The Journal of Systems and Software 110 (2015) 193–204

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Periodic resource integration

Xiayu Hua*, Zheng Li, Hao Wu, Chunhui Guo, Shangping Ren

Department of Computer science, Illinois Institute of Technology, 10 W 31st ST., Chicago, IL USA

ARTICLE INFO

Article history:
Received 4 February 2015
Revised 24 August 2015
Accepted 27 August 2015
Available online 8 September 2015

Keywords:
Real-time scheduling
Multi-resource scheduling
Periodic resource

ABSTRACT

Scheduling periodic real-time tasks on multiple periodic resources is an emerging research issue in the real-time scheduling community and has drawn increased attention over the last few years. This paper studies a sub-category of the scheduling problem which focuses on scheduling a periodic task on multiple periodic resources where none of these resources have sufficient capacity to support the task. Instead of splitting the task into sub-tasks, which is not always practical in real systems, we integrate resources together to jointly support the task. First, we develop a method to integrate two fixed but arbitrary pattern periodic resources into an equivalent periodic resource. Second, for two periodic resources with unknown but fixed resource occurrence patterns, we give the lower and upper bounds of the available time provided by an integrated periodic resource within a period. Third, we present theoretical and empirical analysis on the schedulability of a non-splittable periodic task on two periodic resources and their integrated periodic resource.

1. Introduction

As technology advances, computation power of individual computing devices has been elevated significantly over past decades. With the increasing computation capacity, multiple real-time applications can be deployed on the same computing device. As time granularities of different real-time application groups may be different, to avoid interferences among application groups from the scheduling perspective, the concept of *periodic resources* has been proposed (Shirero et al., 1999; Mok et al., 2001; Shin and Lee, 2003). A periodic resource provides fixed amounts of processing time periodically to an task set it serves. With this concept, one resource can be split into several isolated portions for multiple sets of real-time tasks. Such isolation allows scheduling algorithm design and schedulability analysis to be done independently for each task set and all sets are able to coexist on one resource. Based on the periodic resource scheduling theories, real-time schedulers in time-sharing systems, such as virtualization platforms with real-time virtual machines (Xi et al., 2011; 2014), can then be realized.

Although the scheduling issue of real-time periodic tasks on periodic resources was raised in the mid 90's, research works mainly focus on how to schedule multiple tasks on a *single* periodic resource. Efforts to deal with the scheduling issues on multiple periodic resources are rare and are often attached with strong conditions. From the experience the real-time community has had about single processor scheduling and multiple processor scheduling, we know that multiple processor scheduling problem is much more challenging than the single processor scheduling problem. In fact, as of today, we still do not have an equivalent understanding on multiple processor scheduling as we did on single processor scheduling. For instance, we still do not know under what conditions a scheduling anomaly may occur and when a relative priority among tasks may impact the schedulability of a given task set (Buttazzo, 2011; Liu, 2000). Similar challenging situations are anticipated for the study of scheduling tasks on multiple periodic resources.

Compared to the multiprocessor scheduling problem, scheduling periodic tasks on multiple periodic resources is a more difficult problem to handle because in case of the later, two distinct scenarios must be considered from the perspective of task utilization. These two scenarios are: (1) each periodic resource is large enough to support the largest task in the task set, and (2) no periodic resource is large enough to support any single task in the task set. Under the first scenario, Guo et al. have developed a heuristic scheduling algorithm, i.e., the Best-Harmonically-Fit scheduling algorithm (Guo et al., 2015) to schedule a given set of periodic tasks on a given set of periodic resources.

For the second scenario where no single periodic resource in the resource set is large enough to support the smallest task in the task set, a simple and intuitive approach is to split tasks into smaller subtasks (Guan et al., 2010; Fan and Quan, 2012). However, such splitting do not provide a viable solution for many reasons. First, it is possible that tasks are not splittable. Second, in order for the task splitting approach to work properly, subtask synchronization across different

---

* Corresponding author. Tel.: +13128236875.
E-mail addresses: xhua@hawk.iit.edu, xiayuhuahxy@gmail.com, _lion-heart_@live.cn (X. Hua), zli80@hawk.iit.edu (Z. Li), hwu28@hawk.iit.edu (H. Wu), cguo13@hawk.iit.edu (C. Guo), ren@iit.edu (S. Ren).

http://dx.doi.org/10.1016/j.jss.2015.08.050
0164-1212/

resources has to be done at run-time, which can incur great challenge and large time overhead as well. More importantly, task splitting does not solve the problem at the root level — it may require many levels of splitting.

Another approach yet to be explored in the literature is to integrate multiple periodic resources into a larger one. In this paper, we are to investigate this new approach to address the problem of scheduling non-splittable tasks on small periodic resources. The main contributions of this paper are three-fold: (1) develop a method to integrate two arbitrary fixed-pattern periodic resources into an equivalent periodic resource; (2) determine a lower and an upper bounds for the available time provided by an integrated periodic resource within its period; and (3) provide the schedulability analysis for an integrated periodic resource.

The organization of the paper is as follows: we first discuss related work in Section 2. In Section 3, we formally define the task model and the resource model this paper is based upon, and then formulate the problem that we are to address. Based on the models, we show in Section 4 that how resource available time is calculated for an integrated resource. In Section 5, we theoretically analyze both the lower and upper bounds of the available time of an integrated resource in a period. The theoretical schedulability analysis of a non-splittable task on an integrate resource is given in Section 6, followed by the experimental schedulability analysis in Section 7. We conclude the paper and point out our future work in Section 8.

2. Related work

The research on periodic resource can be traced back to 1999 when Shigero et al. first introduced the concept of periodic resource (Shirero et al., 1999). In contrast to continuous resources, which are constantly available to tasks running upon, periodic resources have time instances that the resources are not available. A periodic resource is often modeled as $R(\theta, \pi, \overrightarrow{V})$, where θ is the total available time within every π time interval, and vector $\overrightarrow{V}$ gives the time instance in a period where the resource is available. The vector $\overrightarrow{V}$ is also called resource occurrence pattern. If a resource occurrence pattern repeats among different resource periods, the resource is called *fixed-pattern* periodic resources, otherwise, it is called *dynamic-pattern* periodic resource. For fixed-pattern periodic resources, to reflect whether the resource occurrence is evenly distributed within its period, Shigero et al. introduced the concept of *resource pattern regularity*. Particular, for a periodic resource, if its occurrences are evenly distributed within its period, the periodic resource is called *regular* periodic resource, otherwise it is called *irregular* periodic resource.

With the fixed-pattern periodic resource model, Mok et al. (2001), Feng (2004) provided schedulability tests for both RM and EDF scheduling policies using task set demand bound function and resource least supply function. The least supply function is also known as supply bound function in the later works (Shin and Lee, 2003; Easwaran et al., 2007; Shin and Lee, 2008; Fisher and Dewan, 2012). Moreover, they extended the fixed-pattern periodic resource model and proposed the *bounded delay* resource model $R = (\alpha, \Delta)$ in Mok et al. (2001) where α is the capacity of the resource, i.e., $\alpha = \frac{\theta}{\pi}$, and Δ is maximum delay of the resource R. The schedulability analysis for the bounded delay resource model was also provided in the same paper.

In Feng (2004), Mok and Alex (2001), Mok and Feng (2002), Mok and Feng et al. also provided a formal definition for resource regularity introduced in Shirero et al. (1999) and a method to calculate the regularity for a resource occurrence pattern. For a *regular* periodic resource, they developed task set utilization bounds under both RM and EDF policies which are equal to the Liu's EDF and RM bounds provided in Liu and Layland (1973) multiplying by the resource capacity α, respectively. For a *irregular* periodic resource, they introduced a virtual time scheduling approach to transform an *irregular* periodic resource into an equivalent *regular* one.

In addition to the research on fixed-pattern periodic resources, dynamic-pattern periodic resource has also been studied in the literature. In particular, Shin and Lee (2003), Shin and Lee (2008) provided a formal definition of the dynamic-pattern periodic resource model. In their works, they developed an approach to abstract the resource requirement of a task set into the resource requirement of a single task under EDF and RM schedulers. In addition, they derived task set utilization bounds for the given periodic resource under both EDF and RM schedulers, respectively. Similarly, the resource capacity bounds for the given task set under both EDF and RM schedulers were also developed in their work.

For multiple periodic resources, Shin et al. (2008), Easwaran et al. (2009) studied the schedulability issues under a multiprocessor dynamic-pattern periodic resource (MPRs) model. The MPR model $R = \langle \pi, \theta, m' \rangle$ assumes that an unit-capacity and identical multiprocessor platform jointly provides θ units of resource in every π time instants while at most m' resources are available concurrently at any time instant. Under a global EDF scheduler, they further provided a schedulability analysis under the MPR model. Xi et al. later implemented the MPR model on XEN virtualization platform with both the global EDF and the partition EDF schedulers (Xi et al., 2014; 2011; Lee et al., 2012). Easwaran et al. also extended the dynamic-pattern periodic resource model introduced in Shin and Lee (2003) and (2008) to an Explicit Deadline Periodic (EDP) resource model in Easwaran et al. (2007). An EDP resource is described as $R = (\pi, \theta, \delta)$ which indicates the resource R provides θ units of resource within δ time units and the period of the resource is π. In their work, the authors also presented how to compose multiple EDP resources into a single equivalent EDP resource. Based on the EDP model, Nathan et.al developed a *fully-polynomial-time approximation* (FPTAS) scheme to find the suboptimal capacity assignment for multiple EDP resources with a given optimization accuracy (Fisher and Dewan, 2012). Yoon et al. (2013) further considered a special case of the EDP model where $\theta = \delta$.

For mutiple fixed-pattern periodic resources, Yu et.al developed the *Adjusted Availability Factor-Multi* (AAF-Multi) algorithm to arrange multiple periodic resources with known regularity on multiple processors (Li et al., 2012). The AAF-Multi prevents a periodic resource being mapped to more than one processor at the same time. In addition, the algorithm also preserves the schedulability bound given by Feng (2004) on a single processor. In our earlier work (Hua et al., 2014), we have developed a method to integrate periodic resources into one equivalent periodic resource for deploying the non-split and migratable tasks. However, our work is based on a strong assumption that the periods of the two periodic resources have to be co-prime. Therefore, a more generic integration method is required.

In this paper, we address the problem of scheduling a periodic non-splittable task on multiple periodic resources where none of them have sufficient capacity to support the task.

3. System models and problem formulation

In this section, we first define the models and terms that the paper is based upon. Then we formally state the problem we are to address.

3.1. Models and definitions

Task model: Tasks are periodic real-time tasks and are denoted as $\tau(e, P)$, where e and P are the task's execution time and period, respectively. Every task starts from time zero. We assume that the relative deadlines of tasks are equal to their periods. Task instances are allowed to migrate among different resources at run-time.

For a task instance, we assume it cannot be further split into sub-instances and be executed concurrently. We also assume that the time cost of task migration is negligible and is included in the task's