Compositional Schedulability Analysis

Insup Lee

PRECISE (Penn Research in Embedded Computing and Integrated Systems Engineering) Center

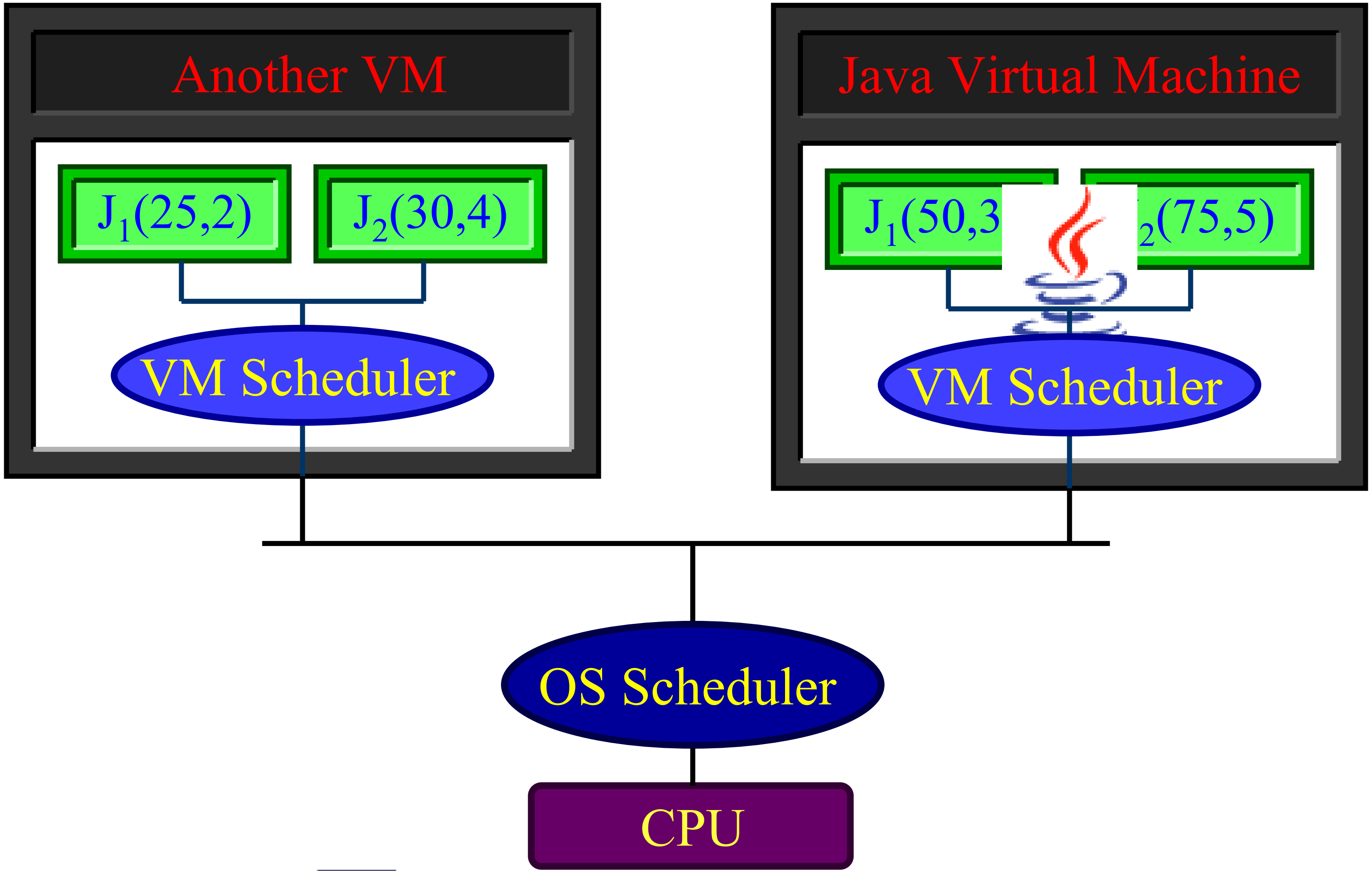
Department of Computer and Information Science

University of Pennsylvania

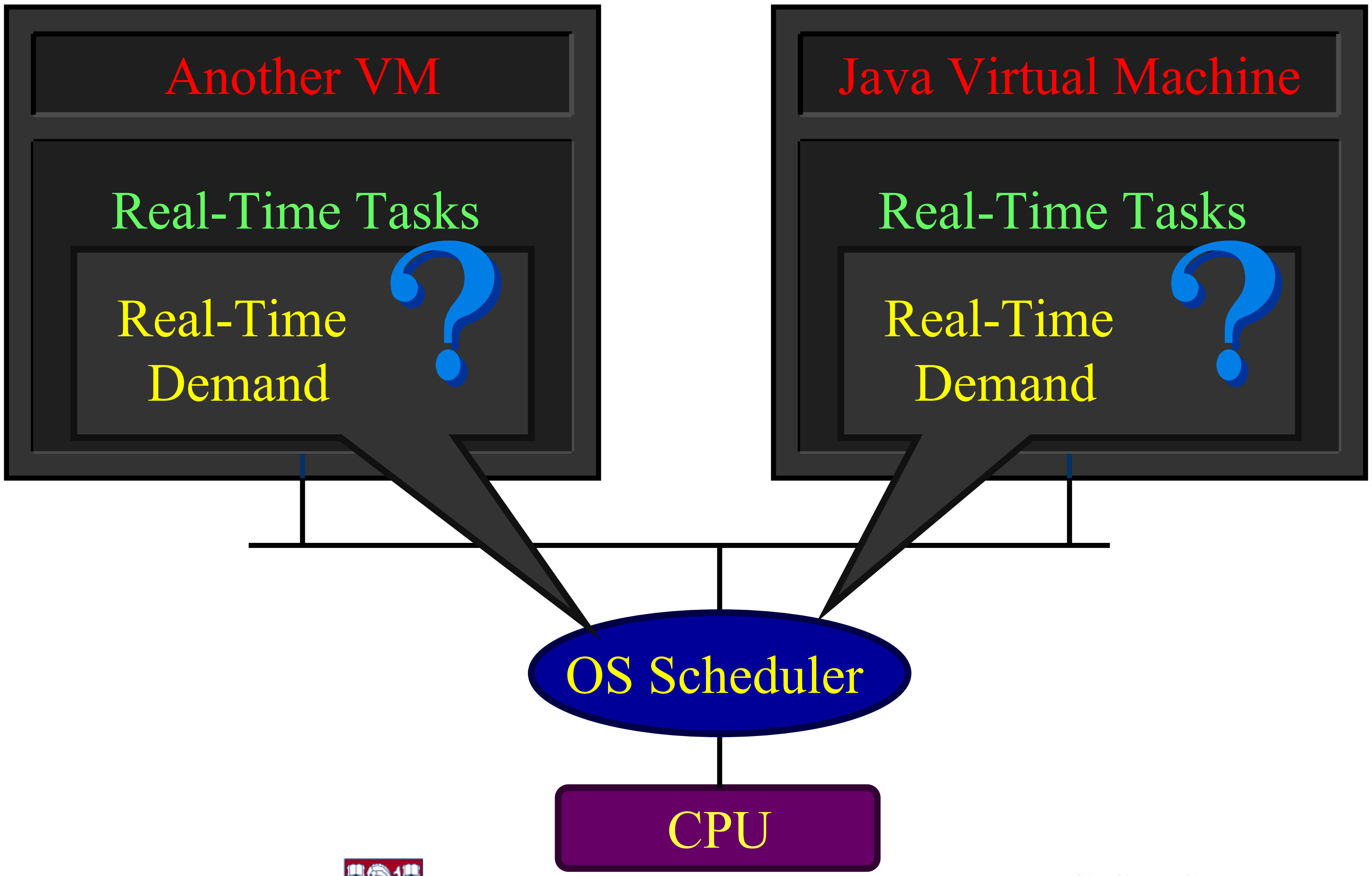
October 19, 2008

Workshop on Foundations and Applications of Component-based Design (WFCD'2008)

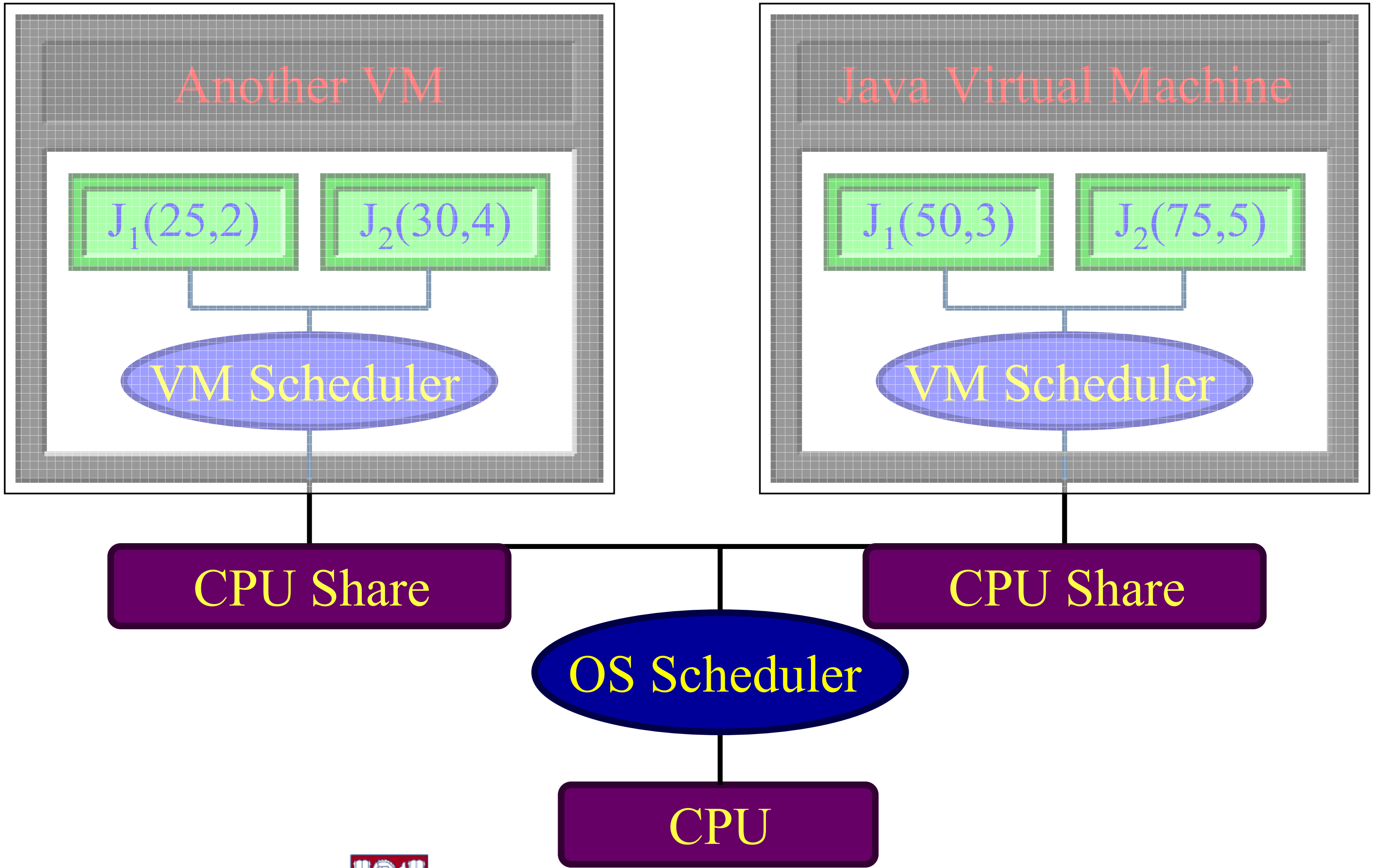
Motivating Example



Component Abstraction



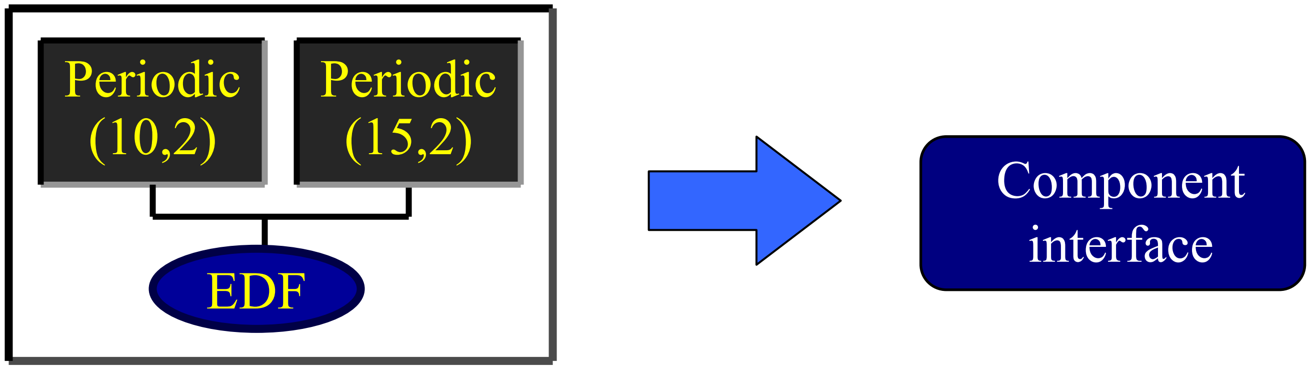
Component Abstraction



The CSA Problem Statement

Two Problems: Abstraction & Composition

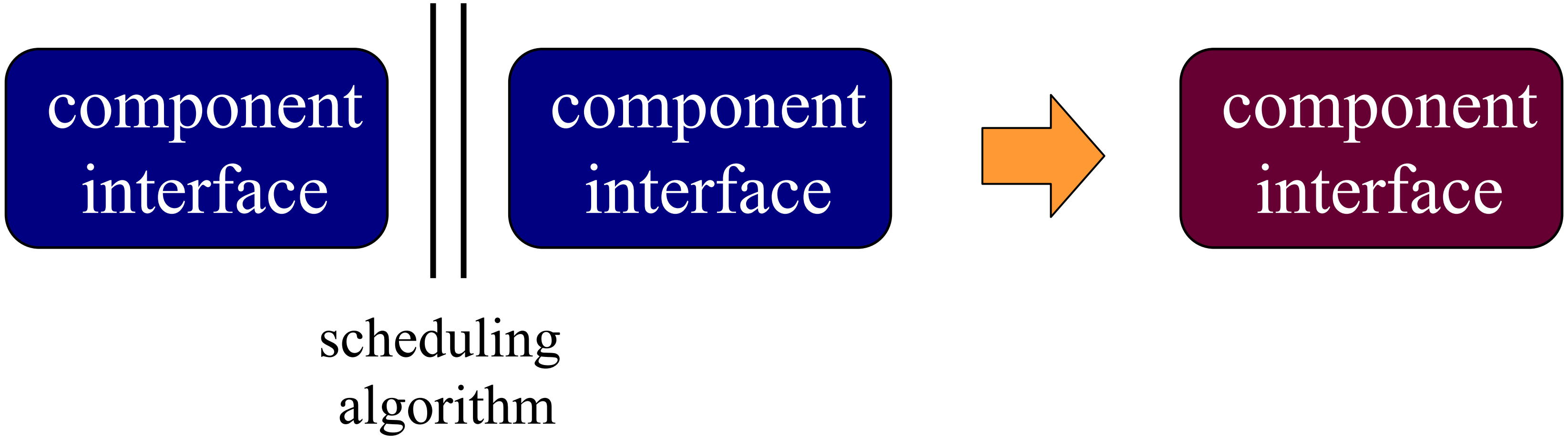
- *Abstraction Problem:* abstracts the real-time requirements of component (application) with *interface*



- Compute the minimum real-time requirements necessary for guaranteeing the schedulability of a component

Two Problems: Abstraction & Composition

- *Composition Problem:* composes component-level properties into system-level (or next-level component) properties



Compositionality

- Compositionality:
  - system-level properties can be established by composing independently analyzed component-level properties
- Compositional reasoning based on assume/guarantee paradigm
  - components are combined to form a system such that properties established at the component-level still hold at the system level.
- Compositional schedulability analysis using the demand/supply bounds
  - Establish the system-level timing properties by combining component-level timing properties through interfaces

Resource Satisfiability Analysis

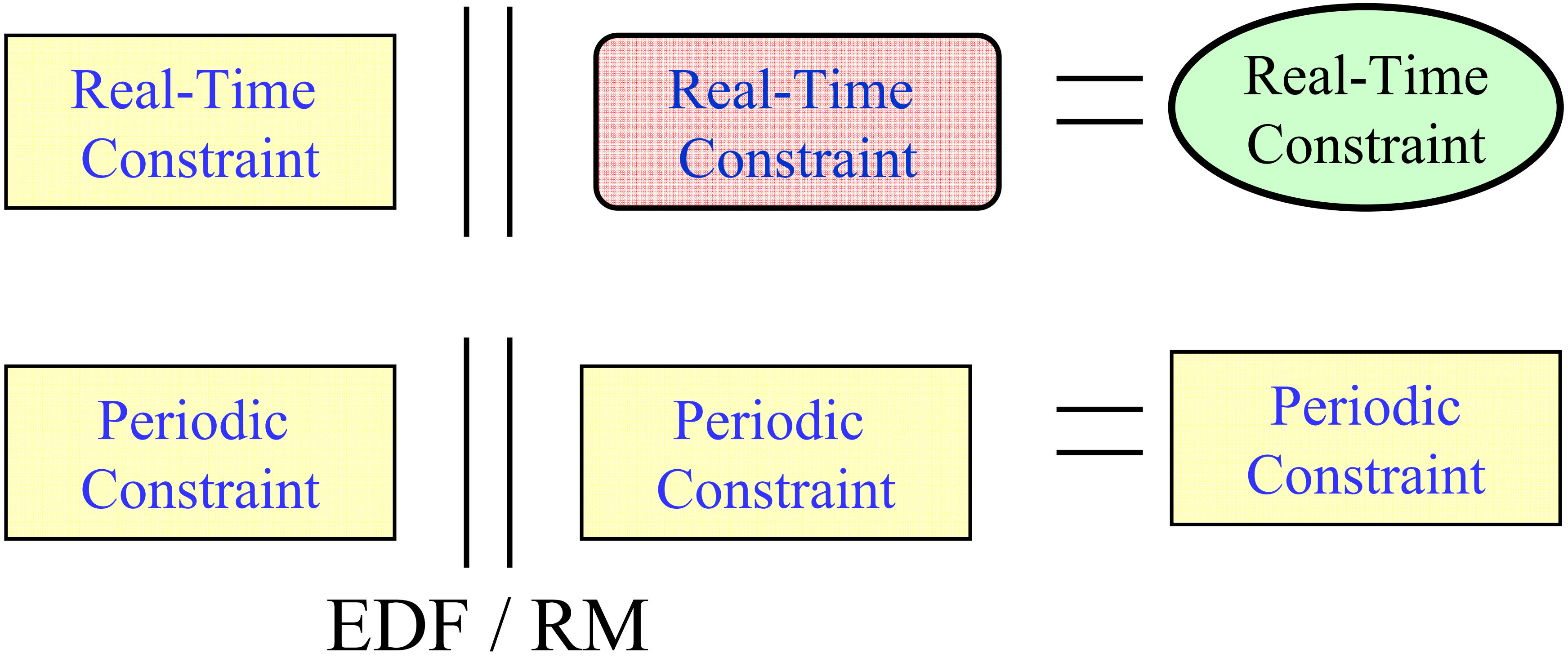
- Given a task set and a resource model, resource satisfiability analysis is to determine if, for every time,

resource demand, which a task set needs under a scheduling algorithm	≤	(minimum possible) resource supply

Resource Demand Models

Real-time demand composition

- Combine real-time requirements of multiple tasks into real-time requirement of a single task



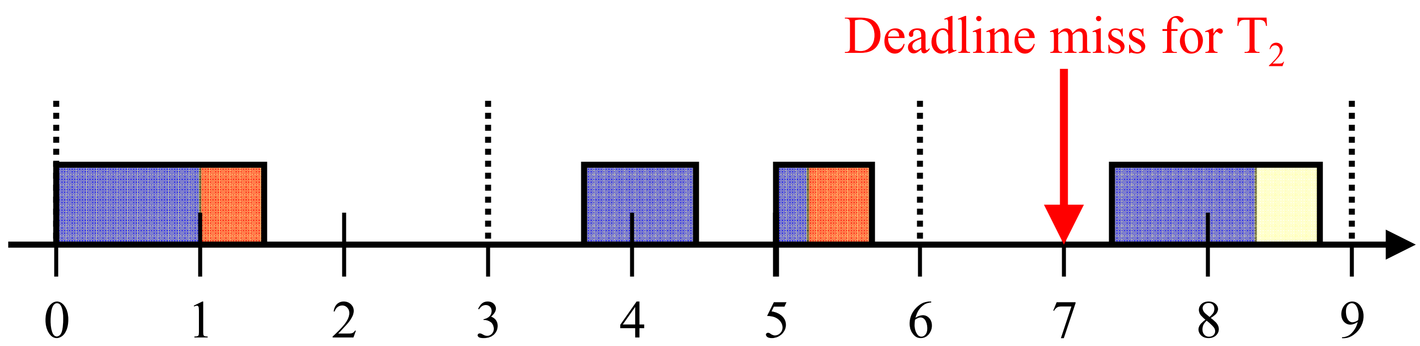
Non-composable periodic models?

- What are right abstraction levels for real-time components?
- (execution time, period)
- P1 = (p1,e1); e.g., (3,1)
- P2 = (p2,e2); e.g., (7,1)
- What is P1 || P2?
  - (LCM(p1,p2), e1*n1 + e2*n2); e.g., (21,10) where n1*p1 = n2*p2 = LCM(p1,p2)
- What is the problem?
  - beh(P1) || beh(P2) = beh(P1||P2)?
- Can we do
  - (P1 || P2) || P3 = P || P3, where P = P1 || P2?

Simple Observation (1)

- Given a task group G such that
  - Scheduling algorithm : EDF
  - A set of periodic tasks : { $T_1(3,1)$, $T_2(7,1)$ },

  model the timing requirements of the task group with a periodic task model
- G (3, 1.43) based on utilization does not work !!

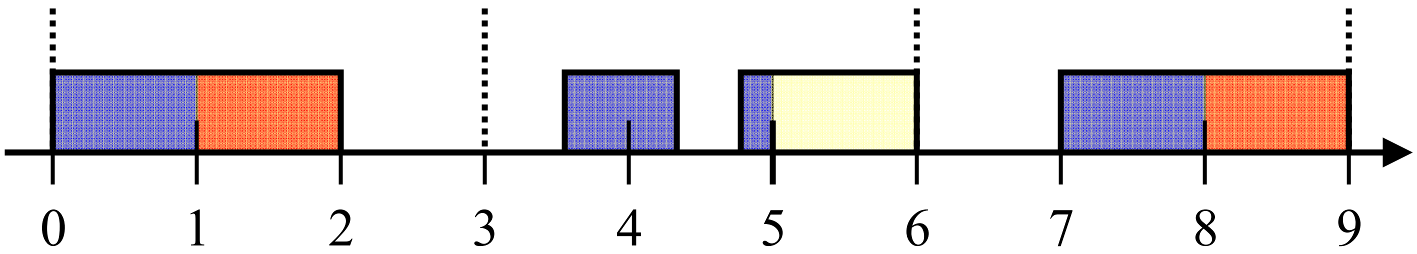


Simple Observation (2)

- Given a task group G such that
  - Scheduling algorithm : EDF
  - A set of periodic tasks : { $T_1(3,1)$, $T_2(7,1)$ },

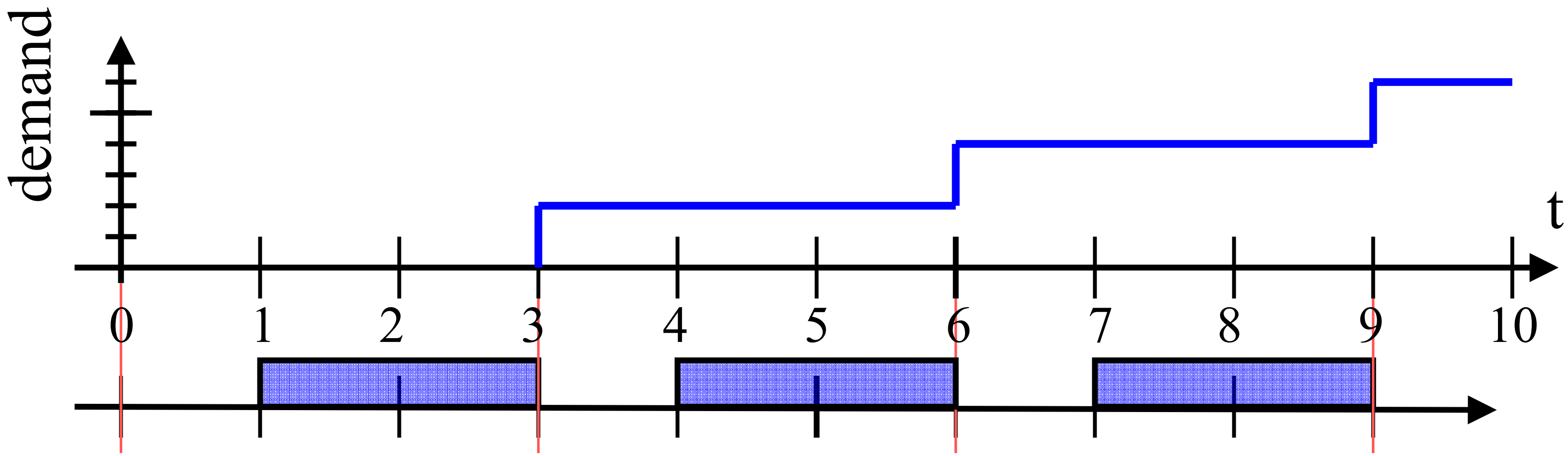
  model the timing requirements of the task group with a periodic task model

- G (3, 2.01) works !!



Resource Demand Bound

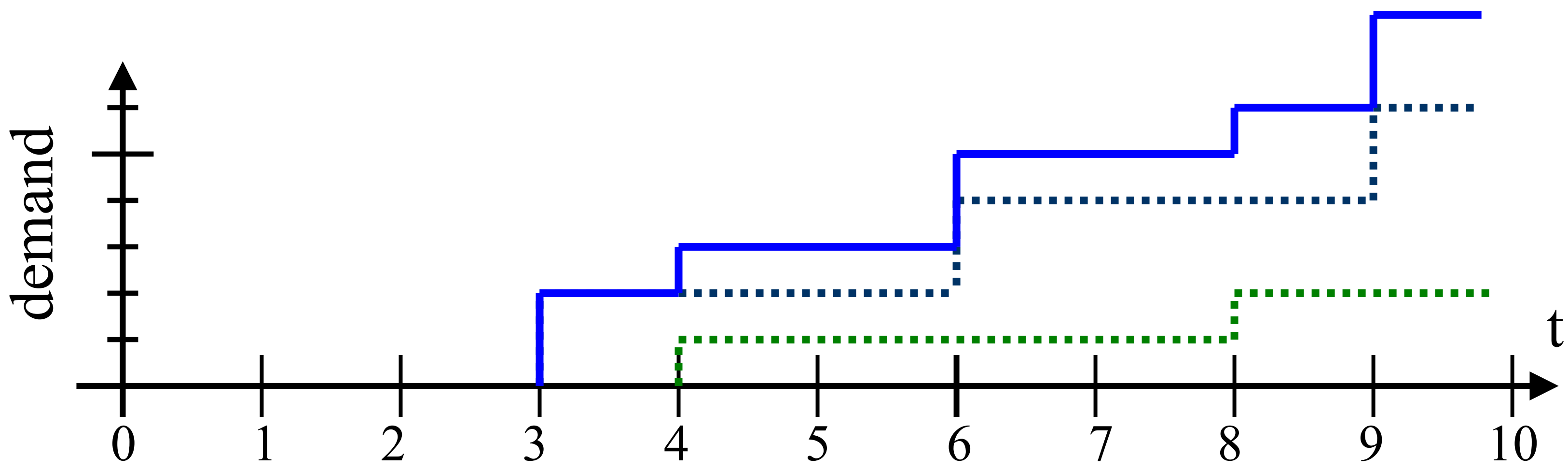
- Resource demand bound during an interval of length t
  - dbf(W,A,t) computes the maximum possible resource demand that W requires under algorithm A during a time interval of length t
- Periodic task model T(p,e) [Liu & Layland, '73]
  - characterizes the periodic behavior of resource demand with period p and execution time e
  - Ex: T(3,2)



Demand Bound - EDF

- For a periodic workload set *W* = {Ti(pi,ei)},
  - dbf (*W*,*A*,*t*) for EDF algorithm [Baruah et al.,'90]

$$\mathrm{dbf}\,(\mathrm{W},\mathrm{EDF},\mathrm{t}) = \sum_{T_i \in W} \left\lfloor \frac{t}{p_i} \right\rfloor \cdot e_i$$



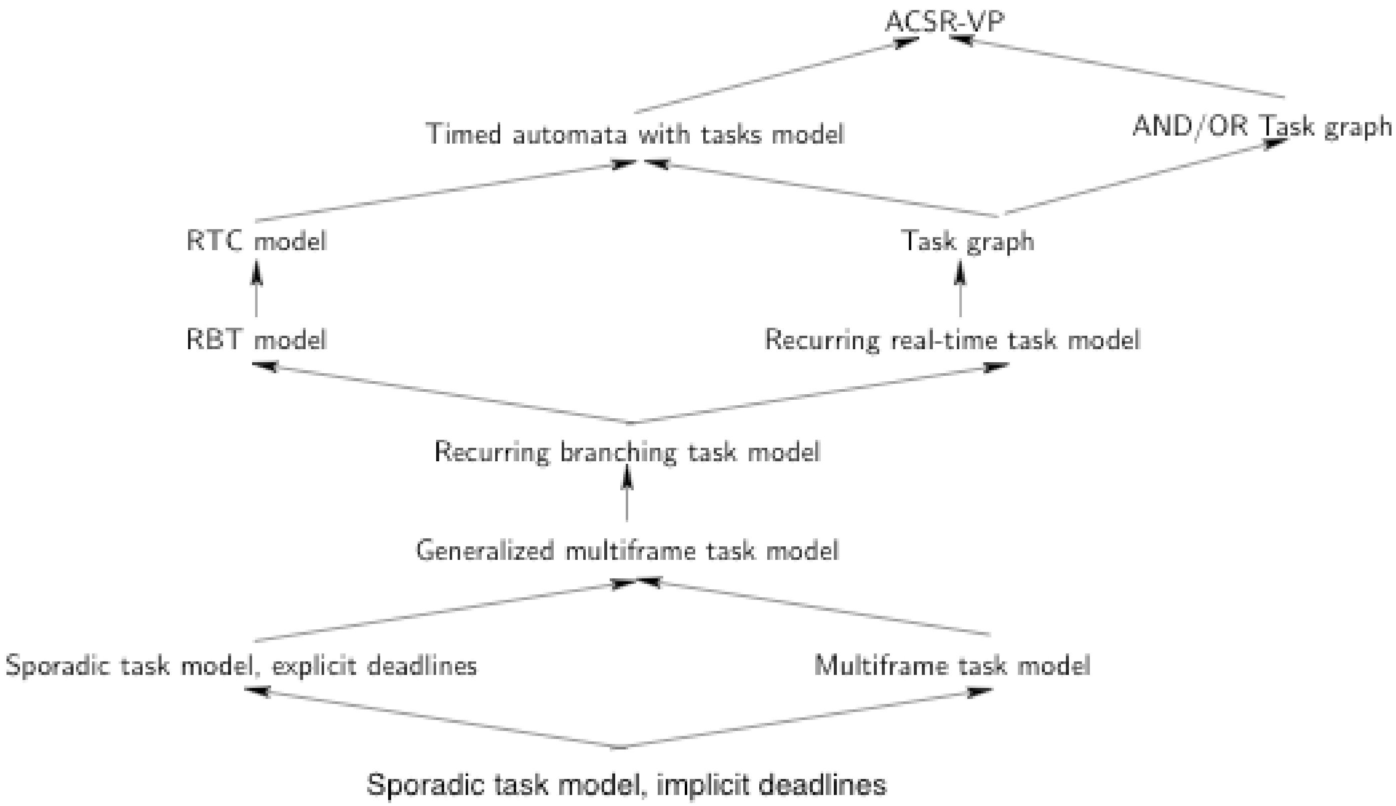
ACSR

$$T_1 = \{(cpu, 1)\} : \emptyset : T_1$$

$$T_2 = \{(cpu, 1)\} : \emptyset : T_2$$
$$+\emptyset : \{(cpu, 1)\} : \emptyset : T_2$$

$$T_3 = \{(cpu, 1)\} : \emptyset : \emptyset : T_3$$
$$+\emptyset : \{(cpu, 1)\} : \emptyset : T_3$$
$$+\emptyset : \emptyset : \{(cpu, 1)\} : T_3$$

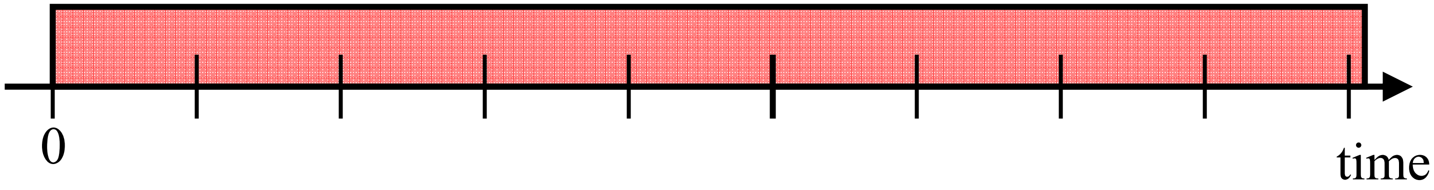
Task (resource demand) representations



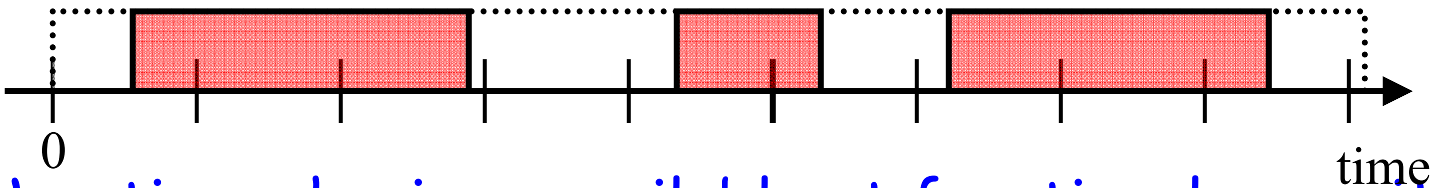
Resource Supply Models

Resource Modeling

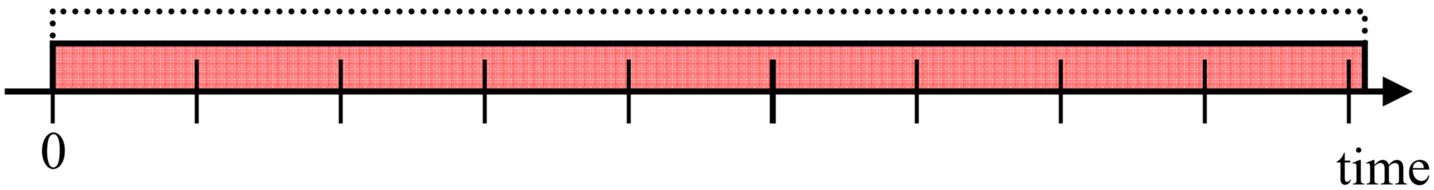
- Dedicated resource : always available at full capacity



- Shared resource : not a dedicated resource
  - Time-sharing : available at some times

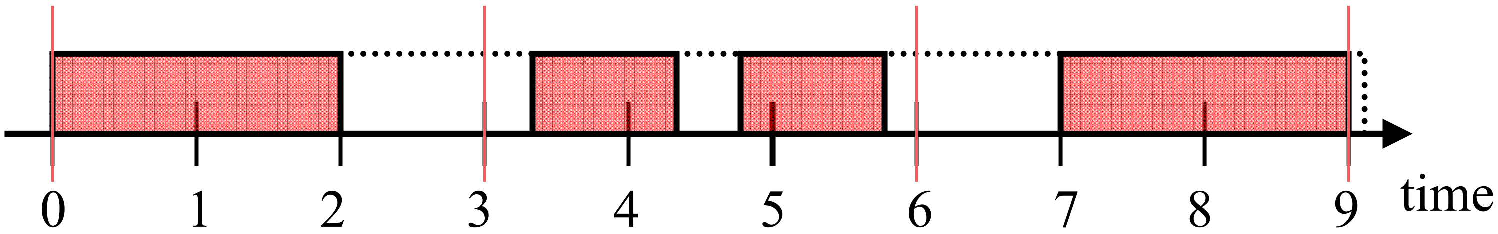


  - Non-time-sharing : available at fractional capacity



Resource Modeling

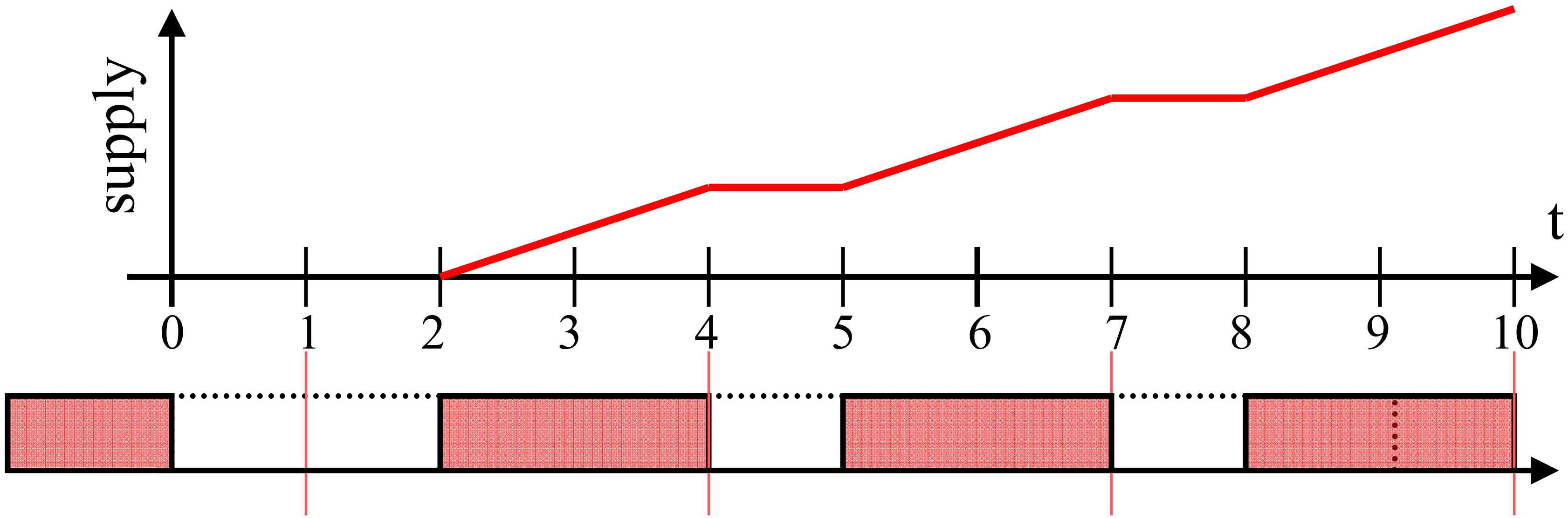
- Time-sharing resources
  - Bounded-delay resource model [Mok et al., '01] characterizes a time-sharing resource w.r.t. a non-time- sharing resource
  - Periodic resource model $\Gamma(\Pi,\Theta)$ [Shin & Lee, RTSS '03] characterizes periodic resource allocations
  - EDP model [Easwaran et all, RTSS 07]



Resource Supply Bound

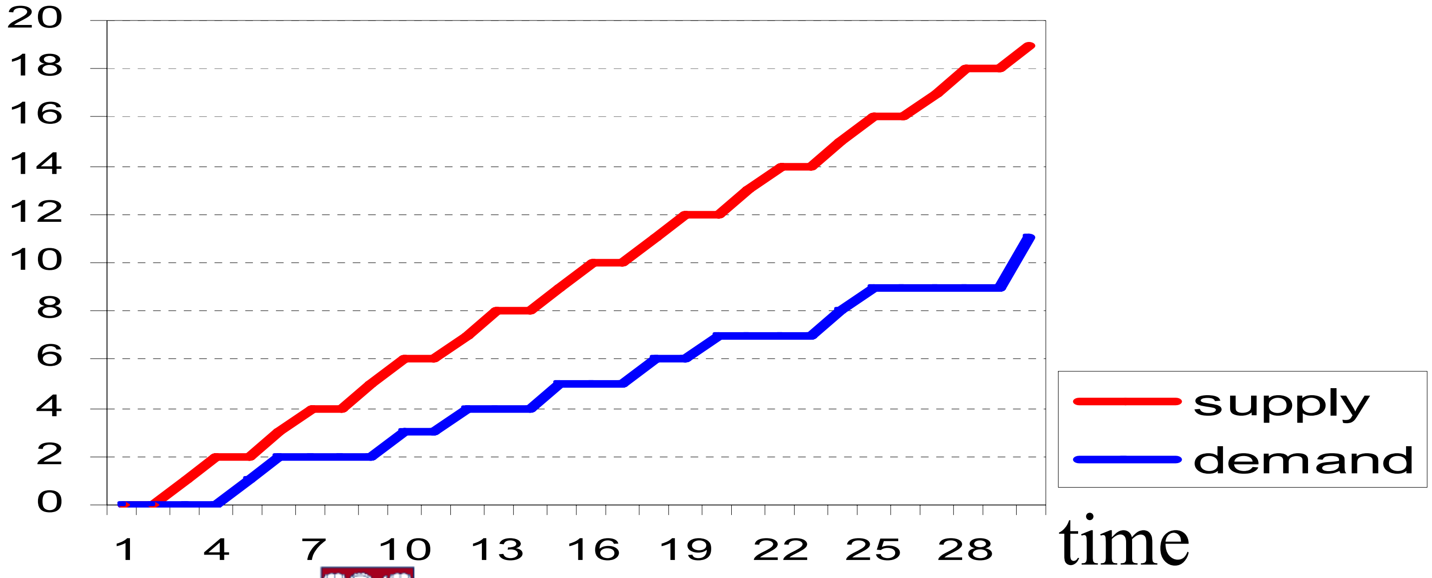
- Resource supply during an interval of length t
  - $sbf_R(t)$: the minimum possible resource supply by resource R over all intervals of length t

- For a single periodic resource model, i.e., Γ(3,2)
  - we can identify the worst-case resource allocation



Schedulability Condition - EDF

- A periodic workload set W is schedulable under a scheduling algorithm A over a periodic resource model $\Gamma(\Pi, \Theta)$ if and only if $\forall t > 0 \quad \mathrm{dbf}(W, EDF, t) \leq \mathrm{sbf}_\Gamma(t)$
  - A = EDF



EDP resource model based Interfaces

- Explicit Deadline Periodic resource
- Specification: $\Omega = (\Pi, \Theta, \Delta)$
  - Explicit deadline Δ
  - Θ resource units in Δ time units
  - Repeat supply every Π time units

EDP supply bound function (sbf_Ω)

- $sbf_\Omega(t)$

$$sbf_\Omega(\mathrm{t}) = y\Theta + \max\{0, t - (\Pi + \Delta - 2\Theta) - y\Pi\}$$

$$where \quad y = \left\lfloor \frac{t - (\Delta - \Theta)}{\Pi} \right\rfloor, \; t \geq \Delta - \Theta$$

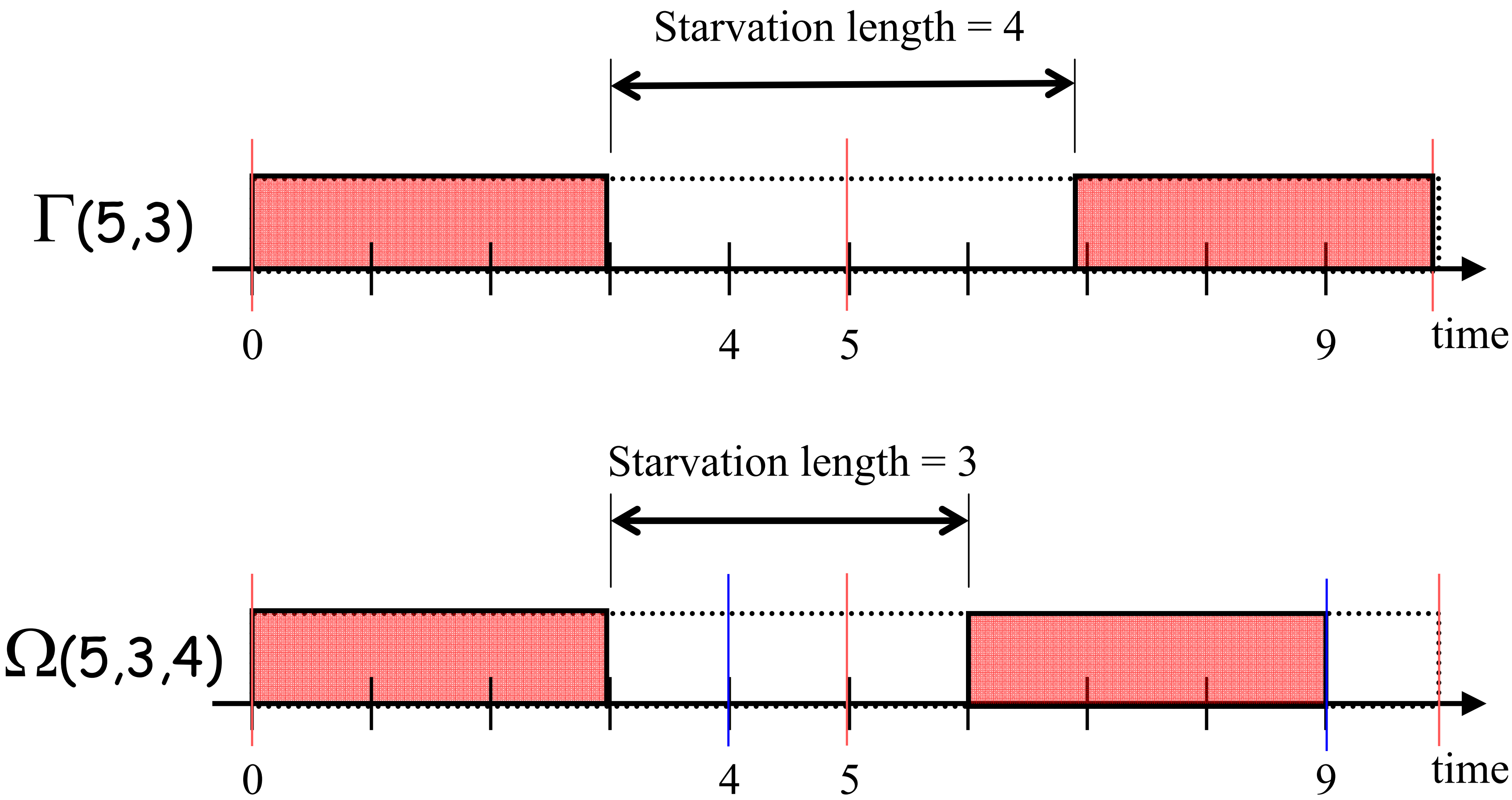
- $lsbf_\Omega(t)$

$$lsbf_\Omega(\mathrm{t}) = \frac{\Theta}{\Pi}(t - \underbrace{(\Pi + \Delta - 2\Theta)})$$

Bandwidth

Starvation length

Supply bound function (sbf_{Ω})



ACSR+ for supply partition specification

$$S_1 = \{(cpu, \_)\} : \emptyset : S_1$$

$$S_2 = \{(cpu, \_)\} : \emptyset : S_2 \oplus \emptyset : \{(cpu, -)\} : S_2$$

$$T_1 = \{(cpu, 1)\} : \emptyset : T_1$$

$$T_2 = \{(cpu, 1)\} : \emptyset : T_2$$
$$+\emptyset : \{(cpu, 1)\} : \emptyset : T_2$$

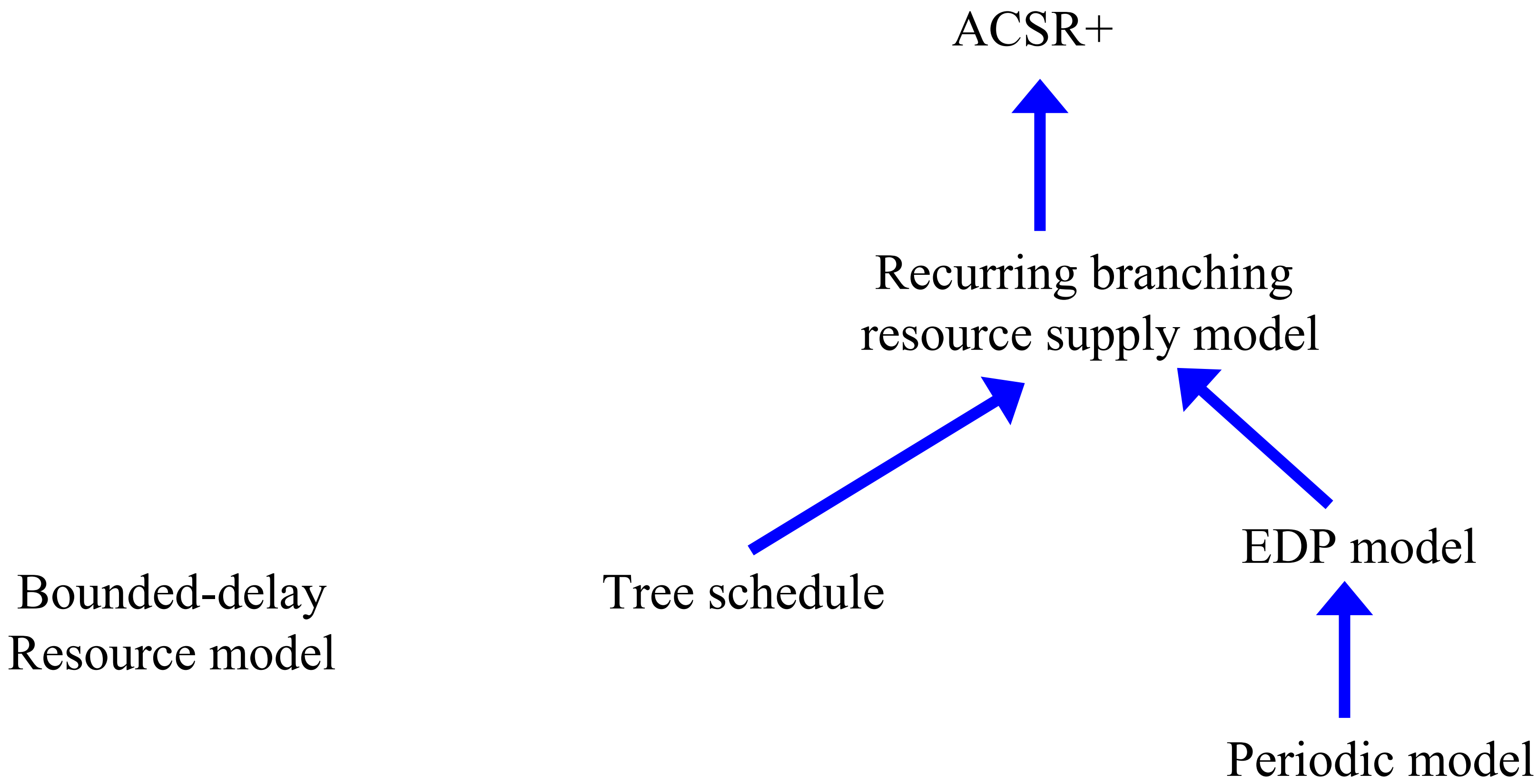
Notion of “schedulable under”

(1) T_1 is schedulable under S_1

(2) T_2 is schedulable under S_2

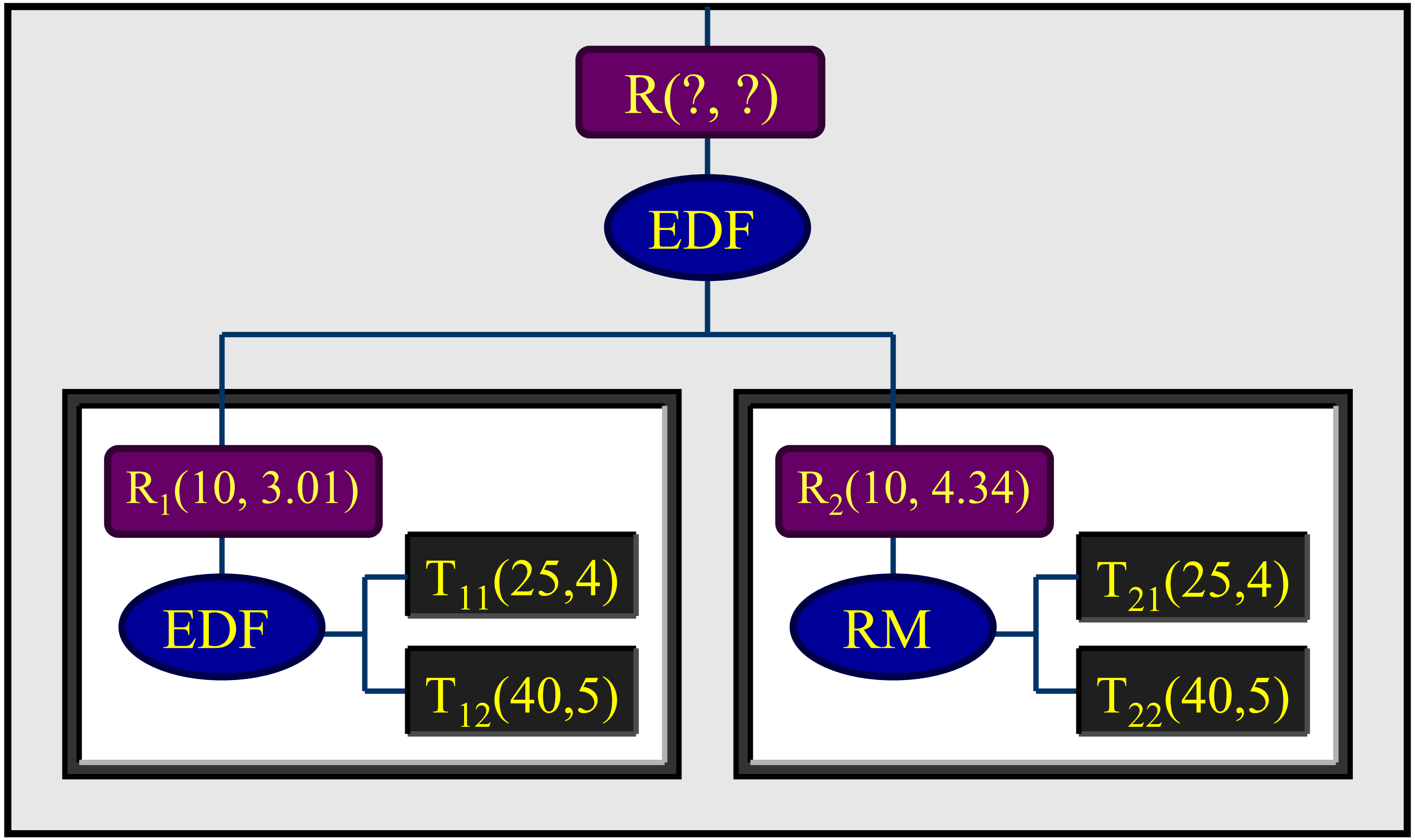
(3) T_1 is not schedulable under S_2

Resource Supply Models

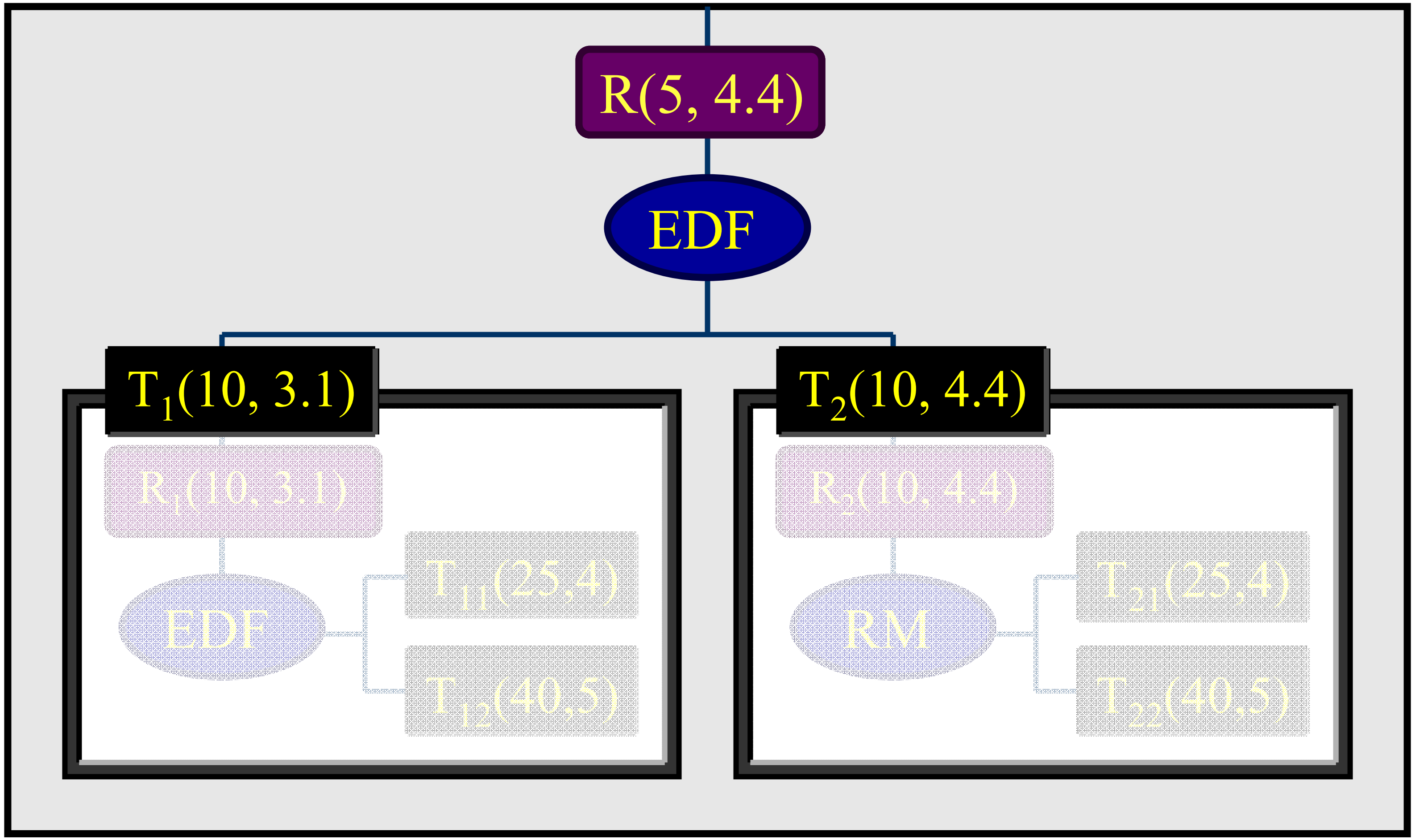


Compositional Schedulability Analysis

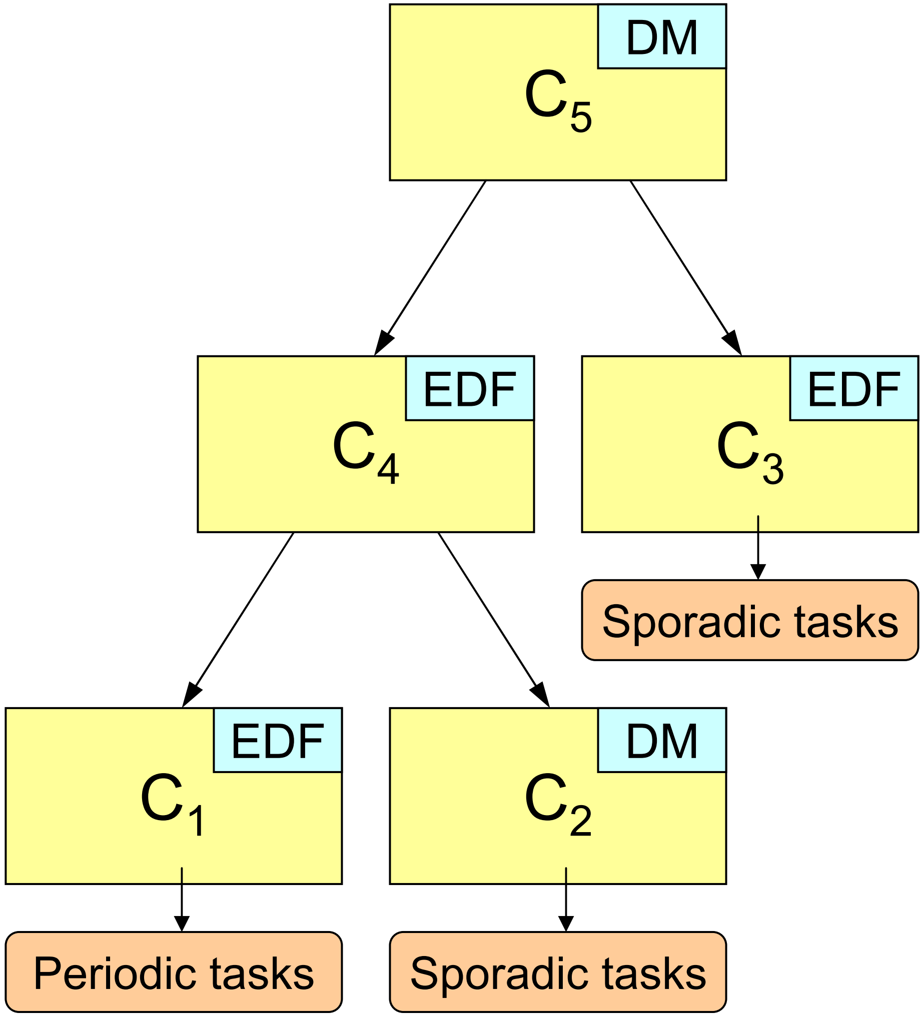
Component Abstraction



Compositional Real-Time Guarantees



Hierarchical Scheduling Framework



- Resource allocation from parent to child
- Notations
  - Leaf → C_1, C_2, C_3
  - Non-leaf → C_4, C_5
  - Root → C_5

ARINC 653 → Two-level hierarchical framework

Compositional Schedulability Analysis (CSA)

- Assume/Guarantee reasoning
  - Let C_R be a system configuration: Component C is running on resource R.
  - Let supply(C_R) be the residual supply of R after C; I.e., supply to the rest of the system.
  - If
    - C1 guarantees schedulability assuming demand(C1) <= supply(C2_R)
    - C2 guarantees schedulability assuming demand(C2) <= supply(C1_R)
  - Then,
    - C1 || C2 guarantees schedulability in (C1 || C2)_R

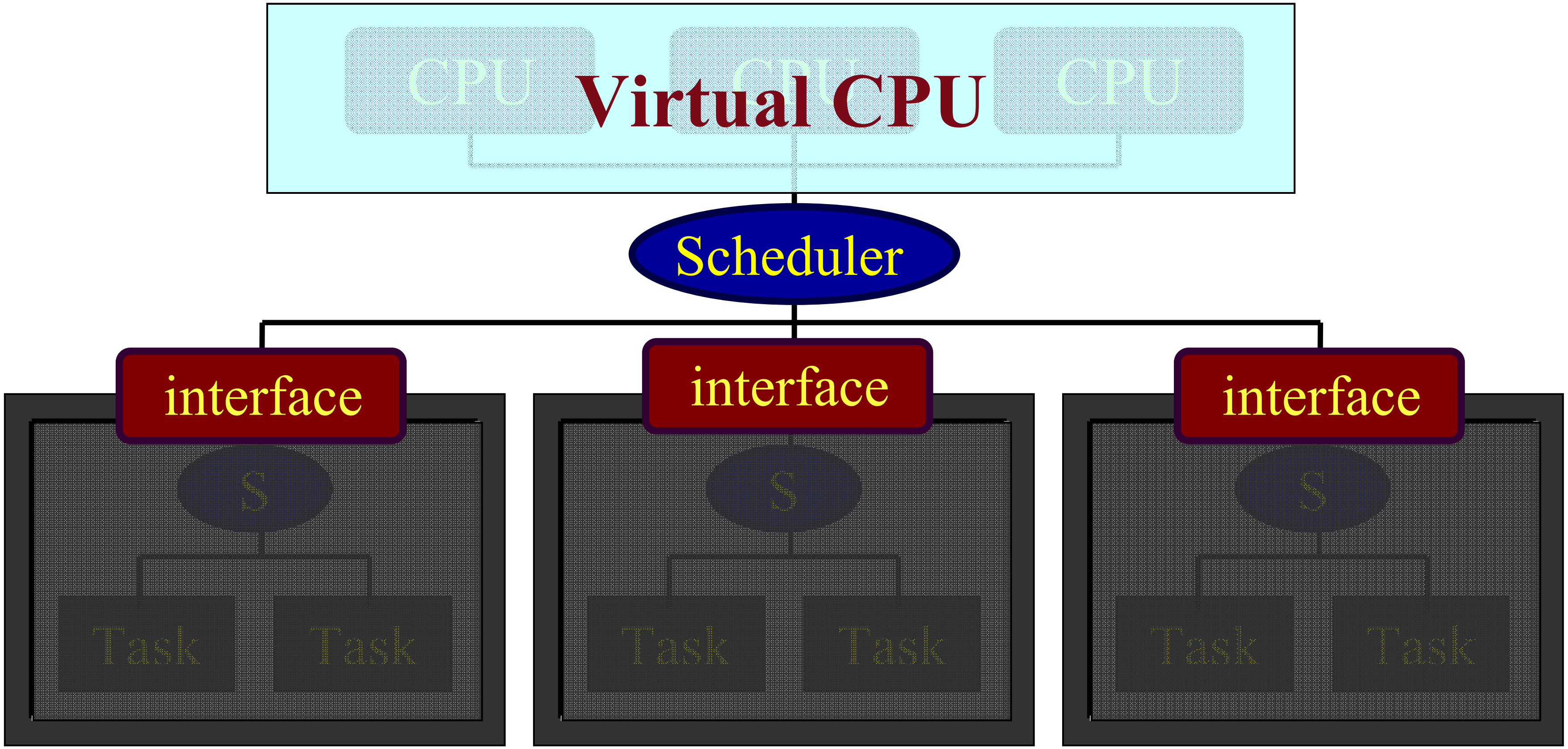
Questions on CSA

- Dbf/sbf bounds
  - Associativity
  - Minimum bounds on hierarchical scheduling
- ACSR/ACSR+
  - Non-deterministic supply alternatives
  - Definition and characterization of "schedulable under"
    - Given demand process T and supply partition S, when T schedulable with respect to S.
    - Relation to Linear Logic?

Hierarchical Scheduling Framework for Virtual Clustering of Multiprocessors

Multicore Processor Virtualization

1. Compositional analysis of hierarchical multiprocessor real-time systems, through component interfaces
2. Using virtualization to develop new component interface for multiprocessor platforms



Multiprocessor Embedded Systems

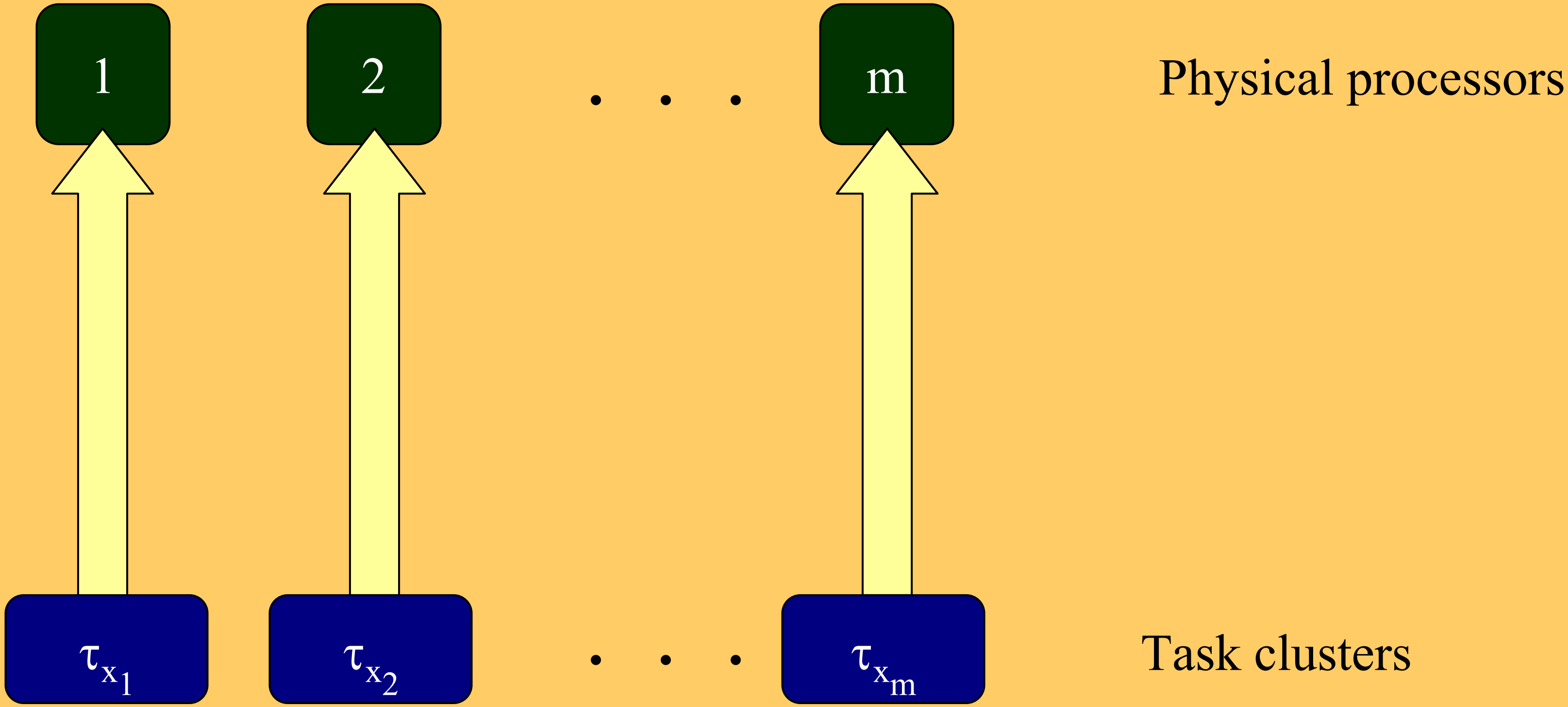
- Why consider multiprocessors
  - Better tradeoff between computational power and costs (energy,fabrication)
  - Ability to exploit inherent concurrency in embedded software
- Problem Statement
  - Constrained deadline sporadic task model

    $\tau = \{\tau_1, \ldots, \tau_n\}$, where each $\tau_i = (T_i, C_i, D_i)$ with $C_i <= D_i <= T_i$

    C_i units must be supplied non-concurrently
  - Identical, unit-capacity multiprocessor platform

    m processors
  - How can they be scheduled?

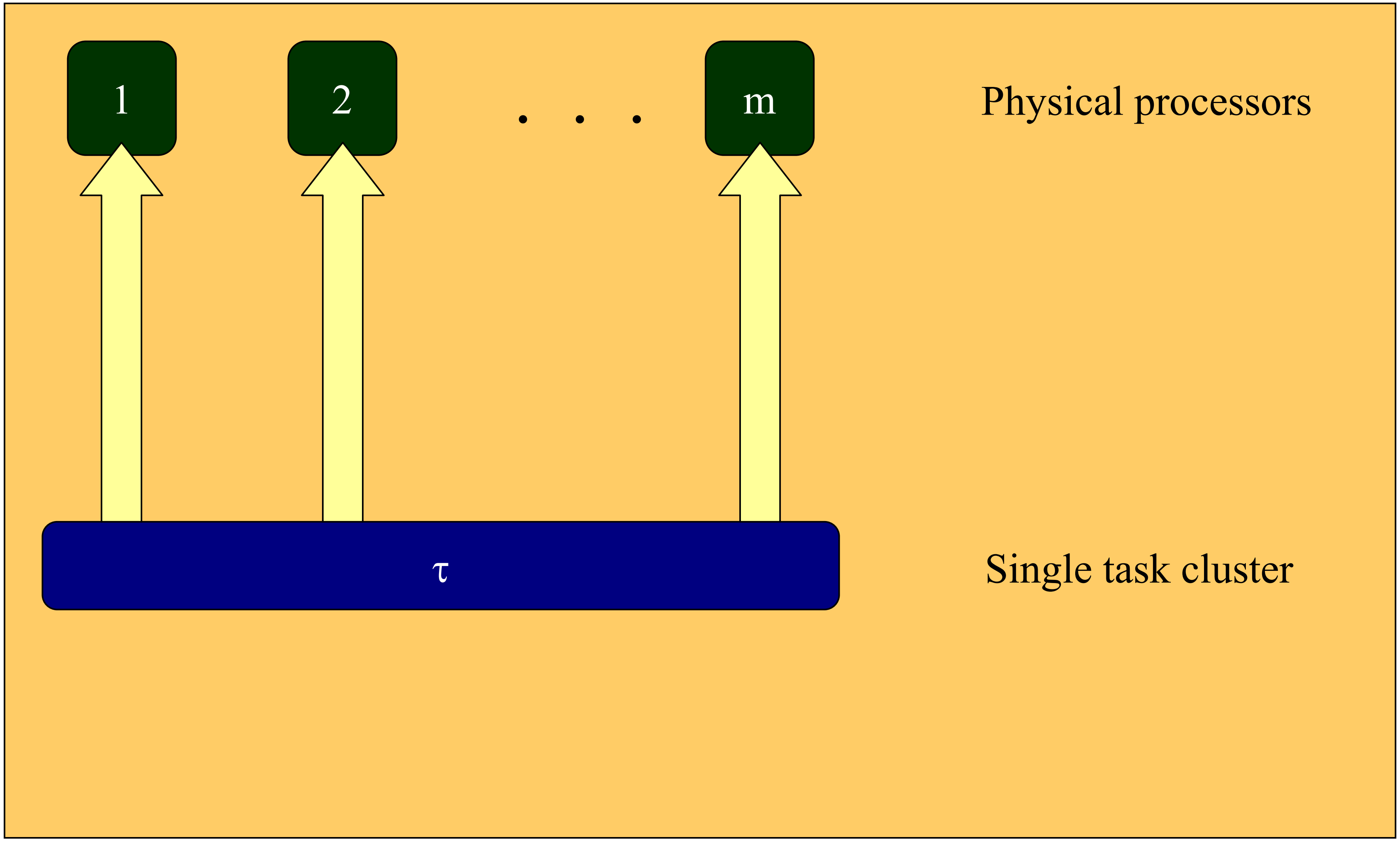
Partitioned Scheduling



$$\tau_{x_1} \cup \tau_{x_2} \ldots \cup \tau_{x_m} = \tau$$

$$\tau_{x_i} \cap \tau_{x_j} = \phi \text{ for all i and j}$$

Global Scheduling



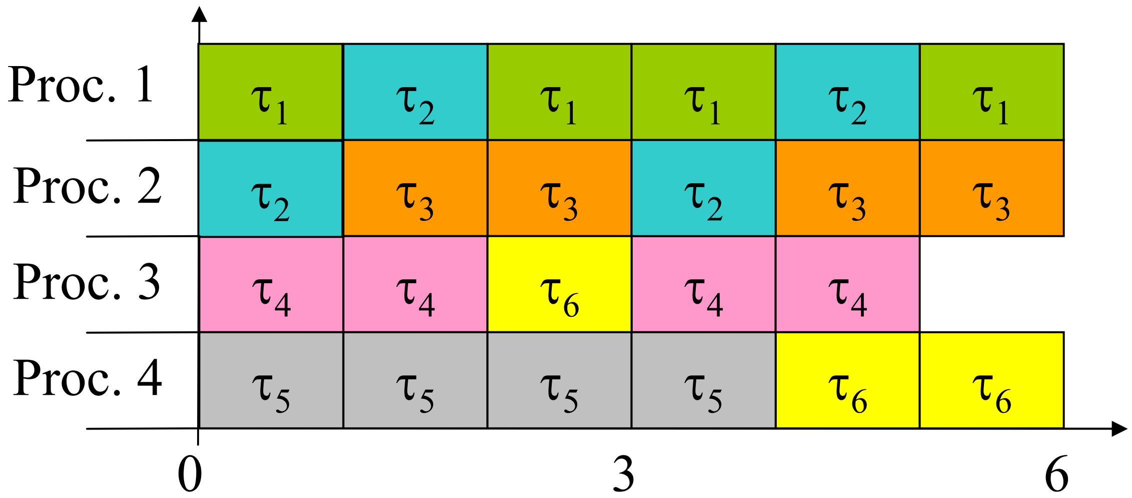
Motivation for Virtual Clustering

- Task set and number of processors

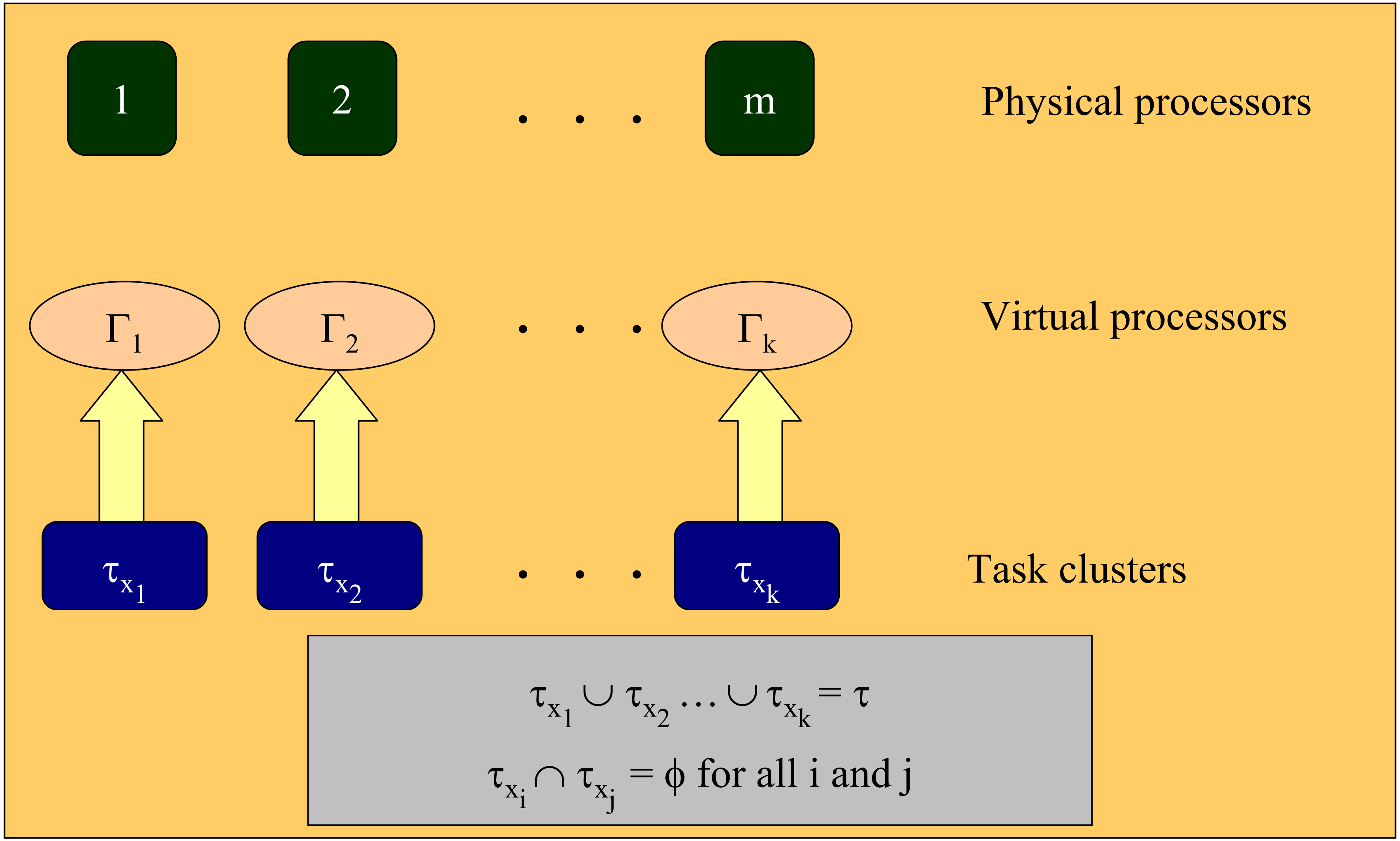
  $\tau_1=\tau_2=\tau_3=\tau_4=(3,2,3)$, $\tau_5=(6,4,6)$, and $\tau_6=(6,3,6)$, m=4
- Schedule under clustered scheduling

  τ_1, τ_2, τ_3 scheduled on processors 1 and 2

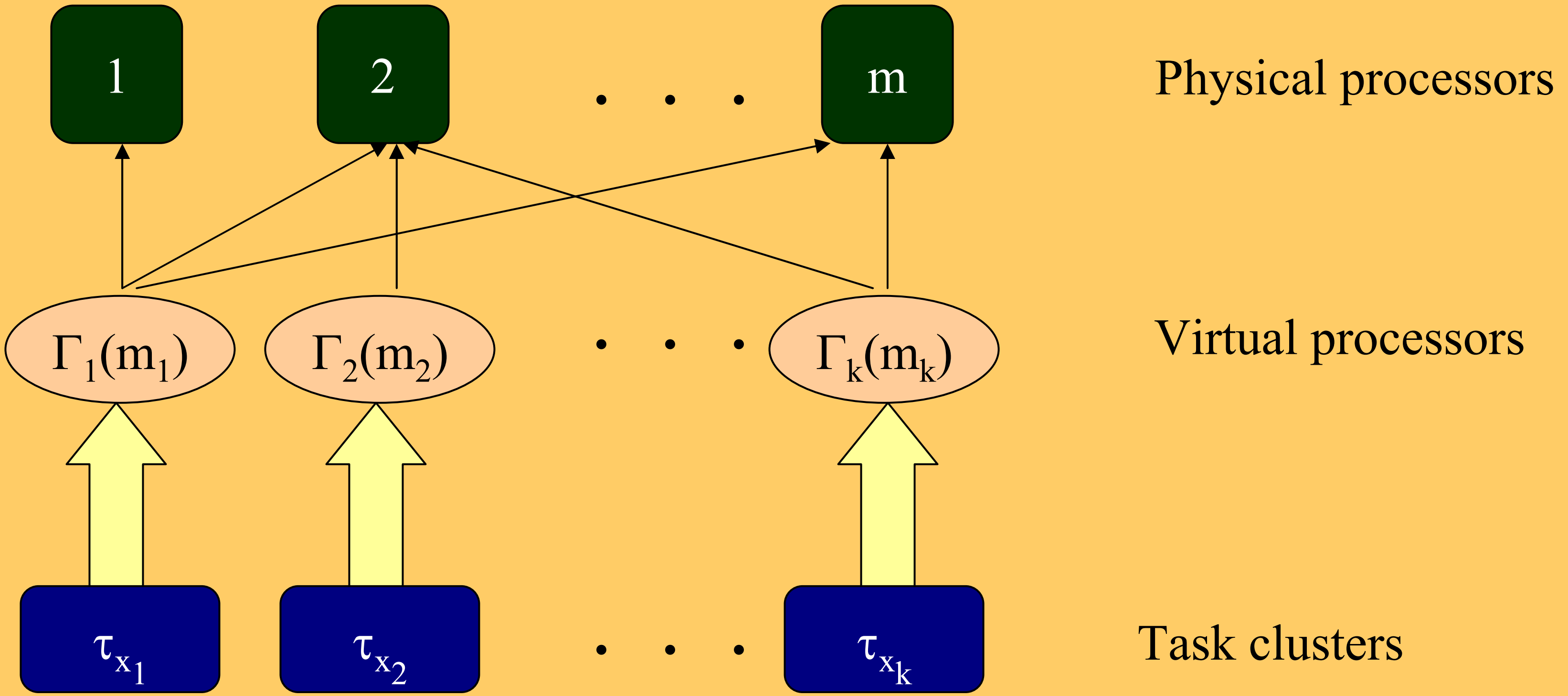
  τ_4, τ_5, τ_6 scheduled on processors 3 and 4



Virtual Clustering Interface

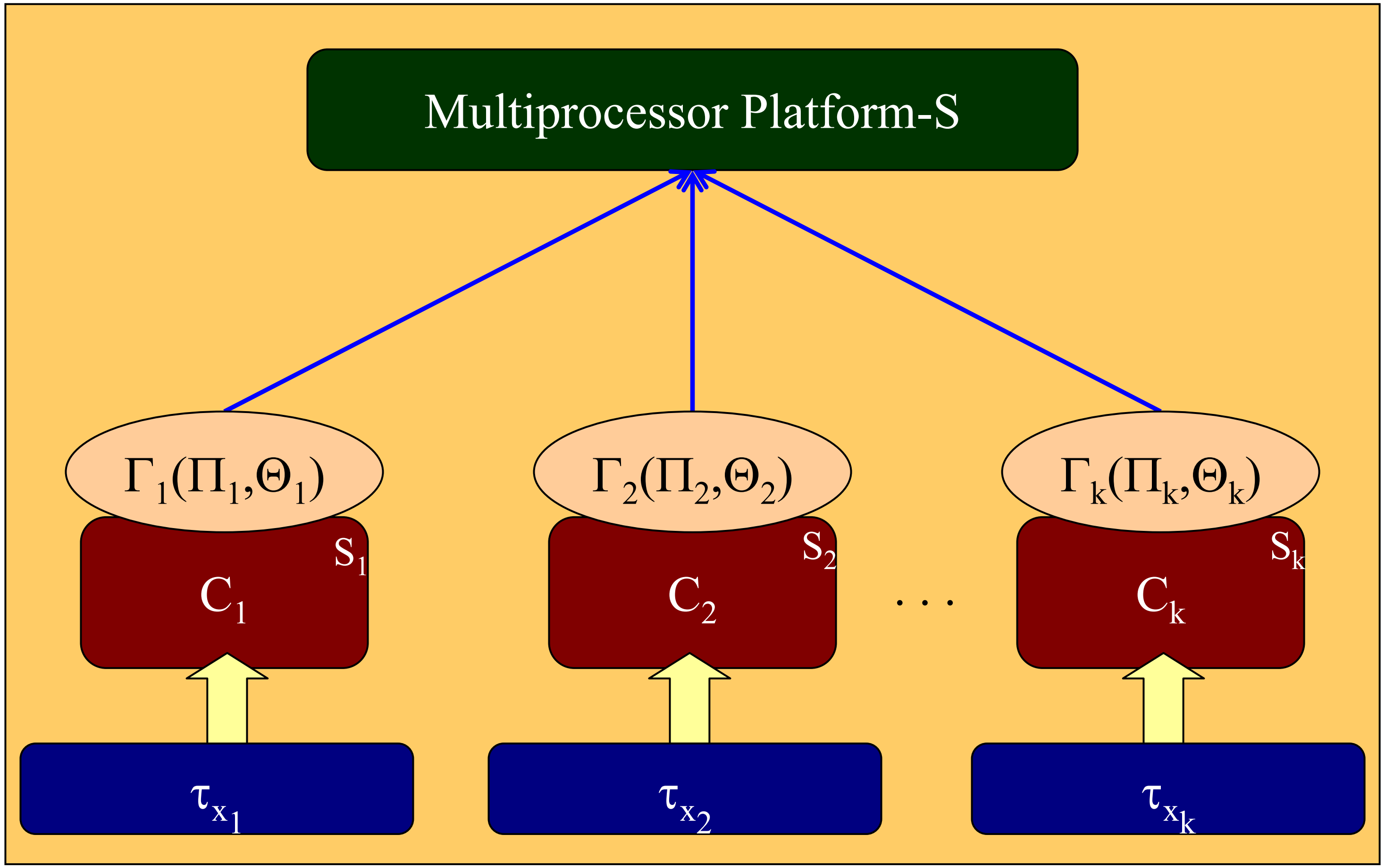


Virtual Clustering Interface

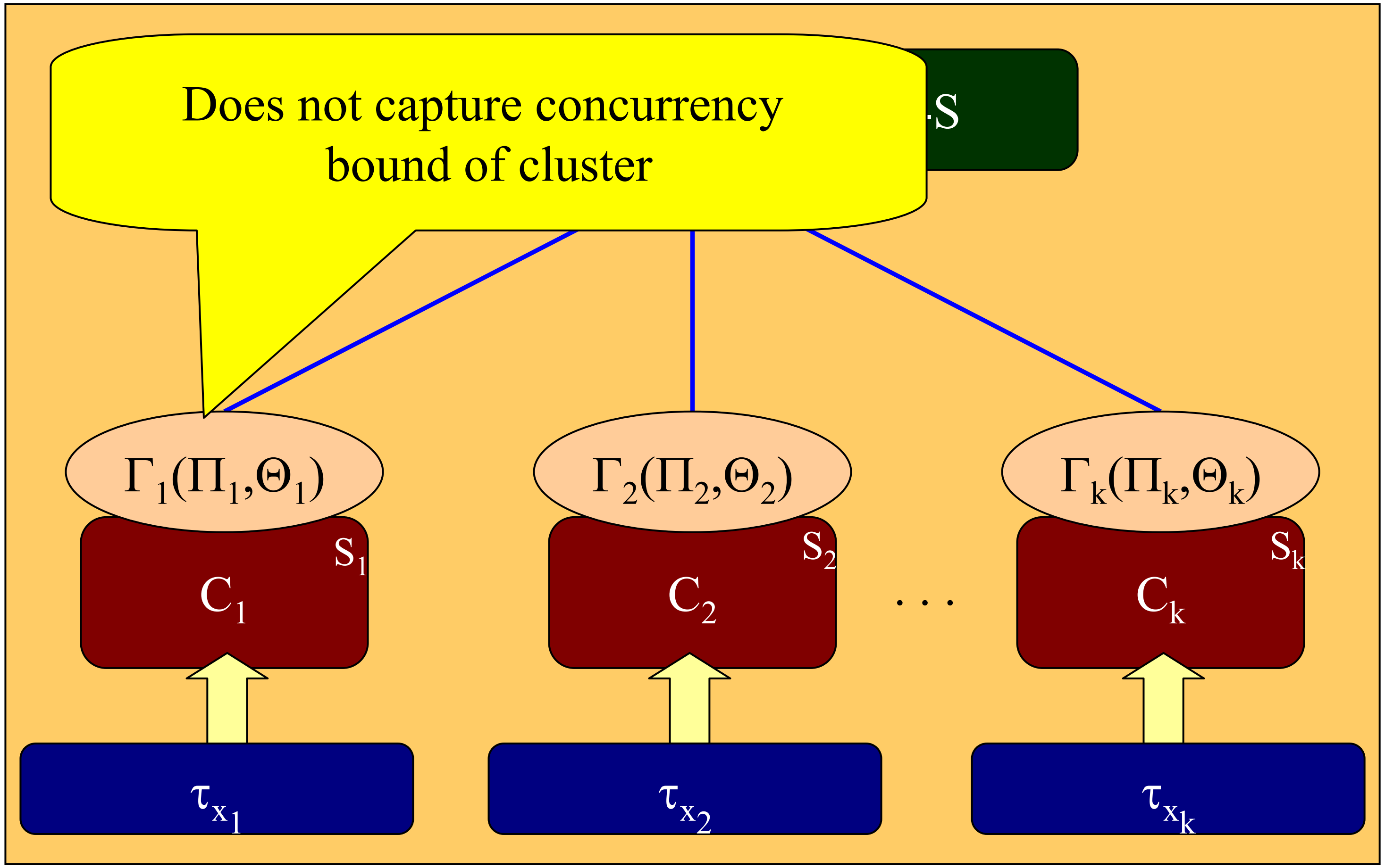


For each Γ_i, m_i ($\le m$) is maximum number of physical processors that can be assigned to Γ_i at any instant

Multiprocessor Periodic Resource (MPR) model



Multiprocessor Periodic Resource (MPR) model



Multiprocessor Periodic Resource (MPR) model

- $\Gamma = (\Pi, \Theta, m')$
  - Θ units of resource guaranteed in every Π units of time, with amount of concurrency at most m' in any time slot

- Why MPR model?
  - Periodicity enables transformation of resource model to tasks that can be used by various inter-cluster schedulers (schedulers at higher level)

Virtual Cluster based Interface

1. Split task set τ into clusters $\tau_{x_1}, \ldots, \tau_{x_k}$
   - We assume that clusters are given
2. Abstract cluster τ_{x_i} into MPR model Γ_i
   - Solution for global EDF intra-cluster interface
   - Present sufficient schedulability condition and minimize overhead of Γ_i
3. Transform each Γ_i into periodic tasks
   - Enables inter-cluster scheduler to schedule Γ_i
   - Preserves concurrency bound of Γ_i

Conclusions

- Interface framework for real-time system based on hierarchical schedulability analysis
  - Independent implementation of components
  - Interface-based component composition
  - Virtual clustering for multiprocessors
- Other issues
  - Task blocking due to synchronization
  - Context switch overheads
- Applications
  - ARINC 653
  - Automotive SAE J2056/Class C Vehicle Communication Requirements
  - Real-Time Virtual Machines (esp. multicore processors)

References

- Hierarchical Scheduling Framework for Virtual Clustering of Multiprocessors, Insik Shin, Arvind Easwaran, Insup Lee, ECRTS, Prague, Czech Republic, July 2-4, 2008 (Runner-up in the best paper award)
- Robust and Sustainable Schedulability Analysis of Embedded Software, Madhukar Anand and Insup Lee, LCTES, Tucson, AZ, Jun 12-13, 2008
- Interface Algebra for Analysis of Hierarchical Real-Time Systems, Easwaran et al, FIT 2008
- Compositional Feasibility Analysis for Conditional Task Models, Anand et al, ISORC 2008
- Compositional Analysis Framework using EDP Resource Models, Easwaran et al, RTSS 2007
- Incremental Schedulability Analysis of Hierarchical Real-Time Components, Easwaran et al., EMSOFT 2006.
- Compositional Real-Time Scheduling Framework, Shin & Lee, RTSS 2004
- Periodic Resource Model for Compositional Real-Time Guarantees, Shin & Lee, RTSS 2003

Related work

- Much work on hierarchical scheduling
  - Provide schedulability conditions that are needed for instantiation
  - Serves as the basis for abstraction
    - Shin and Lee, '03 '04, Easwaran et al., '06
- Real-time interface frameworks
  - Henzinger and Matic, '06
  - Wandeler and Thiele, '06

Compositional Analysis (Dependency using Task Parameters)

- [AlPe04] Luis Almeida and Paulo Pedreiras. "Scheduling within temporal partitions: Response-time analysis and server design". In EMSOFT 95-103, 2004.
- [DaBu05] R. I. Davis and A. Burns, "Hierarchical fixed priority preemptive scheduling". In RTSS 2005.
- [MaHe05] S. Matic and T. A. Henzinger, "Trading end-to-end latency for composability". In RTSS 2005.
- [DaBu06] R. I. Davis and A. Burns, "Resource sharing in hierarchical fixed priority pre-emptive systems". In RTSS 2006.

Compositional Analysis (Dependency using Protocols)

- [DaBu06] R. I. Davis and A. Burns, “Resource sharing in hierarchical fixed priority pre-emptive systems”. In RTSS 2006.
- [BSN07] M. Behnam, I. Shin, T., Nolte, and M. Nolin. "SIRAP: a synchronization protocol for hierarchical resource sharingin real-time open systems”. In EMSOFT 2007.
- [DaBu06] N. Fisher, M. Bertgona, S. Baruah. “The Design of an EDF-Scheduled Resource-Sharing Open Environment”. In RTSS 2007.

Compositional Analysis (Conditional Task Models)

- [AESL08] Madhukar Anand, Arvind Easwaran, Sebastian Fischmeister, and Insup Lee. "Compositional analysis for conditional task models". ISORC, 2008.

Incremental and Compositional Analysis

- [WaTh05] Ernesto Wandeler and Lothar Thiele. “Real-time interface for interface-based design of real-time systems with fixed priority scheduling”. In EMSOFT, pages 80-89, 2005.
- [TWS06] L. Thiele, E. Wandeler, and N. Stoimenov, “Real-time interfaces for composing real-time systems.” In EMSOFT 2006.
- [WaTh06] E. Wandeler and L. Thiele, “Interface-based design of real time systems with hierarchical scheduling.” In RTAS 2006.
- [HeMa06] T. A. Henzinger and S. Matic, “An interface algebra for real time components.” In RTAS 2006.
- [ESSL06] A. Easwaran, I. Shin, O. Sokolsky, and I. Lee. ”Incremental Schedulability Analysis of Hierarchical Real-Time Components”. EMSOFT 2006.

Two-level hierarchical scheduling

- [DeLi97] Z. Deng and Jane Liu. "Scheduling real-time applications in an open environment". In RTSS, pages 308-319, 1997.
- [KuLi99] T.-W. Kuo and C. Li, "A fixed-priority-driven open environment for real-time applications." In RTSS 1999.
- [LiBa00] G. Lipari and S. Baruah. "Efficient scheduling of real-time multi-task applications in dynamic systems". In RTAS 2000.
- [LCB00] G. Lipari, J. Carpenter and S. Baruah, "A framework for achieving inter-application isolation in multiprogrammed hard-real-time environments." In RTSS 2000.

Bounded-delay Resource Models

- MFC01] A. Mok, X. Feng, and D. Chen, "Resource partition for real time systems." In RTAS 2001.
- [FeMo02] X. Feng and A. Mok, "A model of hierarchical real-time virtual resources." In RTSS 2002.
- [ShLe04] I. Shin and I. Lee. "Compositional real-time scheduling framework". In RTSS 2004.

Periodic Resource Models

- [SRL02] S. Saewong, R. Rajkumar, J. Lehoczky, and M. Klein, "Analysis of hierarchical fixed-priority scheduling." In ECRTS 2002.
- [LiBi03] G. Lipari and E. Bini, "Resource partitioning among real time applications." In ECRTS 2003.
- [ShLe03] I. Shin and I. Lee. "Periodic resource model for compositional real-time guarantees". In RTSS 2003.
- [ZhBu07] F. Zhang, A. Burns. "Analysis of Hierarchical EDF Pre-emptive Scheduling". In RTSS 2007.
- [EAL07] A. Easwaran, M. Anand, I. Lee. "Compositional Analysis Framework Using EDP Resource Models". In RTSS 2007.

Virtual Clustering

- [ACD06] James Anderson, John Calandrino, and UmaMaheswari Devi. “Real-time scheduling on multicore platforms”. In RTAS, 179-190, 2006.
- [LeAn08] Hennadiy Leontyev and James H. Anderson. “A hierarchical multiprocessor bandwidth reservation scheme with timing guarantees”. In ECRTS, pages 191- 200, 2008. (soft real-time)
- [MoRa99] Mark Moir and Srikanth Ramamurthy. “Pfair scheduling of fixed and migrating periodic tasks on multiple resources”. In RTSS, 1999.
- [HoAn01] Philip Holman and James H. Anderson. “Guaranteeing pfair supertasks by reweighting”. In RTSS, pages 203-212, 2001.
- [SEL08] Insik Shin, Arvind Easwaran, and Insup Lee. “Hierarchical Scheduling Framework for Virtual Clustering of Multiprocessors”. In ECRTS 2008.

Partitioned Scheduling

- [OhBa98] Dong-Ik Oh and Theodore Baker. “Utilization bounds for n-processor rate monotone scheduling with static processor assignment”. Real-Time Systems Journal, 15(2):183{192, 1998.
- [LDG01] J. M. L□opez, J. L. D□□az, and D. F. Garc□□a. “Minimum and maximum utilization bounds for multiprocessor RM scheduling”. In ECRTS, pages 67-75, 2001.
- [BaFi06] Sanjoy Baruah and Nathan Fisher. “The partitioned multiprocessor scheduling of deadline-constrained sporadic task systems”. IEEE Transactions on Computers, 55(7), pages 918-923, 2006.
- [FBB06] Nathan Fisher, Sanjoy Baruah, and Theodore P. Baker. “The partitioned scheduling of sporadic tasks according to static-priorities”. In ECRTS, pages 118-127, 2006.

Global Scheduling

- [ABJ01] Bjorn Andersson, Sanjoy Baruah, and Jan Jonsson. “Static-priority scheduling on multiprocessors”. In RTSS, pages 193-202, 2001.
- [Baker05] Theodore Baker. “An analysis of EDF schedulability on a multiprocessor”. IEEE Transactions on Parallel Distributed Systems, 16(8), pages 760-768, 2005.
- [Baker06] Theodore Baker. “An analysis of □xed-priority schedulability on a multiprocessor”. Real-Time Systems Journal, 32(1-2), pages 49-71, 2006.
- [Baker03] Theodore P. Baker. “Multiprocessor edf and deadline monotonic schedulability analysis”. In RTSS, page 120, 2003.
- [BCPV96] S. Baruah, N. K. Cohen, C. G. Plaxton, and D. A. Varvel. “Proportionate progress: a notion of fairness in resource allocation”. Algorithmica, 15(6), pages 600-625, 1996.
- [Baruah04] Sanjoy Baruah. “Optimal utilization bounds for the □fixed-priority scheduling of periodic task systems on identical multiprocessors”. IEEE Transactions on Computers, 53(6), pages 781-784, 2004.

Global Scheduling

- [Baruah07] Sanjoy Baruah. “Techniques for multiprocessor global schedulability analysis”. In RTSS, 2007.
- [BaBa08] Sanjoy K. Baruah and Theodore Baker. “Global EDF schedulability analysis of arbitrary sporadic task systems”. In ECRTS, pages 3-12, 2008.
- [BaBa08] Sanjoy K. Baruah and Theodore Baker. “Schedulability analysis of global EDF”. Real-Time Systems Journal, 38(3), pages 223-235, 2008.
- [BaFi07] Sanjoy K. Baruah and Nathan Fisher. “Global deadline-monotonic scheduling of arbitrary-deadline sporadic task systems”. In OPODIS, pages 204-216, 2007.
- [BeCi07] Marko Bertogna and Michele Cirinei. “Response-time analysis for globally scheduled symmetric multiprocessor platforms”. In RTSS, 2007.
- [BCL05] Marko Bertogna, Michele Cirinei, and Giuseppe Lipari. “Improved schedulability analysis of EDF on multiprocessor platforms”. In ECRTS, pages 209-218, 2005.
- [CBJ06] Hyeonjoong Cho, Binoy Ravindran, and E. Douglas Jensen. “An optimal realtime scheduling algorithm for multiprocessors”. In RTSS, pages 101-110, 2006.

Global Scheduling

- [CLA02] S Cho, S-K Lee, S Ahn, and K-J Lin. "Effi□cient real-time scheduling algorithms for multiprocessor systems". IEICE Transactions on Communications, E85-
- B(12), pages 2859-2867, 2002.
- [CiBa07] Michele Cirinei and Theodore P. Baker. "EDZL scheduling analysis". In ECRTS, pages 9-18, 2007.
- [FKY08] Kenji Funaoka, Shinpei Kato, and Nobuyuki Yamasaki. "Work-conserving optimal real-time scheduling on multiprocessors". In ECRTS, pages 13-22, 2008.
- [GFB03] Joel Goossens, Shelby Funk, and Sanjoy Baruah. "Priority-driven scheduling of periodic task systems on multiprocessors". Real-Time Systems Journal, 25(2-3), pages 187-205, 2003.
- [SrBa02] Anand Srinivasan and Sanjoy Baruah. "Deadline-based scheduling of periodic task systems on multiprocessors". Information Processing Letters, 84(2), pages 93-98, 2002.
- [ZMM03] Dakai Zhu, Daniel Moss□e, and Rami Melhem. "Multiple-resource periodic scheduling problem: how much fairness is necessary?". In RTSS, page 142, 2003.

Portioning and Task Splitting

- [KaYa07] Shinpei Kato and Nobuyuki Yamasaki. "Real-time scheduling with task splitting on multiprocessors". In RTCSA, pages 441-450, 2007.
- [ABB08] Bjorn Andersson, Konstantinos Bletsas, and Sanjoy K. Baruah. "Scheduling arbitrary-deadline sporadic tasks on multiprocessors". In RTSS, 2008.
- [AnBl08] Bjorn Andersson and Konstantinos Bletsas. "Sporadic multiprocessor scheduling with few preemptions". In ECRTS, pages 243-252, 2008.
- [AnTo06] Bjorn Andersson and Eduardo Tovar. "Multiprocessor scheduling with few preemptions. In RTCSA, pages 322-334, 2006.

Other Multiprocessor Algorithms

- [BaCa03] Sanjoy K. Baruah and John Carpenter. "Multiprocessor fixed-priority scheduling with restricted interprocessor migrations". In ECRTS, 2003. (restricted task migration)
- [CAB07] John M. Calandrino, James H. Anderson, and Dan P. Baumberger. "A hybrid real-time scheduling approach for large-scale multicore platforms". In ECRTS, 2007. (physical clustering)

Questions?

Thank You!