Schedulability Analysis for a Mode Transition in Real-Time Multi-Core Systems

Jinkyu Lee and Kang G. Shin

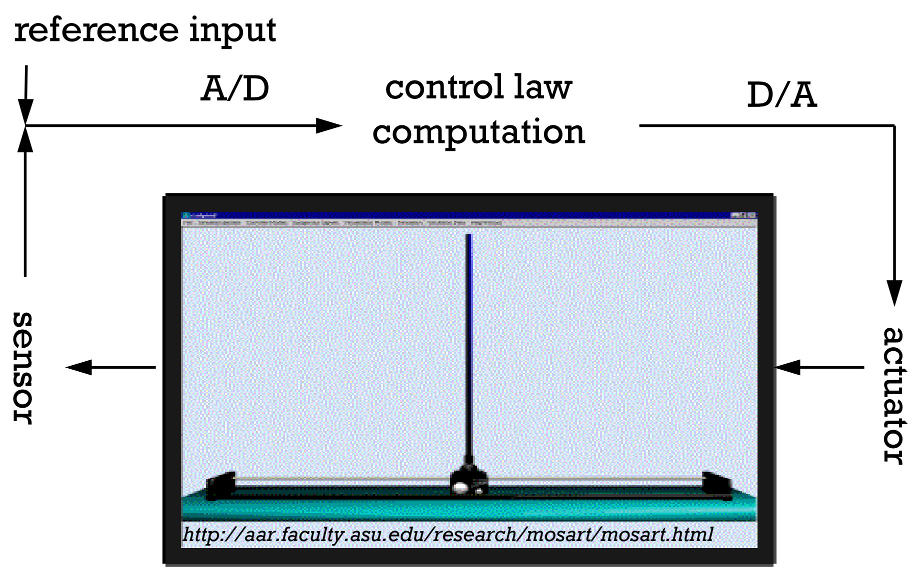
Dept. of EECS, University of Michigan

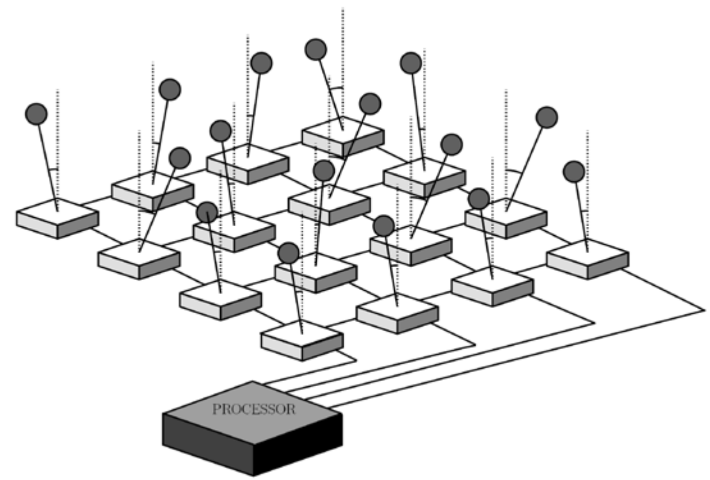
December 4th, 2013

The 34th IEEE Real-Time Systems Symposium

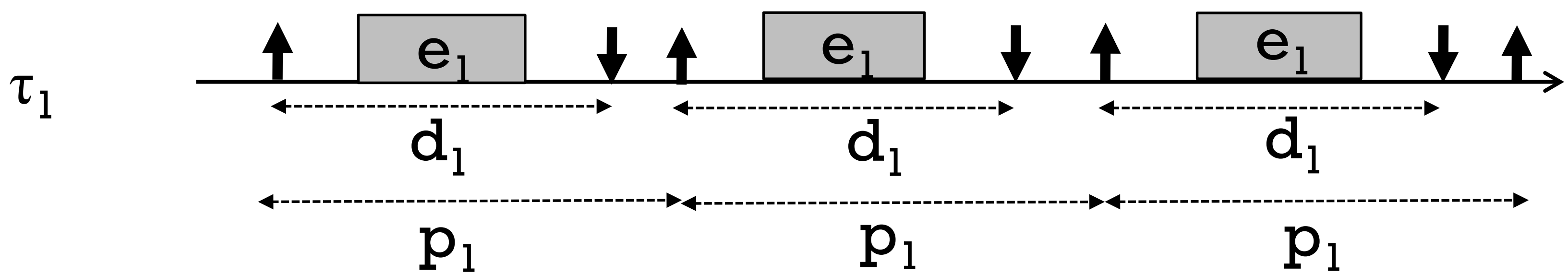
Real-time scheduling

- Can we guarantee all deadlines?





"Task Scheduling for Control Oriented Requirements for Cyber-Physical Systems," RTSS 2008.

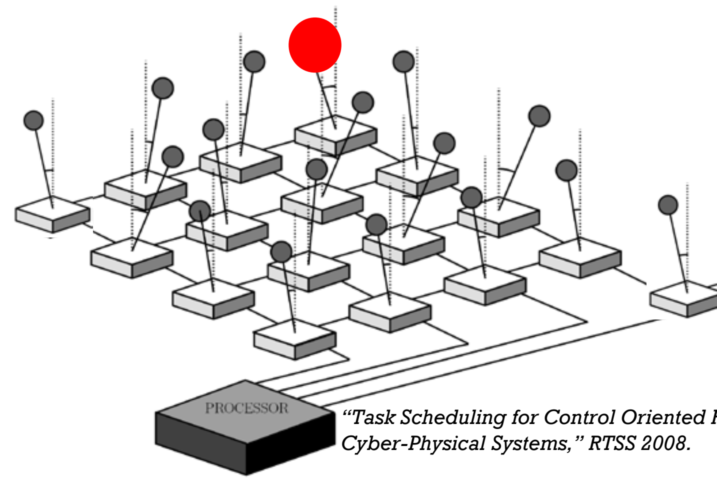


Real-time scheduling

- Can we guarantee all deadlines?
  - Scheduling algorithm
    - Determines which job will be serviced at each time slot
    - e.g., EDF (Earliest Deadline First), FP (task-level Fixed priority)
  - Schedulability analysis
    - Provides offline guarantees on deadlines under a specific scheduling algorithm
    - e.g., response time analysis, utilization bound

Mode transition

- Can we guarantee all deadlines in the presence of?
  - Addition of tasks,
  - Deletion of tasks, and/or
  - Parameter change of tasks

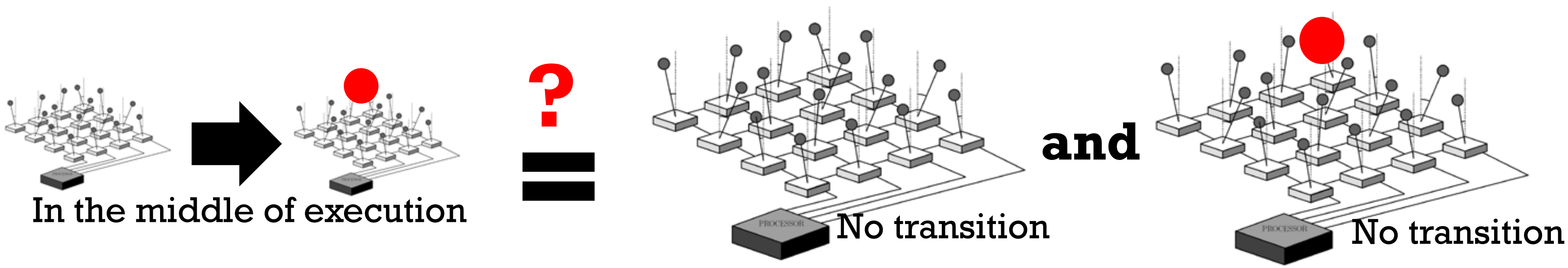


"Task Scheduling for Control Oriented Requirements for Cyber-Physical Systems," RTSS 2008.

- Both unchanged and changed tasks coexist and should not skip/suspend their control updates
  - An additional transition delay that skips/suspends control updates may cause system instability.

Mode transition

Deadline guarantee of | Deadline guarantee of | Deadline guarantee of

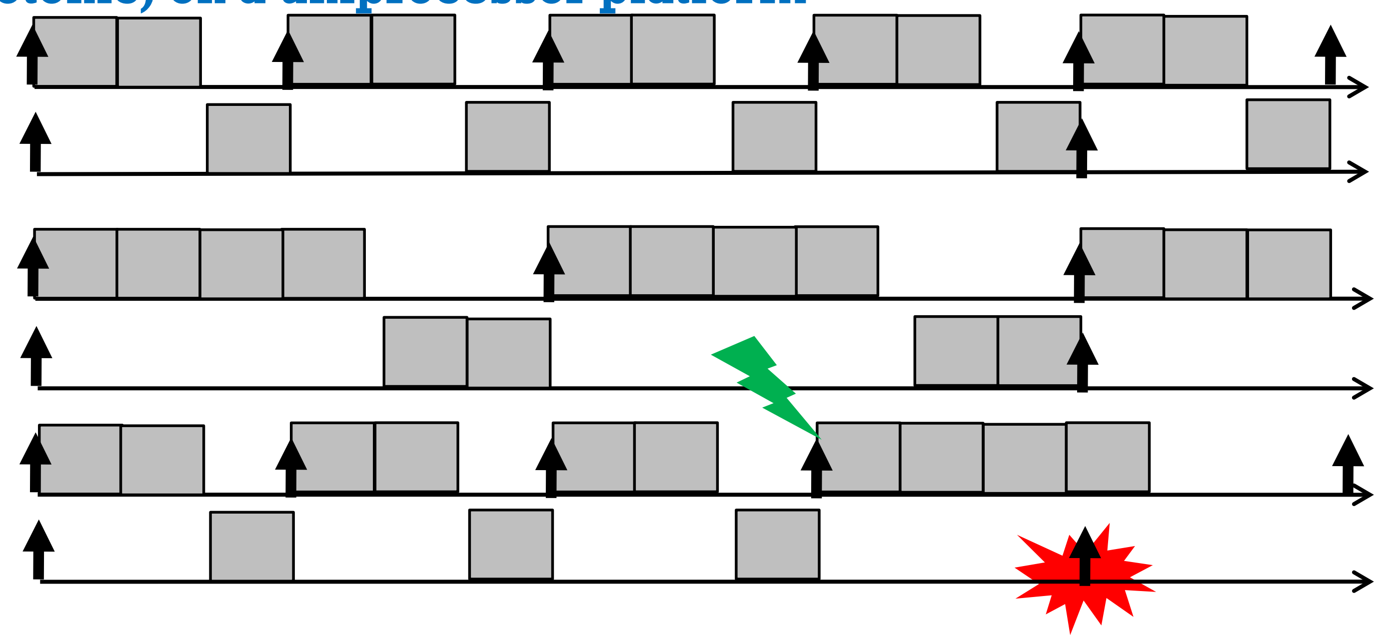


e.g., RM (Rate Monotonic) on a uniprocessor platform

τ^g= (3,2),
(12,4)

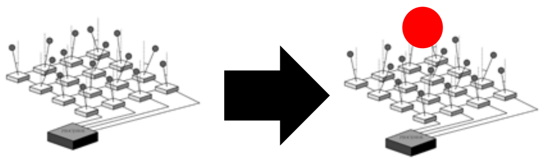
τ^h= (6,4),
(12,4)

$\tau^g \rightarrow \tau^h$= (3,2)→(6,4),
(12,4)



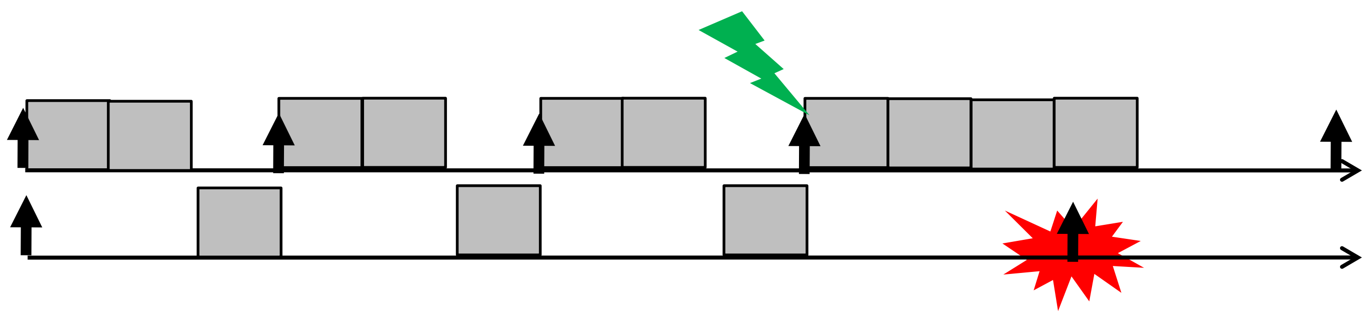
+ Mode transition

Deadline guarantee of



- To develop a schedulability analysis for a mode transition

$\tau^g \rightarrow \tau^h = (3,2) \rightarrow (6,4), (12,4)$



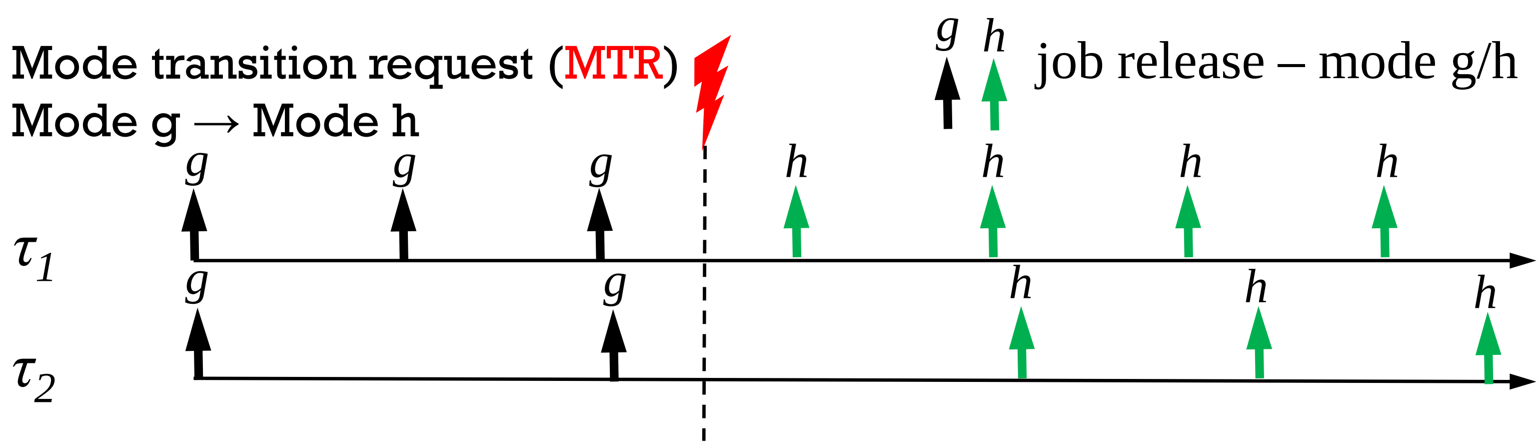
Mode transition: existing work / ours

- To develop a schedulability analysis for a mode transition
  - For real-time multi-core systems
  - Without skipping/suspending control updates

	Uniprocessor	Multi-cores
Skipping/ suspending control updates allowed	Many algorithms Sha et al., RTSJ 1989 Buttazzo et al., TC 2002 Real and Crespo, RTSJ 2004 Guangming, RTSJ 2009 Ahmed et al., RTCSA 2012	EDF Nelis et al, ECRTS 2009 Nelis et al, ECRTS 2011 Rattanatamrong and Fortes, RTCSA 2011
Without skipping/ suspending control updates	FP Tindell et al, RTSS 1992 Kim et al, ICCPS 2012	Few studies

Mode transition: our protocol

- Both unchanged and changed tasks coexist and should not skip/suspend their control updates



Schedulability analysis: properties

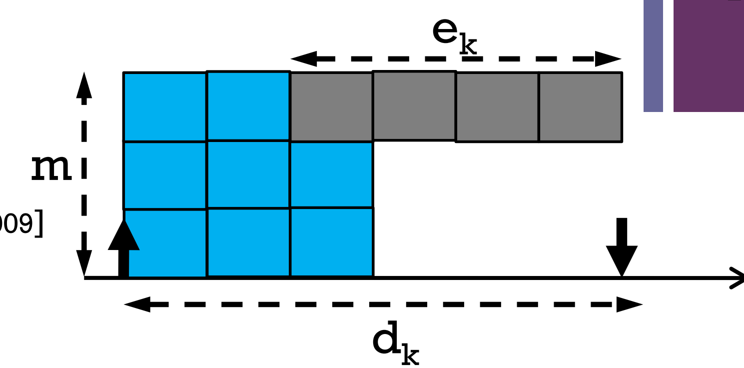
- To develop a schedulability analysis for a mode transition in real-time multi-core systems
  - For a transition protocol without skipping/suspending control updates

Mode g → Mode h

τ_i^g and τ_i^h

- The analysis
  - does not require any online information;
  - focuses on a single transition and is independent of the history of previous transitions; and
  - generalizes the existing schedulability analysis without a transition. [Bertogna et al, TPDS 2009]

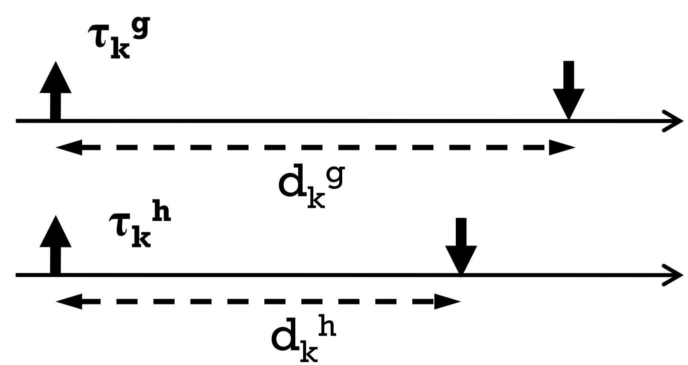
+ Schedulability analysis: framework



- Without a transition [Bertogna et al, TPDS 2009]
- For all task τ_k,

$$\sum_{\tau_i \in \tau \setminus \{\tau_k\}} \min\left(\mathbf{I}(\tau_k \leftarrow \tau_i), d_k - e_k + 1\right) < m \cdot (d_k - e_k + 1)$$

Amount of higher-priority jobs' execution of τ_i than τ_k



- With a transition
  - For all task τ_k, and mode u = {g, h}

$$\sum_{\tau_i \in \tau \setminus \{\tau_k\}} \min\left(\mathbf{I}(\tau_k^u \leftarrow \tau_i^{g \Rightarrow h}), d_k^u - e_k^u + 1\right) < m \cdot (d_k^u - e_k^u + 1)$$

Amount of higher-priority jobs' execution of τ_i^g or τ_i^h than τ_k^u

Schedulability analysis

Amount of higher-priority jobs' execution of τ_i^g or τ_i^h than τ_k^u

- How to calculate $\mathbf{I}(\tau_k^u \leftarrow \tau_i^{g \Rightarrow h})$?
  - Dependent on scheduling algorithms
  - Calculate an upper-bound
    - The amount of execution of τ_i's jobs in an interval of length l with a transition from Mode g to Mode h

$= d_k^g$

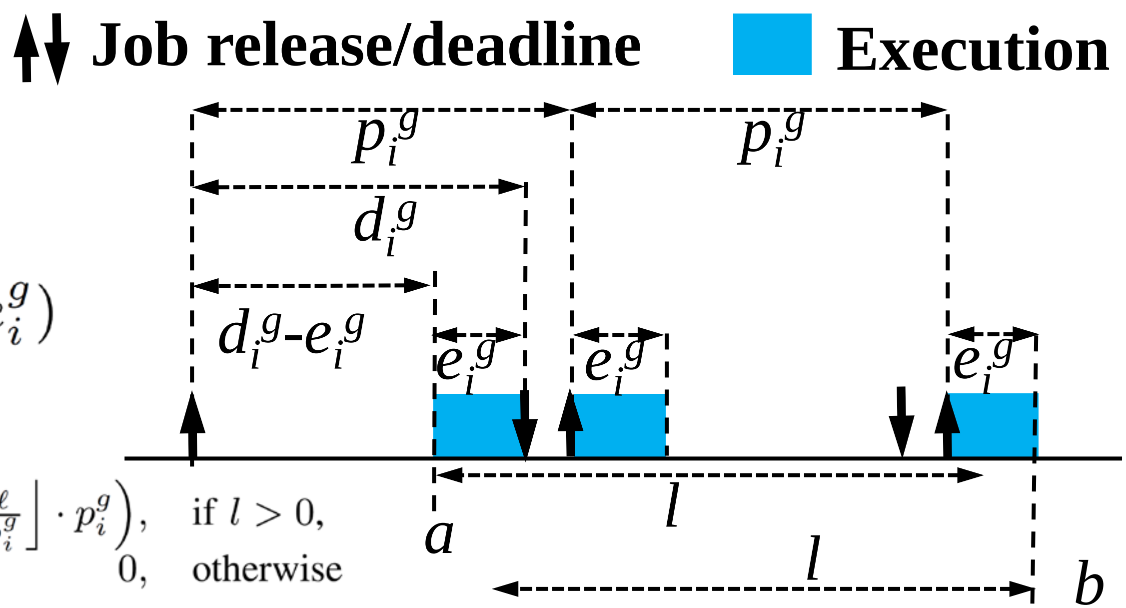
$= d_k^h$

  - In this paper, we focused on
    - FP (task-level Fixed Priority)
    - EDF (Earliest Deadline First) – see the paper

+ Schedulability analysis

- The amount of execution of τ_i's jobs in an interval of length l with a transition from Mode g to Mode h
  - (A) Only jobs of τ_i^g are executed.
  - (B) Only jobs of τ_i^h are executed.
  - (C) Jobs of both τ_i^g and τ_i^h are executed.

[Bertogna et al, TPDS 2009]



$$W_i^g(\ell) \triangleq F_i^g(\ell + d_i^g - e_i^g)$$

$$F_i^g(\ell) \triangleq \begin{cases} \left\lfloor \frac{\ell}{p_i^g} \right\rfloor \cdot e_i^g + \min\left(e_i^g, \ell - \left\lfloor \frac{\ell}{p_i^g} \right\rfloor \cdot p_i^g\right), & \text{if } l > 0, \\ 0, & \text{otherwise} \end{cases}$$

Schedulability analysis

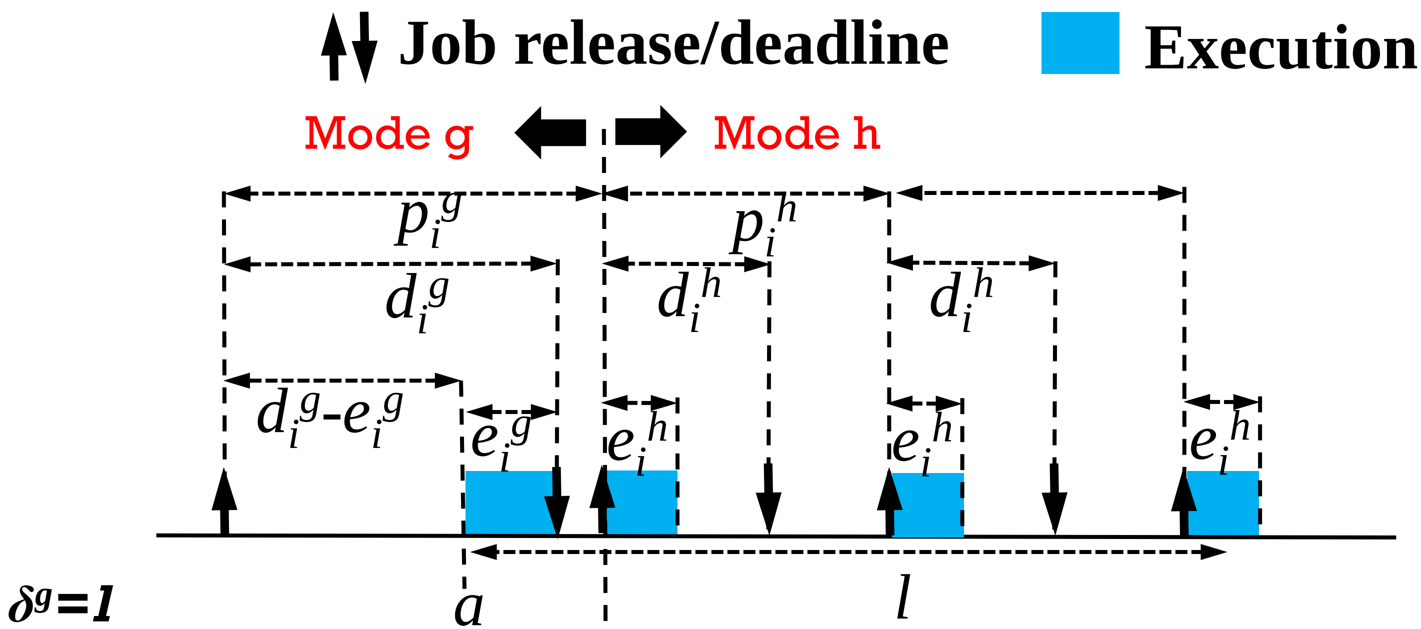
- (C) Jobs of both τ_i^g and τ_i^h are executed.

Observation: the amount of execution of jobs of both τ_i^g and τ_i^h in the interval is maximized with one of the following situations:

- (C1) a job of τ_i^g is executed as late as possible and starts its execution at the beginning of the interval
- (C2) a job of τ_i^h is executed as early as possible and finishes its execution at the end of the interval

Schedulability analysis

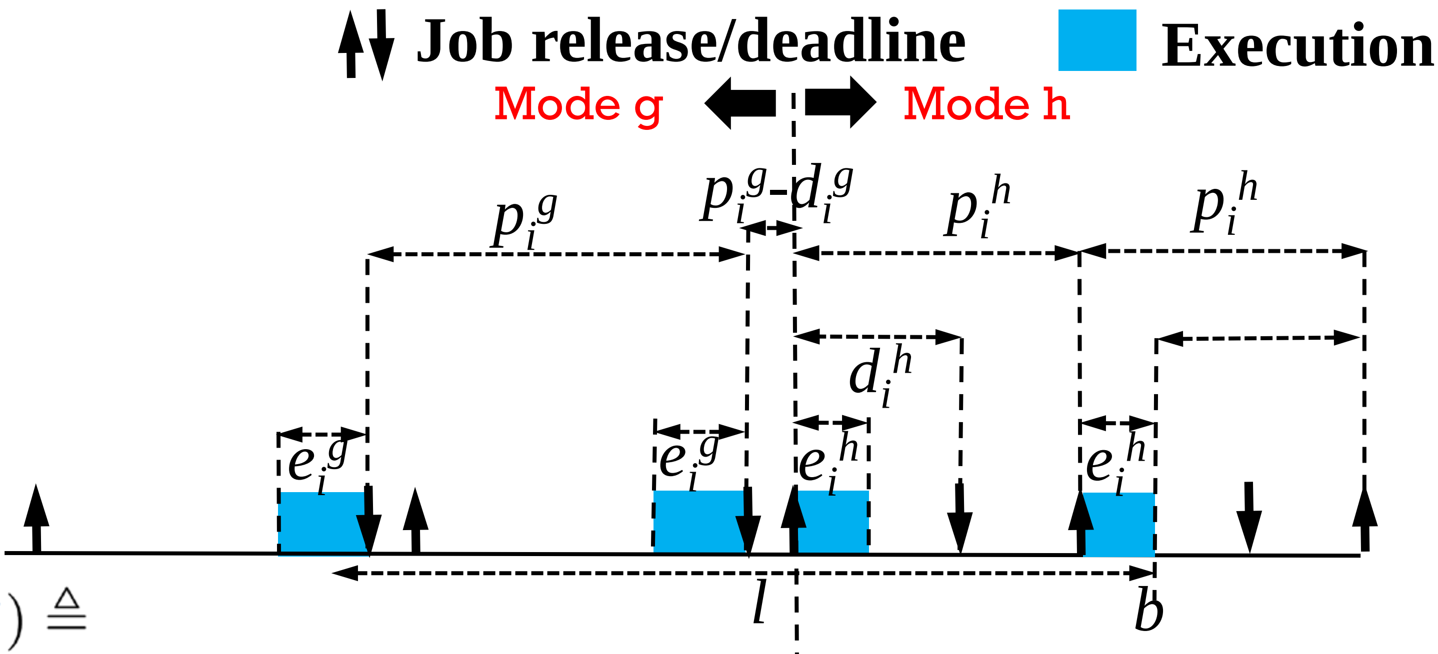
- (C) Jobs of both τ_i^g and τ_i^h are executed.
  - (C1) a job of τ_i^g is executed as late as possible and starts its execution at the beginning of the interval (at a)



$$W_i^{g \Rightarrow h}(\ell, \delta^g) \triangleq \delta \cdot e_i^g + F_i^h(\ell + d_i^g - e_i^g - \delta^g \cdot p_i^g)$$

+ Schedulability analysis

- (C) Jobs of both τ_i^g and τ_i^h are executed.
  - (C2) a job of τ_i^h is executed as early as possible and finishes its execution at the end of the interval (at b)



$$W_i^{\overline{g\Rightarrow h}}(\ell, \delta^h) \triangleq$$

$$\delta^h \cdot e_i^h + F_i^g(\ell + p_i^h - e_i^h - (p_i^g - d_i^g) - \delta^h \cdot p_i^h)$$

δ^h=2

+ Schedulability analysis

- The amount of execution of τ_i's jobs in an interval of length l with a transition from Mode g to Mode h
  - (A) Only jobs of τ_i^g are executed.
  - (B) Only jobs of τ_i^h are executed.
  - (C) Jobs of both τ_i^g and τ_i^h are executed.
    - (C1) when a job of τ_i^g is executed as late as possible and starts its execution at the beginning of the interval (at a)
    - (C2) when a job of τ_i^h is executed as early as possible and finishes its execution at the end of the interval (at b)

An upper-bound of $\mathbf{I}(\tau_k^u \leftarrow \tau_i^{g \Rightarrow h})$ in any case under FP

$$\mathbf{W}_i^{g \Rightarrow h}(\ell) \triangleq \max \Big\{ \overset{(A)}{W_i^g(\ell)}, \overset{(B)}{W_i^h(\ell)}, \overset{(C1)}{\max_{1 \le \delta^g \le \lfloor (\ell + d_i^g - e_i^g)/p_i^g \rfloor} W_i^{g \Rightarrow h}(\ell, \delta^g)}, \overset{(C2)}{\max_{1 \le \delta^h \le \lfloor (\ell + p_i^h - e_i^h)/p_i^h \rfloor} W_i^{\overline{g \Rightarrow h}}(\ell, \delta^h)} \Big\}$$

Schedulability analysis

$$\sum_{\tau_i \in \tau \setminus \{\tau_k\}} \min\left(\mathbf{I}(\tau_k^u \leftarrow \tau_i^{g \Rightarrow h}), d_k^u - e_k^u + 1\right) < m \cdot (d_k^u - e_k^u + 1)$$

$$\sum_{\tau_i \in \tau \setminus \{\tau_k\}} \min\left(\mathbf{W}_i^{g \Rightarrow h}(d_k^u), d_k^u - e_k^u + 1\right) < m \cdot (d_k^u - e_k^u + 1)$$

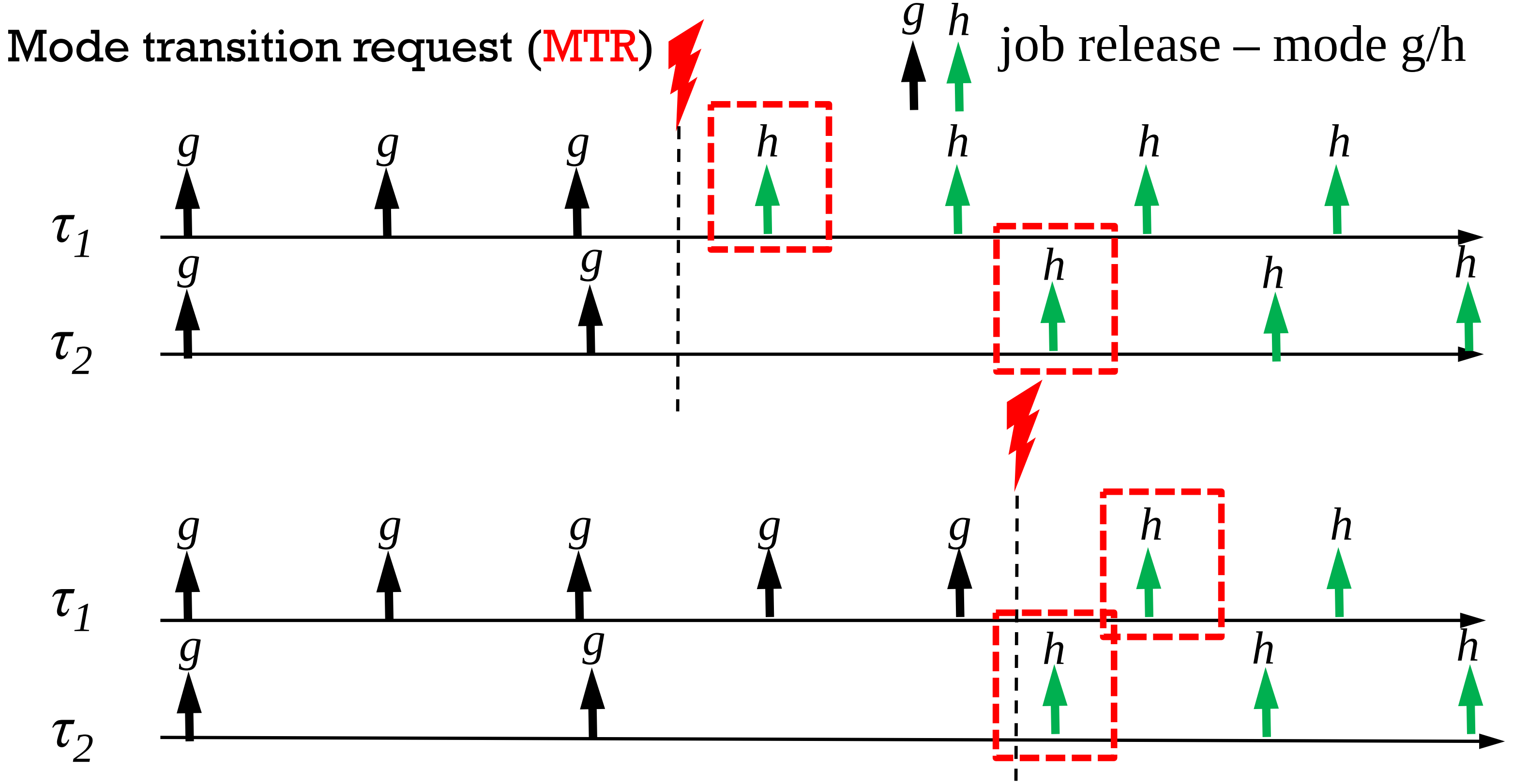
Whose priority is higher than τ_k

An upper-bound of $\mathbf{I}(\tau_k^u \leftarrow \tau_i^{g \Rightarrow h})$ in any case under FP

- *Theorem:* suppose that a task set τ makes a transition from Mode g to Mode h. Then, a task set τ with the transition is schedulable under FP, if the above equation holds for all τ_k and u={g,h}.
- Schedulability analysis is done!
  - Can we improve schedulability by enforcing a specific transition order among tasks?

Transition order assignment

- Our mode transition protocol does not control the order of transitions among tasks.

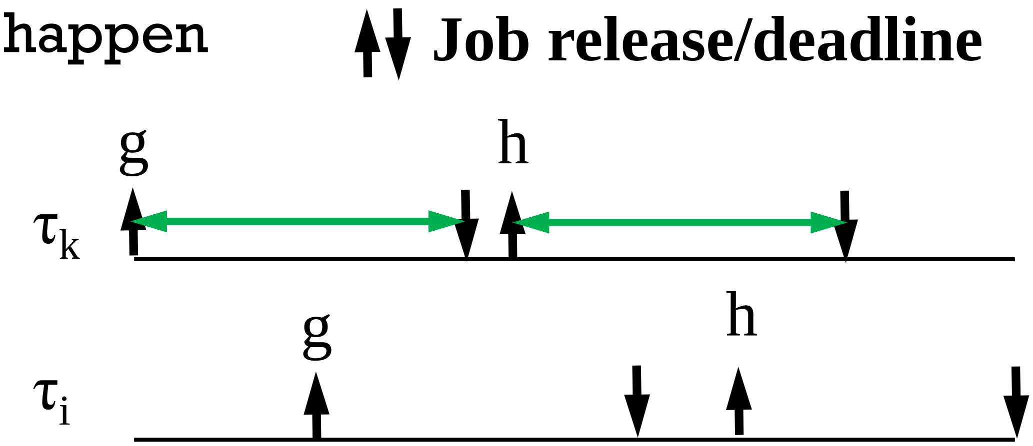


Transition order assignment

- Our mode transition protocol does not control the order of transitions among tasks.
- What if we can control the order of transitions?
  - e.g, task i's transition is performed before task j's transition.
  - Can we improve schedulability?
  - If so, how can we find an optimal transition sequence?

Transition order assignment

- What if τ_k's transition from Mode g to Mode h is performed before τ_i's transition?
  - τ_k^g: Only (A) can happen
  - τ_k^h: (A), (B) and (C) can happen



The amount of execution of τ_i's jobs in an interval of length l with a transition from Mode g to Mode h

(A) Only jobs of τ_i^g are executed.

(B) Only jobs of τ_i^h are executed.

(C) Jobs of both τ_i^g and τ_i^h are executed.

+ Transition order assignment

- What if τ_k's transition from Mode g to Mode h is performed before τ_i's transition?
  - τ_k^g: Only (A) can happen

$$\mathbf{W}_i^{g\Rightarrow h}(\ell) \triangleq \max\left\{ \underbrace{W_i^g(\ell)}_{\text{(A)}},\ \underbrace{W_i^h(\ell)}_{\text{(B)}},\ \underbrace{\max_{1\le\delta^g\le\lfloor(\ell+d_i^g-e_i^g)/p_i^g\rfloor} W_i^{g\Rightarrow h}(\ell,\delta^g)}_{\text{(C1)}},\ \underbrace{\max_{1\le\delta^h\le\lfloor(\ell+p_i^h-e_i^h)/p_i^h\rfloor} W_i^{\overline{g\Rightarrow h}}(\ell,\delta^h)}_{\text{(C2)}} \right\} \Rightarrow \underbrace{W_i^g(\ell)}_{\text{(A)}}$$

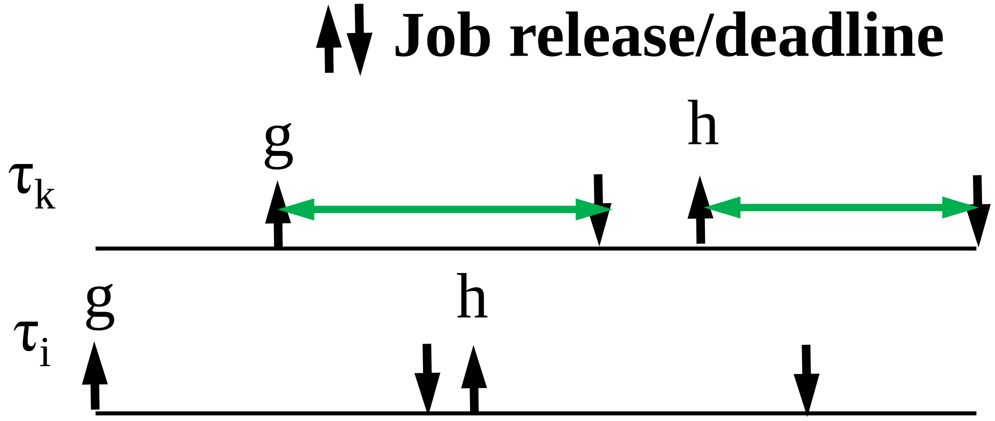
Reduced!

$$\sum_{\tau_i\in\tau\setminus\{\tau_k\}} \min\left(\overset{W_i^g}{\cancel{\mathbf{W}_i^{g\Rightarrow h}}}(d_k^u),\ d_k^u-e_k^u+1\right) < m\cdot(d_k^u-e_k^u+1)$$

Schedulability improved!

Transition order assignment

- What if τ_k's transition from Mode g to Mode h is performed after τ_i's transition?
  - τ_k^g: (A), (B) and (C) can happen
  - τ_k^h: Only (B) can happen



The amount of execution of τ_i's jobs in an interval of length l with a transition from Mode g to Mode h

(A) Only jobs of τ_i^g are executed.

(B) Only jobs of τ_i^h are executed.

(C) Jobs of both τ_i^g and τ_i^h are executed.

+ Transition order assignment

- What if τ_k's transition from Mode g to Mode h is performed after τ_i's transition?
  - τ_k^h: Only (B) can happen

$$\mathbf{W}_i^{g \Rightarrow h}(\ell) \triangleq \max \begin{cases} \overset{\text{(A)}}{W_i^g(\ell)}, \quad \overset{\text{(B)}}{W_i^h(\ell)}, \\ \text{(C1)} \max_{1 \le \delta^g \le \lfloor (\ell + d_i^g - e_i^g)/p_i^g \rfloor} W_i^{g \Rightarrow h}(\ell, \delta^g), \\ \text{(C2)} \max_{1 \le \delta^h \le \lfloor (\ell + p_i^h - e_i^h)/p_i^h \rfloor} W_i^{\overline{g \Rightarrow h}}(\ell, \delta^h) \end{cases}$$

➡ (B) $W_i^h(\ell)$

Reduced!

$$\sum_{\tau_i \in \tau \setminus \{\tau_k\}} \min \left(\sout{\mathbf{W}_i^{g \Rightarrow h}} \; W_i^h (d_k^u), d_k^u - e_k^u + 1 \right) < m \cdot (d_k^u - e_k^u + 1)$$

Schedulability improved!

Transition order assignment

- We can improve the schedulability by determining the transition order of tasks!

- How can we find an optimal transition sequence?
  - Effect of my order placement on my schedulability
  - Effect of my order placement on others' schedulability

- Derived properties for the two effects

+ Transition order assignment

- Effect of my order placement on my schedulability
  - Find a set of tasks that should be placed in the first: (F1)
  - Find a set of tasks that should be placed in the last: (L1)
- Effect of my order placement on others' schedulability
  - Find a set of tasks that should be placed in the first: (F2)
  - Find a set of tasks that should be placed in the last: (L2)
- Optimal group-level transition order assignment
  - $\tau(F)$: tasks in (F1) ∩ (F2) In the first
  - $\tau(M)$: remainders
  - $\tau(L)$: tasks in (L1) ∩ (L2) In the last

Transition order assignment

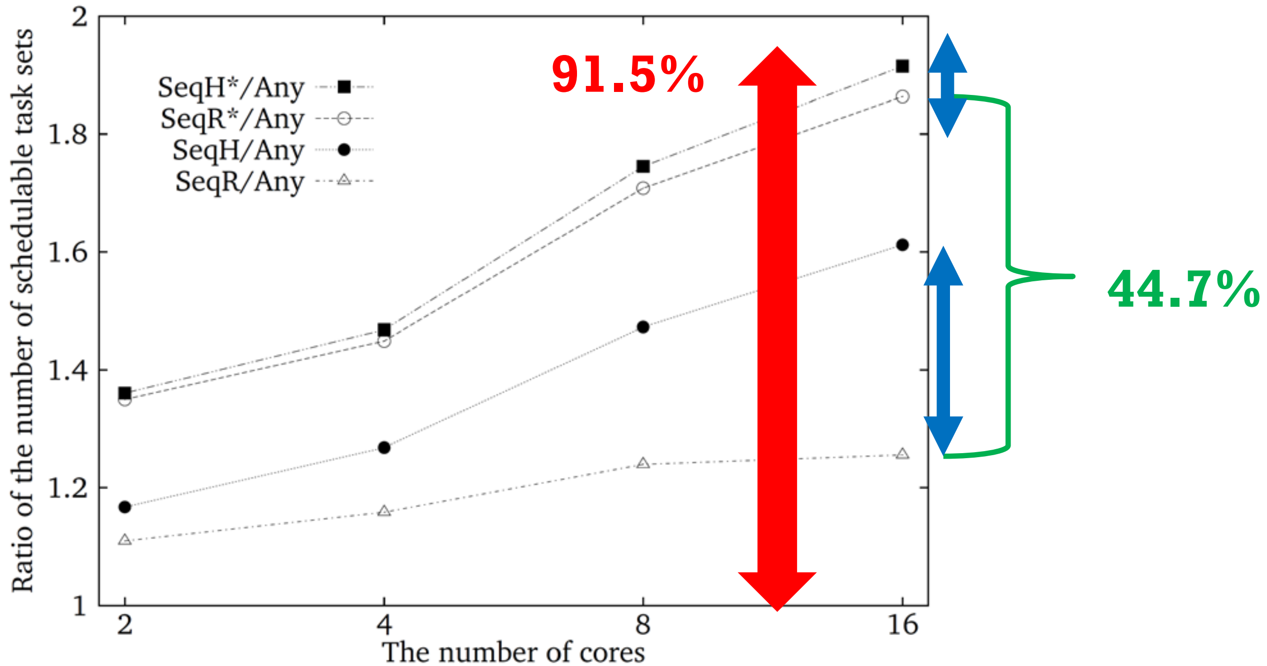
- Optimal group-level transition order assignment
  - $\tau(F)$: tasks in $(F1) \cap (F2)$ In the first
  - $\tau(M)$: remainders
  - $\tau(L)$: tasks in $(L1) \cap (L2)$ In the last

- Proved the relative orders in $\tau(F)$ and $\tau(L)$ do not affect the schedulability of every task. Optimality

- Developed a heuristic algorithm for determining the relative orders in $\tau(M)$

Evaluation

- Any: any-order transition
- SeqR: the entire order is determined randomly
- SeqH: the entire order is determine by our heuristic algorithm
- SeqR*: grouped by τ(F), τ(M) and τ(L), and the relative order of tasks in τ(M) is determined randomly
- SeqH*: grouped by τ(F), τ(M) and τ(L), and the relative order of tasks in τ(M) is determined by our heuristic algorithm



Conclusion

- Addressed the problem of guaranteeing the timing requirements of task sets with mode transitions without disrupting task execution in real-time multi-core systems.
  - Generalized existing theories for no mode transition
  - Introduced the problem of determining transition sequence, and solved it by deriving the useful properties of an optimal transition order

- Future work
  - Improve schedulability – applying response time analysis
    - Need the history of previous modes – how to manage?

References

[1] C. Liu and J. Layland, "Scheduling algorithms for multi-programming in a hard-real-time environment," Journal of the ACM, vol. 20, no. 1, pp. 46–61, 1973.

[2] M. Wahler, S. Richter, and M. Oriol, "Dynamic software updates for real-time systems," in Proceedings of the 2nd International Workshop on Hot Topics in Software Upgrades, 2009.

[3] M. Wahler, S. Richter, S. Kumar, and M. Oriol, "Non-disruptive largescale component updates for real-time controller," in Proceedings of the 27th International Conferences on Data Engineering Workshops, 2011, pp. 174.178.

[4] J. C. Romero and M. Garcia-Valls, "Scheduling component replacement for timely execution in dynamic systems," To appear in Software - Practice and Experience, vol. -, no. -, pp. ., -.

[5] L. Sha, R. Rajkumar, J. Lehoczky, and K. Ramamritham, "Mode change protocols for priority-driven preemptive scheduling," Real-Time Systems, vol. 1, no. 3, pp. 243.264, 1989.

[6] K. W. Tindell, A. Burns, and A. J. Wellings, "Mode changes in priority pre-emptively scheduled systems," in Proceedings of IEEE Real-Time Systems Symposium (RTSS), 1992, pp. 100.109.

[7] J. Real and A. Crespo, "Mode change protocols for real-time systems: A survey and a new proposal," Real-Time Systems, vol. 26, no. 2, pp. 161.197, 2004.

[8] M. Ahmed, N. Fisher, and D. Grosu, "A parallel algorithm for EDFschedulability analysis of multi-modal real-time systems," in Proceedings of IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 2012, pp. 154.163.

[9] J. Kim, K. Lakshmanan, and R. Rajkumar, "Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems," in Proceedings of IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS), 2012, pp. 55.64.

[10] G. C. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni, "Elastic scheduling for flexible workload management," IEEE Transactions on Computers, vol. 51, no. 3, pp. 289.302, 2002.

[11] Q. Guangming, "An earlier time for inserting and/or accelerating tasks," Real-Time Systems, vol. 41, no. 3, pp. 181.194, 2009.

[12] V. Nelis, J. Goossens, and B. Andersson, "Two protocols for scheduling multi-mode real-time systems upon identical multiprocessor platforms," in Proceedings of Euromicro Conference on Real-Time Systems (ECRTS), 2009, pp. 151.160.

[13] V. Nelis, B. Andersson, J. Marinho, and S. M. Petters, "Global-EDF scheduling of multimode real-time systems considering mode independent tasks," in Proceedings of Euromicro Conference on Real-Time Systems (ECRTS), 2011, pp. 205.214.

+ References

[14] P. Rattanatamrong and J. A. B. Fortes, "Mode transition for online scheduling of adaptive real-time systems on multiprocessors," in Proceedings of IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), 2011, pp. 25.32.

[15] L. T. X. Phan, S. Chakraborty, and P. S. Thiagarajan, "A multimode real-time calculus," in Proceedings of IEEE Real-Time Systems Symposium (RTSS), 2008, pp. 59.69.

[16] N. Stoimenov, S. Perathoner, and L. Thiele, "Reliable mode changes in real-time systems with fixed priority or EDF scheduling," in Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE), 2009, pp. 99.104.

[17] Y. Hang and H. Hansson, "Timing analysis for mode switch in component-based multi-mode systems," in Proceedings of Euromicro Conference on Real-Time Systems (ECRTS), 2012, pp. 255.264.

[18] N. Audsley, A. Burns, M. Richardson, and A. Wellings, "Hard realtime scheduling: the deadline-monotonic approach," in Proceedings of the IEEE Workshop on Real-Time Operating Systems and Software, May 1991, pp. 133.137.

[19] T. P. Baker, "Multiprocessor EDF and deadline monotonic schedulability analysis," in RTSS, 2003.

[20] M. Bertogna, M. Cirinei, and G. Lipari, "Improved schedulability analysis of EDF on multiprocessor platforms," in Proceedings of Euromicro Conference on Real-Time Systems (ECRTS), 2005, pp. 209.218.

[21] A. Mok, "Fundamental design problems of distributed systems for the hard-real-time environment," Ph.D. dissertation, Massachusetts Institute of Technology, 1983.

[22] M. Bertogna, M. Cirinei, and G. Lipari, "Schedulability analysis of global scheduling algorithms on multiprocessor platforms," IEEE Transactions on Parallel and Distributed Systems, vol. 20, pp. 553.566, 2009.

[23] T. P. Baker, M. Cirinei, and M. Bertogna, "EDZL scheduling analysis," Real-Time Systems, vol. 40, pp. 264.289, 2008.

[24] J. Lee, A. Easwaran, and I. Shin, "LLF schedulability analysis on multiprocessor platforms," in Proceedings of IEEE Real-Time Systems Symposium (RTSS), 2010, pp. 25.36.

[25] J. Lee and K. G. Shin, "Supplement: schedulability analysis for continuous task execution during a mode transition in real-time multi-core systems,"http://kabru.eecs.umich.edu/papers/publications/2013/supplement.pdf.

[26] T. P. Baker, "Comparison of empirical success rates of global vs. paritioned fixed-priority and EDF scheduling for hand real time," Dept. of Computer Science, Florida State University, Tallahasee, Tech. Rep. TR-050601, 2005.