments are mainly focusing on integrating two periodic resources. Next, we investigate the properties of periodic resource integration by integrating more than two resources. In this set of experiments, we fix the periodic resource set's total capacity, and then change the number of periodic resources. For each number, we randomly distribute the resource set's total capacity to each periodic resource. For each periodic resource set's total utilization and each number of periodic resources in the resource set, we repeat the experiment 50 times. Fig. 4 shows the average value of the results. The X-axis represents the total capacity for each periodic resource set and Y-axis represents the capacity of integrated periodic resource.

As shown in Fig.4, the resource integrated by two periodic resources has higher capacity compared with the resource integrated by three periodic resources. This is because the overlap situation between periodic resources happens more frequent when the number of periodic resources increases and hence makes the overhead larger.

VII. Conclusion

In this paper, we address the issue of scheduling a periodic task set on a set of periodic resources. We first transform a set of periodic resources into an equivalent single periodic

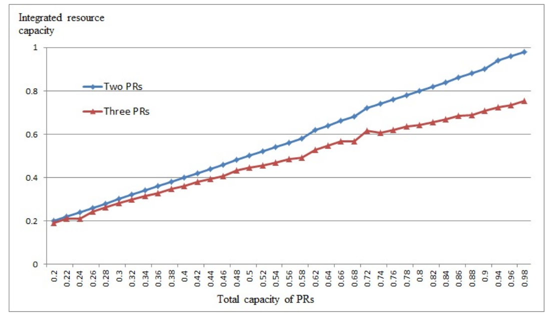

Fig. 4. Average utilization rate of integrated resources

resources and then apply extended schedulability test of multiple tasks on a single periodic resources. We also investigate the properties of periodic resource integration process and provide theoretic analysis for integrated periodic resource capacity. More specifically, we give the formal calculation of integrated periodic resource capacity for mutually prime periodic resources. In addition, we extend the existing schedulability test for periodic tasks on single periodic resource from continuous time domain to discrete time domain. We further experimentally study the behavior and performance of periodic resource integration. It is worth pointing out that there are some interesting observations from the experimental results: 1) increasing of the number of periodic resources does not necessarily increase the integrated resource's capability; 2) integration of small capacity periodic resources gains more benefit compared to the integration of large capacity periodic resources; and 3) integration of two periodic resources that have same capacities can maximize benefit.

Our future work is to take the task migration overhead into consideration and implement the proposed algorithm on the real virtualization platform, such as XEN, to utilize the resources of different virtual machines. Also, we plan to implement the proposed ideas on the Cloud platform in order to meet the requirement while taking minimum cost [18].

References

[1] I. Shin and I. Lee, "Compositional real-time scheduling framework with periodic model," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 7, no. 3, p. 30, 2008.

[2] S. Shirero, M. Takashi, and H. Kei, "On the schedulability conditions on partial time slots," in *Real-Time Computing Systems and Applications, 1999. RTCSA'99. Sixth International Conference on*. IEEE, 1999, pp. 166–173.

[3] A. K. Mok and X. Alex, "Towards compositionality in real-time resource partitioning based on regularity bounds," in *Real-Time Systems Symposium, 2001.(RTSS 2001). Proceedings. 22nd IEEE*. IEEE, 2001, pp. 129–138.

[4] A. K. Mok, X. Feng, and D. Chen, "Resource partition for real-time systems," in *Real-Time Technology and Applications Symposium, 2001. Proceedings. Seventh IEEE*. IEEE, 2001, pp. 75–84.

[5] Y. Li, A. M. Cheng, and A. K. Mok, "Regularity-based partitioning of uniform resources in real-time systems," in *Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012 IEEE 18th International Conference on*. IEEE, 2012, pp. 368–377.

[6] J. Lee, S. Xi, S. Chen, L. T. Phan, C. Gill, I. Lee, C. Lu, and O. Sokolsky, "Realizing compositional scheduling through virtualization," in *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2012 IEEE 18th*. IEEE, 2012, pp. 13–22.

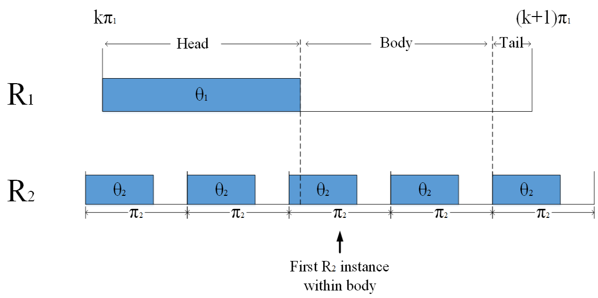

Fig. 5. Resource Integration

[7] W. Chu, "A study of asynchronous time division multiplexing for time-sharing computer systems," in *Proceedings of the November 18-20, 1969, fall joint computer conference*. ACM, 1969, pp. 669–678.

[8] N. Fisher and F. Dewan, "Approximate bandwidth allocation for compositional real-time systems," in *Real-Time Systems, 2009. ECRTS'09. 21st Euromicro Conference on*. IEEE, 2009, pp. 87–96.

[9] F. Dewan and N. Fisher, "Approximate bandwidth allocation for fixed-priority-scheduled periodic resources," in *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2010 16th IEEE*. IEEE, 2010, pp. 247–256.

[10] I. Shin and I. Lee, "Periodic resource model for compositional real-time guarantees," in *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*. IEEE, 2003, pp. 2–13.

[11] Xiayu Hua, Hao Wu, Zheng Li and S. Ren, "Scheduling periodic tasks on multiple periodic resources," in *Advanced Communications and Computation, 2014. The 4th International Conference on*. IARIA, 2014, pp. 2–13.

[12] S. Baruah, J. Goossens, and G. Lipari, "Implementing constant-bandwidth servers upon multiprocessor platforms," in *Real-Time and Embedded Technology and Applications Symposium, 2002. Proceedings. Eighth IEEE*, 2002, pp. 154–163.

[13] B. Sprunt, L. Sha, and J. Lehoczky, "Aperiodic task scheduling for hard-real-time systems," *Real-Time Systems*, vol. 1, no. 1, pp. 27–60, 1989.

[14] J. Strosnider, J. Lehoczky, and L. Sha, "The deferrable server algorithm for enhanced aperiodic responsiveness in hard real-time environments," *Computers, IEEE Transactions on*, vol. 44, no. 1, pp. 73–91, Jan 1995.

[15] L. Sha, J. P. Lehoczky, and R. Rajkumar, "Solutions for some practical problems in prioritized preemptive scheduling." in *RTSS*, 1986, pp. 181–191.

[16] X. Feng and A. K. Mok, "A model of hierarchical real-time virtual resources," in *Real-Time Systems Symposium, 2002. RTSS 2002. 23rd IEEE*. IEEE, 2002, pp. 26–35.

[17] I. Shin and I. Lee, "Compositional real-time scheduling framework," in *Real-Time Systems Symposium, 2004. Proceedings. 25th IEEE International*, 2004, pp. 57–67.

[18] Hao Wu, Xiayu Hua, Zheng Li and Shangping Ren, "Resource minimization for real-time applications using computer clouds," in *6th IEEE/ACM International Conference on Utility and Cloud Computing*, 2013, pp. 99–106.

Appendix A
Proof of Lemma 5

Proof:

To calculate the integrated resources for two synchronized periodic continues resources, we calculate each integration within each instance of R_1, and then add them together to get the final integrated resources within one hyperperiod. Fig. 5 dispatches the two resources within time interval $[k\pi_1, (k+1)\pi_1]$, where k is the k^{th} instance of R_1 within one hyperperiod. We partition the time interval into three parts, i.e., head, body and tail. The head part is from the beginning to the end of θ_1. The tail part is from the release time of the last instance of R_2 to $(k+1)\pi_1$. The body is the time interval between head and tail. In order to calculate the integrated resource within one period of R_1, we need to calculate all three parts.

Head:

The integrated resource in the head is trivial, which is θ_1.

$$\theta'_{head} = \theta_1 \tag{21}$$

Tail:

The length of the tail is $(k+1)\pi_1 \text{mod} \pi_2$. Let $tail_k = (k+1)\pi_1 \text{mod} \pi_2$. We first calculate the resources that R_1 has within the tail part. There are two cases needs to be considered: θ_1 is ended within the tail part or before the tail part. Notation θ_1^{tail} to denote the resources R_1 within the tail part. Then we can get the following:

$$\theta_1^{tail} = \max\{\theta_1 - (\pi_1 - tail_k), 0\} \tag{22}$$

We then calculate the resources R_2 has within the tail part. There are also two cases need to be considered: θ_2 ends within the tail part or exceeds the tail part. Denote θ_2^{tail} as the resources R_2 has within the tail part. Then we can get:

$$\theta_2^{tail} = \min\{\theta_2, tail_k\} \tag{23}$$

Since the whole θ_1 is calculated as head, we need to avoid over calculating θ_1 if it ends within the tail part. Hence, the integrated resource of the tail part after integration can be calculated as follow:

$$\theta'_{tail} = \max\{\theta_2^{tail} - \theta_1^{tail}, 0\} \tag{24}$$

Body:

We calculate the body part from the end point of the body part and subtract the empty slots within the body. Since the end of the body is $\pi_1 - tail_k$, there are $k' = \lfloor \frac{\pi_1 - \theta_1 - tail_k}{\pi_2} \rfloor$ entire R_2 instances within the body. Hence, we have $k'\theta_2$ time slots available from R_2.

For the first R_2 instance that within the body, i.e. the third R_2 instance in Fig. 5, there are two cases need to be considered. Case 1, θ_1 is ended at the middle of θ_2 of the first R_2 instance within the body part. The example shown in Fig. 5 falls into case 1. Case 2, θ_1 ends within the time interval of the first R_2 instance within the body part that doesn't provide any resources, i.e. in Fig. 5, if θ_1 ends between the end point of the third θ_2 and the beginning point of the fourth R_2 instance. The resources that the first R_{c2} instance within the body can be calculated as:

$$\theta_2^{body} = \max\{0, \pi_1 - tail_k - (k'+1)\pi_2 + \theta_2 - \theta_1\} \tag{25}$$

Note that, if the θ_1 ends within the tail part, there is no body part. And under such condition, $k' < 0$ and $\theta_2^{body} < 0$. Hence, the calculation of the body part is as follow:

$$\theta'_{body} = \max\{k'\theta_2 + \theta_2^{body}, 0\} \tag{26}$$

Combine equations (21), (26) and (24), we can get the integrated resources within any R_1 instance within the hyperperiod. Since within one hyperperiod, there are π'/π_1 R_1 instances, add all the integrated resources for each R_1 instance, we can easily get equation 1. ∎