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?

reference input

A/D  control law  D/A
computation

sensor  actuator

τ₁
e₁  d₁  e₁  d₁  e₁  d₁  e₁

p₁  p₁  p₁


http://aar.faculty.asu.edu/research/mosart/mosart.html
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
Can we guarantee all deadlines in the presence of?

- **Addition** of tasks,
- **Deletion** of tasks, and/or
- **Parameter change** of tasks

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

In the middle of execution

Deadline guarantee of

Deadline guarantee of

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

\[ \tau^g = (3,2), (12,4) \]

\[ \tau^h = (6,4), (12,4) \]

\[ \tau^g \rightarrow \tau^h = (3,2) \rightarrow (6,4), (12,4) \]
Mode transition

Deadline guarantee of

- To develop a schedulability analysis for a mode transition

$\tau^g \rightarrow \tau^h = (3,2) \rightarrow (6,4), (12,4)$
To develop a schedulability analysis for a mode transition
- For real-time multi-core systems
- Without skipping/suspending control updates

<table>
<thead>
<tr>
<th>Skipping/suspending control updates allowed</th>
<th>Uniprocessor</th>
<th>Multi-cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many algorithms</td>
<td>Sha et al., RTSJ 1989</td>
<td>EDF</td>
</tr>
<tr>
<td></td>
<td>Buttazzo et al., TC 2002</td>
<td>Nelis et al, ECRTS 2009</td>
</tr>
<tr>
<td></td>
<td>Real and Crespo, RTSJ 2004</td>
<td>Nelis et al, ECRTS 2011</td>
</tr>
<tr>
<td></td>
<td>Guangming, RTSJ 2009</td>
<td>Rattanamrong and Fortes, RTCSA 2011</td>
</tr>
<tr>
<td></td>
<td>Ahmed et al., RTCSA 2012</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Without skipping/suspending control updates</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tindell et al, RTSS 1992</td>
<td>Few studies</td>
</tr>
<tr>
<td>Kim et al, ICCPS 2012</td>
<td></td>
</tr>
</tbody>
</table>
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

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 \[^{[\text{Bertogna et al, TPDS 2009}]}\]
  
  For all task \(\tau_k\),
  
  \[
  \sum_{\tau_i \in \tau \setminus \{\tau_k\}} \min \left( I(\tau_k \leftarrow \tau_i), d_k - e_k + 1 \right) < m \cdot (d_k - e_k + 1)
  \]

  Amount of higher-priority jobs’ execution of \(\tau_i\) than \(\tau_k\)

- With a transition
  
  For all task \(\tau_k\), and mode \(u = \{g, h\}\)
  
  \[
  \sum_{\tau_i \in \tau \setminus \{\tau_k\}} \min \left( I(\tau_k^u \leftarrow \tau_i^{g \Rightarrow h}), d_k^u - e_k^u + 1 \right) < m \cdot (d_k^u - e_k^u + 1)
  \]

  Amount of higher-priority jobs’ execution of \(\tau_i^g\) or \(\tau_i^h\) than \(\tau_k^u\)
Schedulability analysis

How to calculate \( I(\tau^u_k \leftarrow \tau^g_i \Rightarrow h) \)?

- Amount of higher-priority jobs’ execution of \( \tau^g_i \) or \( \tau^h_i \) than \( \tau^u_k \)
- Dependent on scheduling algorithms

Calculate an upper-bound

- The amount of execution of \( \tau_i \)'s jobs in an interval of length \( l \) with a transition from Mode \( g \) to Mode \( h \)

In this paper, we focused on

- FP (task-level Fixed Priority)
- EDF (Earliest Deadline First) – see the paper

\[ d^g_k = \]
\[ d^h_k = \]
Schedulability analysis

- The amount of execution of τᵢ’s jobs in an interval of length l with a transition from Mode g to Mode h
  - (A) Only jobs of τᵢᵍ are executed. [Bertogna et al, TPDS 2009]
  - (B) Only jobs of τᵢʰ are executed.
  - (C) Jobs of both τᵢᵍ and τᵢʰ are executed.

\[
Wᵢᵍ(ℓ) \triangleq Fᵢᵍ(ℓ + dᵢᵍ - eᵢᵍ)
\]

\[
Fᵢᵍ(ℓ) \triangleq \begin{cases} 
\left\lfloor \frac{ℓ}{pᵢⁱ} \right\rfloor \cdot eᵢᵍ + \min\left(eᵢᵍ, ℓ - \left\lfloor \frac{ℓ}{pᵢⁱ} \right\rfloor \cdot pᵢⁱ\right), & \text{if } ℓ > 0, \\
0, & \text{otherwise}
\end{cases}
\]
Schedulability analysis

- (C) Jobs of both $\tau_{i}^{g}$ and $\tau_{i}^{h}$ are executed.

Observation: the amount of execution of jobs of both $\tau_{i}^{g}$ and $\tau_{i}^{h}$ in the interval is maximized with one of the following situations:

- (C1) a job of $\tau_{i}^{g}$ is executed as late as possible and starts its execution at the beginning of the interval
- (C2) a job of $\tau_{i}^{h}$ is executed as early as possible and finishes its execution at the end of the interval
Schedulability analysis

- (C) Jobs of both $\tau_i^g$ and $\tau_i^h$ are executed.
- (C1) a job of $\tau_i^g$ 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 $\tau_i^g$ and $\tau_i^h$ are executed.
- (C2) A job of $\tau_i^h$ is executed as early as possible and finishes its execution at the end of the interval (at $b$)

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

\[ \delta^h = 2 \]
Schedulability analysis

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

\[
W_i^{g \rightarrow h}(\ell) \triangleq \max \left\{ \begin{array}{ll}
\text{(A)} & W_i^g(\ell), \\
\text{(B)} & W_i^h(\ell), \\
\end{array} \right.
\]

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

\[
\begin{align*}
\text{(C1)} & \quad \max_{1 \leq \delta^g \leq \left\lceil (\ell + d_i^g - e_i^g)/p_i^g \right\rceil} W_i^{g \rightarrow h}(\ell, \delta^g), \\
\text{(C2)} & \quad \max_{1 \leq \delta^h \leq \left\lceil (\ell + p_i^h - e_i^h)/p_i^h \right\rceil} \overline{W_i^{g \rightarrow h}}(\ell, \delta^h)
\end{align*}
\]
Schedulability analysis

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

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

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

**Theorem:** Suppose that a task set \( \tau \) makes a transition from Mode \( g \) to Mode \( h \). Then, a task set \( \tau \) with the transition is schedulable under FP, if the above equation holds for all \( \tau_k \) and \( u=\{g,h\} \).

Schedulability analysis is done!

- Can we improve schedulability by enforcing a specific transition order among tasks?
Our mode transition protocol does not control the order of transitions among tasks.

Mode transition request (MTR)
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 $\tau_k$'s transition from Mode $g$ to Mode $h$ is performed before $\tau_i$'s transition?

- $\tau_k^g$: Only (A) can happen
- $\tau_k^h$: (A), (B) and (C) can happen

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

(A) Only jobs of $\tau_i^g$ are executed.
(B) Only jobs of $\tau_i^h$ are executed.
(C) Jobs of both $\tau_i^g$ and $\tau_i^h$ are executed.
What if \( \tau_k \)'s transition from Mode g to Mode h is performed before \( \tau_i \)'s transition?

\( \tau_k^g \): Only (A) can happen

\[
W^{g\Rightarrow h}_i(\ell) \triangleq \max \left\{ \begin{array}{ll}
(A) & W^g_i(\ell), \\
(B) & W^h_i(\ell), \\
(C1) & \max_{1 \leq \delta^g \leq [(\ell + d^g_i - e^g_i)/p^g_i]} W_i^{g\Rightarrow h}(\ell, \delta^g), \\
(C2) & \max_{1 \leq \delta^h \leq [(\ell + p^h_i - e^h_i)/p^h_i]} W_i^{g\Rightarrow h}(\ell, \delta^h) \end{array} \right. \\
\sum_{\tau_i \in \tau \setminus \{\tau_k\}} \min \left( \frac{W_i^{g\Rightarrow h}(d^u_k), d^u_k - e^u_k + 1}{m \cdot (d^u_k - e^u_k + 1)} \right) < m \cdot (d^u_k - e^u_k + 1)
\]

Reduced!

Schedulability improved!
Transition order assignment

What if $\tau_k$’s transition from Mode $g$ to Mode $h$ is performed after $\tau_i$’s transition?

- $\tau_k^g$: (A), (B) and (C) can happen
- $\tau_k^h$: Only (B) can happen

The amount of execution of $\tau_i$’s jobs in an interval of length $l$ with a transition from Mode $g$ to Mode $h$

(A) Only jobs of $\tau_i^g$ are executed.
(B) Only jobs of $\tau_i^h$ are executed.
(C) Jobs of both $\tau_i^g$ and $\tau_i^h$ are executed.
Transition order assignment

- What if $\tau_k$’s transition from Mode $g$ to Mode $h$ is performed after $\tau_i$’s transition?

- $\tau_k^h$: Only (B) can happen

$$ W_i^{g\Rightarrow h}(\ell) \triangleq \max \left\{ \begin{array}{ll} \text{(A)} & W_i^g(\ell), \\ \text{(B)} & W_i^h(\ell), \\ \text{(C1)} & \max_{1 \leq \delta^g \leq \left\lfloor (\ell + d_i^g - e_i^g)/p_i^g \right\rfloor} W_i^{g\Rightarrow h}(\ell, \delta^g), \\ \text{(C2)} & \max_{1 \leq \delta^h \leq \left\lfloor (\ell + p_i^h - e_i^h)/p_i^h \right\rfloor} W_i^{g\Rightarrow h}(\ell, \delta^h) \end{array} \right\} $$

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

Schedulability improved!
Transition order assignment

- 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: \((F_1)\)
  - Find a set of tasks that should be placed in the last: \((L_1)\)

- Effect of my order placement on others’ schedulability
  - Find a set of tasks that should be placed in the first: \((F_2)\)
  - Find a set of tasks that should be placed in the last: \((L_2)\)

- Optimal group-level transition order assignment
  - \(\tau(F)\): tasks in \((F_1) \cap (F_2)\)  **In the first**
  - \(\tau(M)\): remainders
  - \(\tau(L)\): tasks in \((L_1) \cap (L_2)\)  **In the last**
Transition order assignment

- Optimal group-level transition order assignment
  - $\tau(F)$: tasks in $(F_1) \cap (F_2)$  \textbf{In the first}
  - $\tau(M)$: remainders
  - $\tau(L)$: tasks in $(L_1) \cap (L_2)$  \textbf{In the last}

- Proved the relative orders in $\tau(F)$ and $\tau(L)$ do not affect the schedulability of every task. \textbf{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 determined by our heuristic algorithm
- **SeqR**: grouped by $\tau(F)$, $\tau(M)$ and $\tau(L)$, and the relative order of tasks in $\tau(M)$ is determined randomly
- **SeqH**: grouped by $\tau(F)$, $\tau(M)$ and $\tau(L)$, and the relative order of tasks in $\tau(M)$ is determined by our heuristic algorithm

![Graph showing the ratio of the number of schedulable task sets vs. the number of cores]

- **91.5%**
- **44.7%**
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?


