A Schedulability Analysis Framework for Real-time Infrastructure Systems Managing Heterogeneous Resources

Jinhyun Kim and Sungwon Kang
Department of Computer Science
KAIST
Daejeon, Korea
{jinhyun.kim, sungwon.kang}@kaist.ac.kr

***Abstract*—Electricity generating systems, such as smart grid systems, and water management systems are infrastructure systems that manage resources critical to human life. In the systems, resources are produced and managed to supply them to various consumers, such as building, car, factory, and household, according to their needs and priorities. Reliable supply of resources depends not only on sufficient production of resources but also on reliable sharing of resource supply facilities. This paper presents a schedulability analysis framework. A prominent characteristic of the framework is that it considers at once the two types of resources, i.e. consumable resources, such as electricity, energy, and water, and sharable resources, such as pipelines, storages, and processors, are considered. To apply a formal approach to schedulability analysis of infrastructure system, this paper classifies the types of resources and real-time jobs for infrastructure systems. Then based on the classification , it presents an architectural model and a schedulability analysis framework.**

***Keywords*-Resource, Heterogeneous resources, Schedulability analysis, Timing analysis, Formal methods**

I. Introduction

The computer-supported resource management systems, such as smart grid systems, has enabled more precise and elaborate management, distribution, and control of resources. Such a computer-supported technology for resource management will be widely applied in the future to various management systems for resources, such as energy, water, and gas. Reliable supply of consumable resources including electricity, water, and energy, depends on both sufficient production of consumable resources and reliable sharing of limited sharable (non-consumable) resources, such as pipelines, storages, and processors. That is, even though producers produce enough resources, consumers may not be able to consume necessary resources according to their consumption schedules if resources are not delivered on time.

Schedulability is a system property that every task in the system will be completed by its own deadline. So far, most of the studies on schedulability analysis have been limited to shared resources, such as CPU, memory, bus, etc., and have checked if all the tasks sharing limited resources will be completed by their deadlines, ignoring the schedules, amount, and priorities of production and consumption of consumable resources. Moreover, production and consumption of resources should be scheduled considering availability and capacity of supply facilities as well.

This paper presents a schedulability analysis framework that considers sharable resource and consumable resources together. For reliable supply of resources, our frameworks considers not only the production and consumption schedules of consumable resources but also sharing of sharable resources according to priorities. To that end, this paper first classifies the types of resources, and then classifies the types of jobs dealing with the resources. Finally, it presents an architectural model for heterogeneous resources management systems based on the classification of resources and jobs.

The rest of this paper is organized as follows: In Section 2, the problem that we deal with is specified with an example. Section 3 identifies the types of resources and real-time jobs of infrastructure systems. Section 4 proposes a schedulability analysis framework and an architecture model for infrastructure systems. In Section 5, the related works are surveyed. In Section 6, we conclude this paper.

II. The Problem Domain

A. Domain Description

This paper deals with a scheduling problem that arises in infrastructure systems, such as electricity, energy, water resource management and drain facility systems. In such systems, producers generate resources with specific plans or patterns and consumers consume them with various consumption plans or patterns. Resources generated by producers are delivered to consumers over supply facilities, such as pipeline, wire, storage, tank, transformer, etc. Supply facilities are shared by one or more producers to deliver resources exclusively or non-exclusively. Some of them may lose a specific amount of resources during delivery. In this paper, a system managing various types of resources is called a heterogeneous resource-constrained system.

The recent smart grid systems are typical examples of heterogeneous resource-constrained systems. Such systems need the following capabilities[1]:

- Handling uncertainties in schedules and power transfers across regions, such as villages, cities, and nations.
- Optimizing the transfer capabilities of the transmission and distribution networks and meeting the demand for increased quality and reliable supply

Moreover, such a system needs to manage the priorities between consumers. In the case of power outage, consumers with high priorities, such as hospitals and defence facilities, should be preferred to the consumers with low priorities with consumers in being supplied with power.

B. Problem Description

A heterogeneous resource-constrained system has three main components: producers, consumers, and supply facilities, which, respectively, have the following characteristics:

- **Producers** produce resources periodically according to their schedules.
- **Consumers** consume necessary resources periodically according to their schedules.
- **Supply facilities**s are available to multiple users simultaneously or exclusively.

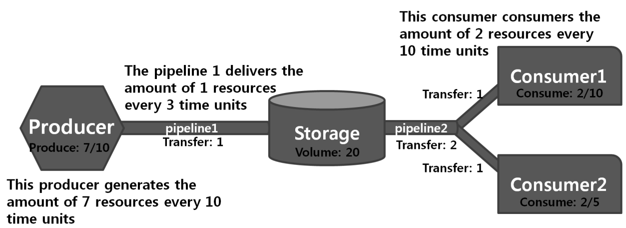

Figure 1. An Example Problem Model

Figure 1 depicts an example of the problem we consider in this paper. It consists of one producer and two consumers that share storage and pipelines. In this system, Producer1 generates the amount of 7 resources every 10 time units. Consumer1 and Consumer2 consume the same amount of resources with two different time intervals: 10 time units for Consumer1 and 5 time units for Consumer2. The storage can contain the amount of 20 resources. Pipeline1 is always available for Producer1, but Pipeline2 is available for both Consumer1 and Consumer2 but exclusively. Thus when they are about to receive resources from Pipeline2 simultaneously, one of them with the higher priority firstly receives resources over the pipeline and the other has to wait for Pipeline2 to be available.

In Figure 1, the problem we face is the following:

> **“Can schedules of production, consumption, and using supply facilities be always met (i.e. executed without violating their schedules)?"**

In order to answer the question, the system should be checked with respect to the following requirements:

- Producers can produce sufficient amount of resources according to schedules.
- Every consumer can consume specific quantities of resources within certain deadlines.
- All supply facilities can be scheduled for producers and consumers to supply resources from producers to consumers on time.

Extending the classical definition of schedulability for sharable resources, we view sharable resources as supplyig facilities that are used to deliver resources. Based on the problem description above, we define schedulability as follows:

> **“All the producers and consumers in a given system are, receptively, able to produce and consume necessary resources following their production and consumption schedules (plans), using the supply facilities according to the giving scheduling policies."**

III. Classification of Resources and Jobs for Infrastructure Systems

A. Resource Classification

This paper classifies resources into two main categories: sharable resource and consumable resource. Figure 2 shows our classification of resources for infrastructure systems.

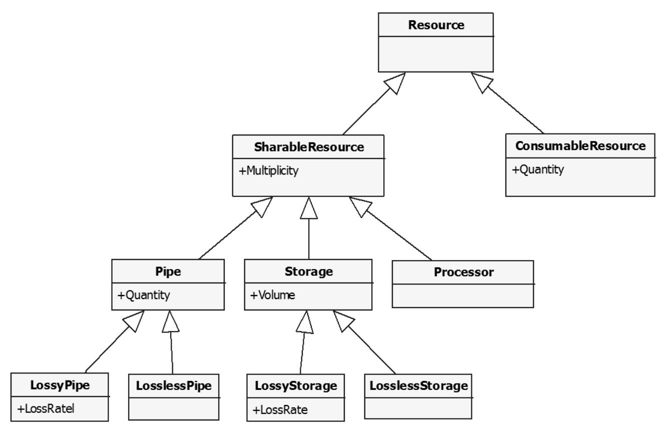

Figure 2. Taxonomy of Resources

Consumable resource is a resource that can be produced by producers and consumed by consumers. The consumable resource may disappear or be transformed into different types of resources by a consumer.

In order to supply and consume consumable resources, consumers and producers may use supply facilities, such as pipelines and storages, that can be shared by two or more participants. **Sharable resource** is a resource that can be used by one or more users. After using such resources, users must yield the right to use them to others. As sharable resources, there are CPU, memory, bandwidth, bus, networks in computer systems, etc. Furthermore, it includes pipelines

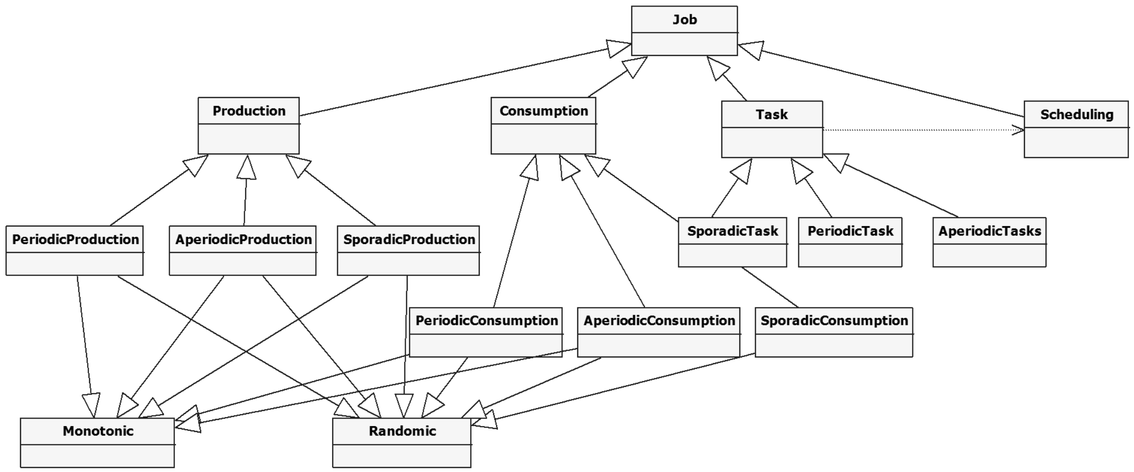

Figure 3. Taxonomy of Jobs

and storages for supply of resources. A sharing policy (scheduling policy) for sharable resources may allow a sharable resource to be preempted by another user even while being used by an user.

Sharable resources can be divided into three different types: link-type, storage-type and processor-type. **Link-type resource** is used to deliver consumable resource from one place to another. **Storage-type resource** is used to store consumable resource, being placed between producers and consumers. **Processor-type resource** is a resource that is used to produce and consume consumable resources.

B. Job Classification

For timing analysis of a heterogeneous resource-constrained system, Figure 3 identifies and classifies the types of jobs related to resources as classified in Section II.A. Job is defined as a unit of work that is scheduled and executed by a system. Jobs are classified into four main types: **Production**, **Consumption**, **Task**, and **Scheduling**.

Production and **Consumption** jobs produce and consume resources. Each of them is either periodic, aperiodic, or sporadic jobs.

Production of resource can be monotonic or randomic. **Monotonic production of a resource** is a production in which the planned amount of resources is generated monotonically with the same production rate by a deadline. That is, the same amount of resources is generated every time unit. A production of a resource for sporadic and aperiodic production generates resources uniformly until its deadline. Meanwhile, **Randomic production of a resource** is a production in which a randomic amount of resources is generated every time unit, but the required amount of resources should be generated for that period. The consumption of resources are also classified into periodic, aperiodic, and sporadic, and further into monotonic and randomic consumption.

A **Scheduling job** arbitrates tasks for using sharable resources according to a specific scheduling policy. Regardless of its type, a sharable resource needs to be managed by a scheduler for a task to use it. Thus, as shown in Figure 3, Task jobs depend on Scheduling jobs. Moreover, Production and Consumption jobs depend on Task jobs because resources become available only when being delivered on time by proper use of sharable resources.

Task jobs are dedicated to sharing or use of resources for the delivery of consumable resources. Task jobs are classified into three categories: sporadic, periodic, and aperiodic jobs similarly with the traditional classification of a task in real-time systems.

This paper extends the traditional classification of tasks in terms of the sharable resource types, link, storage, and processor. In terms of the types of sharable resources, a task is further classified into a **Process**, a **Store** and a **Deliver** job. A **Process job** executes actions using processor-type resources. For example, CPU and memory are resources which jobs in the class use for their execution. A job in the class of Process is a task which has been focused on by traditional studies on schedulability analysis of systems. A **Store job** stores consumable resources for specific time. It uses storage-type sharlable resources to store resources. A job in the class of Store depends on the volume of storage-type resources rather than on the preemption of the resources. For this reason, it is reasonable to distinguish jobs performed by Store and Process. A **Deliver job** delivers consumable resources via link-type sharable resources. A Deliver job can have either an Exclusive Use or a Non-Exclusive Use. A job of Exclusive Use uses a sharable resource exclusively. For example, a pipeline can be dedicated to a sole consumer or shared by more than one consumers. The whole amount of the pipe is used by the sole consumer when pipe has Exclusive Use, while a specific part of the pipe is used by a consumer when it has Non-Exclusive Use. A Store job or

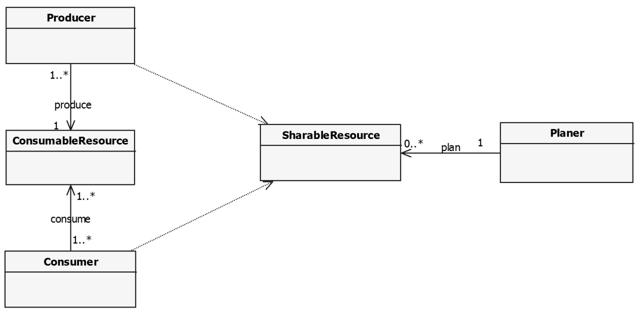

Figure 4. A Meta-model for Heterogeneous Resource-constrained System

a Deliver job can lose consumable resources over sharable resources so they can be either lossless or lossy.

Based on this classification of resources, a meta-model for heterogeneous resource-constrained systems representing infrastructure systems is depicted in Figure 4. There are three main participants for resources: Producer, Consumer, and Planner. They execute actions in interdependent ways. Thus **Producer** can produce resources in the way that its generated consumable resources need be properly delivered according to the schedules for sharable resources. Then **Consumer** can be supplied with resources. Sharable resources are shared by multiple tasks, and each of them should be managed by a specific scheduler. Thus, there need be a manager that controls schedulers and tasks. Just like operating systems, **Planner** is responsible for managing and executing tasks and schedulers. Planner includes multiple schedulers, and those which execute tasks to manage, control, and distribute sharable resources for producers and consumers.

IV. Architectural Model for Infrastructure Systems and Schedulability Analysis Framework

A. Architecture Model

In order to perform the schedulability analysis of heterogeneous resource-constrained systems, this paper presents an architectural model and the attributes of the elements of the architectural model.

A component can encapsulate one or more producers, consumers, or systems. A junction is used for joint or disjoint of two links. The attributes of links starting from a junction can differ from those of links which ends at the same junction.

Figure 5 shows our notations for the architectural model of heterogeneous resource-constrained systems.

There are five different attributes of elements in the architectural model as follows:

- ***prd***: period,
- ***pri***: priority,
- ***qnt***: quantity to produce or consume
- ***vol***: volume to store or deliver, and
- ***loss***: loss rate

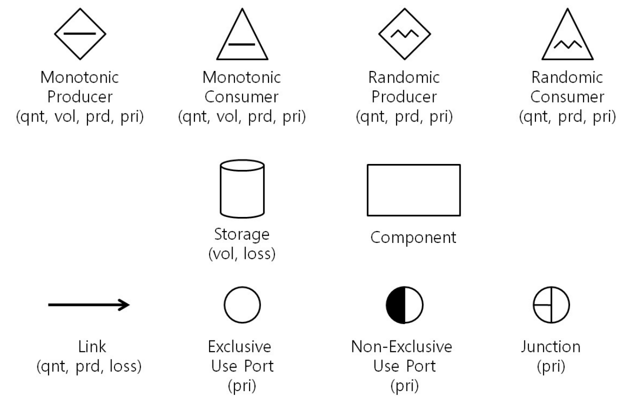

Figure 5. Notations and Attributes for the Elements of Architecture Models

A producer can be monotonic or randomic and has four attributes, (*qnt, vol, prd, pri*), which represent respectively the quantity of resources to produce, a local storage, a production period, and its priority to use shared resource. It carries out delivery of resources when supply facility to deliver resources is available, thus it stores generated resources in its local storage for a moment. A production fails if it cannot store resources at its local storage due to its fullness. A monotonic or randomic consumer also has the same attributes, except for a local storage. A consumption fails if it cannot consumer a specific amount of resources by deadline. A storage has two parameters, (*vol, loss*), which represent respectively the volume of resources and the loss rate of a link. The loss rate is 0 if the storage is lossless. A link has the attributes (*qnt, prd, loss*). *qnt* represents the amount of resources that can be delivered over a link per *prd* with *loss*.

A port is an agent of a sharable resource that determines how a resources should be shared and presents how it interacts with other corresponding ports. The shared resources like links and storages are utilized by producers and consumers according to two types of ports, which are Exclusive Use and Non-Exclusive Use. A port of Exclusive Use allows a producer or consumer to exclusively use a sharable resource. A port of Non-Exclusive Use, on the contrary, allows two or more producers and consumers to share a resource simultaneously. In short, which producer or consumer is to use or share a port by their priorities and which port is to use a link or storage by the priorities of ports. Ports and junctions have priorities for preemption and sharing of resources, and those with higher priorities have priority over others with lower priorities in using resources. Ports of Exclusive Use with higher priorities can occupy resources preferentially over those with lower priorities, and ports of Non-Exclusive Use with higher priorities can send more amount of resources than those with lower priorities.

Figure 6 shows an example of an architectural model of a heterogeneous resource-constrained system. According

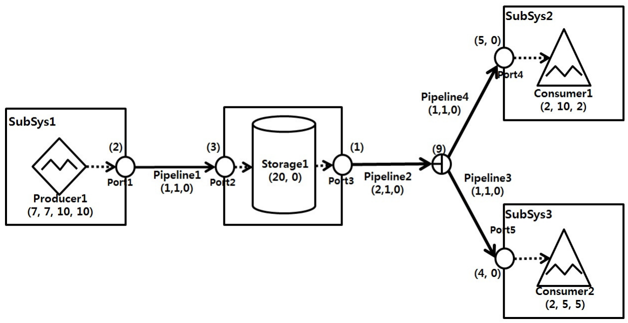

Figure 6. Architecture Model for an Infrastructure System

to the model, a randomic producer Producer1 produces the amount of 7 resources every 10 time unit. The port Port1 has the priority 2 for the link Pipeline1. The priority of Producer1 is useful when more than one producers are about to access the same port as with port1. A link-type resource, Pipeline1, is lossless and conveys the amount of 1 resource per 1 time unit. A storage-type resource, Storage1, receives resources from Port2 and transfer to Port3. The links connected to Port2 and Port3 have different flow rates so Storage1 provides different flow rates for entering resources and exiting resources. Storage1 has the capacity of 20 to store resources and is lossless. Consumer1, a randomic consumer, is supplied with resources through Pipeline2 and Pipeline4 from Storage1 while a randomic consumer. Consumer2 is supplied with the same resources through Pipeline2 and Pipeline3. However, Consumer1 is preferred to Consumer2 in being supplied with resources because the port connected to Consumer1 has a higher priority over the port connected to Consumer2.

B. Schedulability Analysis Framework

Scheduling assigns sharable resources according to a scheduling policy for producers and consumers to produce and consume consumable resources on time. A component of the system is defined as $C(J, R, A)$, where J is a set of jobs that should be carried out with the resources R under the scheduling policy A. It represents a collective requirement for production and consumption plans and available sharable resources. For the system, a demand-bound function of a consumer $\mathtt{dbf_C}(t)$ is defined as the maximum possible resource demand of consumers to satisfy timing requirements of all individual consumers within the time interval t. For the system, a supply-bound function $\mathtt{sbf_P}(t)$ is defined as the maximum possible resources supplies that producers provide during the time interval t. In our framework, all resources that are generated by producers are not immediately available for consumers. Thus, for immediately available resources for consumers, a resource available function $\mathtt{avf_C}(t)$ specifies the minimum amount of resources immediately available for consumers. Besides resources generated by producers, there exist resources that are being delivered or accumulated on supply facilities, such as storages. With respect to the resource under delivery over the system, a constant value denoted by Ω specifies the maximum amount of resources that are accumulated or under delivery on the system. That is, the amount of resources produced by producers can never exceed Ω. A component $C(J, R, A)$ is said to be *schedulable* if $\mathtt{dbf_C}(t) \leq \mathtt{avf_C}(t)$ and $\mathtt{sbf_P}(t) \leq \Omega$ for any interval $t > 0$.

Now, we can define the *schedulability* of the systems as follows: the system is said to be schedulable if and only if the maximum amount of available resources satisfies the maximum resource demand of consumers and the maximum amount of generated resources never exceeds the maximum amount of containable amount of resources, i.e.,

$$\forall t \;\; \mathtt{dbf_C}(t) \leq \mathtt{avf_C}(t) \leq \Omega \wedge \mathtt{sbf_P}(t) \leq \Omega,$$

where t is a time interval.

In Figure 6, for example, Ω is 20, the maximum amount of resources that Storage1 can contain. Producer1 can generate resources until the storage Storage1 is full, and the consumers Comsumer1 and Consumer2 can consume resources until Storage1 is empty. For Producer1, the system is not *schedulable* if Producer1 cannot generate resources by its deadline because Storage1 is full and therefore Producer1 cannot push the generated resources into the storage any more. For Consumer1 and Consumer2, the system is not *schedulable* if they cannot consume resources from Storage1 by their deadlines because the storage is empty and therefore they cannot be supplied with resources.

V. Related Works

There are various scheduling problems related to scheduling policies and their optimality for single or multiple resources. For scheduling for single resource, Liu and Layland[2] derives schedulability conditions for Rate Monotonic (RM) and Earliest Deadline First (EDF). Afterwards, various extensions of their studies, such as [3][4][5][6][7], have been performed in order to check schedulability according to specific scheduling algorithms, to enhance the optimality of scheduling algorithms, and to avoid deadlock of systems. For scheduling and optimal use of multiple resources, some studies, i.e. [8][9], focused on adapting existing uniprocessor scheduling algorithm to multiprocessor scheduling. Other Studies that include [10][11][12][13], introduced the ways of generating optimal schedules for implicit deadline task system on multiprocessor systems. For a hierarchical scheduling of a single resource, Goyal et al.[14] first proposed a hierarchical scheduling framework for supporting different scheduling algorithms for different application classes[15]. Shin et al.[15] presented a compositional scheduling framework. Lee et al.[16] proposed a formalism for resource-constrained real-time systems with which we can specify prioritized relations of concurrent processes for shared resources and verification of the system.

For scheduling of multiple types of resources, Sokolsky et al.[17] presents an extension of the work in [16] that considers both power consumption and scheduling problem, but they didn't include power production in their work.

For the smart grid system, strategies to achieve smart grid performance at the transmission level include the design of analytical tools and advanced technologies with intelligence for performance analysis such as dynamic optimal power flow, robust state estimation, real-time stability assessment, and reliability and market simulation tools[1].

Compared to the previous studies, our framework is unique in that it considers not only availability of resources for consumers but also production and consumption of resources for the schedulability analysis for infrastructure systems with real-time constraints.

VI. Conclusion

For the schedulabilty analysis for real-time jobs in infrastructure systems, this paper presents a classification of heterogeneous resources and jobs for the systems. Based on the classification, it proposes an architectural model, attributes of its elements and a schedulability analysis framework for such systems.

For the schedulability analysis of infrastructure systems, this paper makes the following contributions:

- It considers both real-time characteristics of consumable resources and sharable resources for real-time jobs in infrastructure systems,
- It classifies the types of resources and the types of jobs for the real-time jobs,
- It presents an architectural model for description of composition of the real-time jobs, and
- It proposes a schedulability analysis framework for the architectural model.

Acknowledgment

This research was supported by the KAIST High Risk High Return Project (HRHRP).

References

[1] J. Momoh, *Smart Grid: Fundamentals of Design and Analysis*, ser. IEEE Press Series on Power Engineering. John Wiley & Sons, 2012.

[2] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.

[3] L. Sha, R. Rajkumar, J. Lehoczky, and K. Ramamritham, "Mode change protocols for priority-driven preemptive scheduling," *Real-Time Systems*, vol. 1, pp. 243–264, 1989. [Online]. Available: http://dx.doi.org/10.1007/BF00365439

[4] B. Sprunt, L. Sha, and J. Lehoczky, "Aperiodic task scheduling for hard-real-time systems," *Real-Time Systems*, vol. 1, pp. 27–60, 1989. [Online]. Available: http://dx.doi.org/10.1007/BF02341920

[5] L. Sha, R. Rajkumar, and J. Lehoczky, "Priority inheritance protocols: an approach to real-time synchronization," *Computers, IEEE Transactions on*, vol. 39, no. 9, pp. 1175 –1185, sep 1990.

[6] M.-I. Chen and K.-J. Lin, "Dynamic priority ceilings: a concurrency control protocol for real-time systems," *Real-Time Syst.*, vol. 2, no. 4, pp. 325–346, Oct. 1990. [Online]. Available: http://dx.doi.org/10.1007/BF01995676

[7] J. W. S. Liu and R. Ha, "Advances in real-time systems," S. H. Son, Ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1995, ch. Efficient methods of validating timing constraints, pp. 199–224.

[8] R. Rajkumar, L. Sha, and J. Lehoczky, "Real-time synchronization protocols for multiprocessors," in *Real-Time Systems Symposium, 1988., Proceedings.*, dec 1988, pp. 259 –269.

[9] A. Srinivasan and S. Baruah, "Deadline-based scheduling of periodic task systems on multiprocessors," *Inf. Process. Lett.*, vol. 84, no. 2, pp. 93–98, Oct. 2002.

[10] H. Cho, B. Ravindran, and E. D. Jensen, "An optimal real-time scheduling algorithm for multiprocessors," in *Proceedings of the 27th IEEE International Real-Time Systems Symposium*, ser. RTSS '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 101–110.

[11] B. Andersson and E. Tovar, "Multiprocessor scheduling with few preemptions," in *Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, ser. RTCSA '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 322–334.

[12] B. Andersson and K. Bletsas, "Sporadic multiprocessor scheduling with few preemptions," in *Proceedings of the 2008 Euromicro Conference on Real-Time Systems*, ser. ECRTS '08. Washington, DC, USA: IEEE Computer Society, 2008, pp. 243–252.

[13] J. Lee, A. Easwaran, I. Shin, and I. Lee, "Multiprocessor real-time scheduling considering concurrency and urgency," *SIGBED Rev.*, vol. 7, no. 1, pp. 5:1–5:5, Jan. 2010.

[14] P. Goyal, X. Guo, and H. M. Vin, "Readings in multimedia computing and networking," K. Jeffay and H. Zhang, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, ch. A hierarchical CPU scheduler for multimedia operating systems, pp. 491–505.

[15] I. Shin and I. Lee, "Compositional real-time scheduling framework with periodic model," *ACM Trans. Embed. Comput. Syst.*, vol. 7, no. 3, pp. 30:1–30:39, May 2008.

[16] I. Lee, P. Bremond-Gregoire, and R. Gerber, "A process algebraic approach to the specification and analysis of resource-bound real-time systems," *Proceedings of the IEEE*, vol. 82, no. 1, pp. 158 –171, jan 1994.

[17] O. Sokolsky, A. Philippou, I. Lee, and K. Christou, "Modeling and analysis of power-aware systems," in *Tools and Algorithms for the Construction and Analysis of Systems*, ser. Lecture Notes in Computer Science, H. Garavel and J. Hatcliff, Eds. Springer Berlin Heidelberg, 2003, vol. 2619, pp. 409–424.