MC-Fluid: rate assignment strategies

Saravanan Ramanathan and Arvind Easwaran

Nanyang Technological University, Singapore

December 1, 2015
Outline

1. Introduction and Background
   - Mixed-Criticality (MC) System
   - Fluid Scheduling
   - Dual-rate MC Fluid Scheduling

2. Motivation
   - Challenges in Dual-rate MC Fluid Model

3. Proposed Strategy
   - MC-Sort algorithm
   - MC-Slope algorithm

4. Evaluation
   - Schedulability

5. Future Work
   - Multi-rate model
Outline

1. Introduction and Background
   - Mixed-Criticality (MC) System
   - Fluid Scheduling
   - Dual-rate MC Fluid Scheduling

2. Motivation
   - Challenges in Dual-rate MC Fluid Model

3. Proposed Strategy
   - MC-Sort algorithm
   - MC-Slope algorithm

4. Evaluation
   - Schedulability

5. Future Work
   - Multi-rate model
Mixed-Criticality (MC) Task Model

Implicit Deadline Sporadic Task: \( \tau_i = (T_i, L_i, C_i) \)

- \( T_i \) is the minimum separation between successive job releases
  - Since we consider implicit deadline tasks, deadline = \( T_i \)
- \( L_i \) denotes the criticality level of task (assume 2 levels)
  - LO denoting low-criticality and HI denoting high-criticality
- \( C_i = \{C_i^L, C_i^H\} : C_i^L \) denotes LO worst-case execution time (WCET), and \( C_i^H (\geq C_i^L) \) denotes HI WCET
  - \( C_i^H = C_i^L \) if \( L_i = LC \)
Task system behaviours: A MC task system with two criticality levels can exhibit the following behaviours

- **LO mode**: The system is in this behaviour as long as no task has executed beyond its LO WCET
- **HI mode**: The system switches to this behaviour when any HI task executes beyond its LO WCET

**MC Correctness**: A MC system is said to be correct if

- In LO mode: All tasks with LO WCETs are schedulable
- In HI mode: Only HI tasks with HI WCETs are schedulable
  - All LO tasks are dropped
Fluid Scheduling: Each task is assigned a fractional processing capacity at each time instant

- **Schedulability**: A task $\tau_i$ can meet its deadline if
  - Rate $(\theta_i) \times \text{Period } (T_i) \geq \text{WCET}$

- **Feasibility**: A task rate $\theta_i$ is valid under a $m$ core system if
  - $\theta_i \leq 1$
  - $\sum_{\tau_i \in \tau} \theta_i \leq m$

**Figure**: Fluid scheduling
MC-Fluid Scheduling

MC-Fluid Platform: Each task is executed with LO-rate ($\theta_i^L$) in LO mode and HI-rate ($\theta_i^H$) in HI mode

- At mode switch, execution requirement is **changed**
- Execution rate is **changed**
- **Carry-over job**: A job released in LO mode and finished in HI mode

**Figure**: Carry-over job
MC-Fluid Scheduling

Rate Assignment:

- **Worst-case mode switch pattern**
  - Minimum \( \theta^L_i - u^L_i \)
- Construct an optimization problem
  - Solve it by convex optimization
- **Optimal** rate assignment algorithm
  - Schedulable rate assignment for all feasible task sets
- Has polynomial complexity
MC-Fluid Scheduling

- $\theta_i^H$ is determined by solving the convex optimization problem

\[
\begin{align*}
\text{minimize} & \quad \sum_{\tau_i \in \tau_H} (\theta_i^L - u_i^L) \\
\text{subject to} & \quad \sum_{\tau_i \in \tau_H} \theta_i^H \leq m \\
& \forall \tau_i \in \tau_H, \quad \theta_i^H \geq u_i^H \\
& \forall \tau_i \in \tau_H, \quad \theta_i^H \leq 1
\end{align*}
\]

- $\theta_i^L = \frac{u_i^L \cdot \theta_i^H}{\theta_i^H - u_i^H + u_i^L}$
MCF Scheduling

MCF: Simplified variant of MC-Fluid algorithm

- **Rate Assignment:**
  - For all HI tasks $\theta_i^H$ is given by $\frac{u_i^H}{\rho}$
  - $\rho = \max \{\text{normalized utilization}, \max \{u_i^H\}\}$
  - $\theta_i^L$ is computed same way as MC-Fluid
  - **Linear** run-time complexity
  - Compensates on schedulability
Outline

1. Introduction and Background
   - Mixed-Criticality (MC) System
   - Fluid Scheduling
   - Dual-rate MC Fluid Scheduling

2. Motivation
   - Challenges in Dual-rate MC Fluid Model

3. Proposed Strategy
   - MC-Sort algorithm
   - MC-Slope algorithm

4. Evaluation
   - Schedulability

5. Future Work
   - Multi-rate model
Challenges in Dual-rate MC Fluid Model

**Non-optimality**: Dual-rate fluid scheduling of MC task systems on multi-core is *not optimal*

- **Feasible** task sets are deemed to be *not schedulable*
  - Example: Multi-rate model
- We cannot extend MC-Fluid or MCF to multi-rate model
  - **Complexity** of MC-Fluid is high
  - MCF compromises on the schedulability
- **Solution**: Algorithm with better schedulability and reduced complexity
Outline

1. Introduction and Background
   - Mixed-Criticality (MC) System
   - Fluid Scheduling
   - Dual-rate MC Fluid Scheduling

2. Motivation
   - Challenges in Dual-rate MC Fluid Model

3. Proposed Strategy
   - MC-Sort algorithm
   - MC-Slope algorithm

4. Evaluation
   - Schedulability

5. Future Work
   - Multi-rate model
MC-Sort algorithm

MC-Sort:
- Maximum rate to a task with a larger HI utilization
- MC-Sort HI rate assignment
  - Assign initial rate of $\frac{u_i^H}{\rho_i}$
    - $\rho_i = \max \left\{ \left( \frac{u_i^H}{m} \right), u_i^H \right\}$
  - Sorts all HI tasks in decreasing HI utilization
  - Assigns maximum rate to tasks in the sorted order until slack remains
- **Linearithmic** complexity (i.e., $n \log n$)
MC-Sort algorithm

Example: $m = 2$

<table>
<thead>
<tr>
<th>Task</th>
<th>$T_i$</th>
<th>$u_i^L$</th>
<th>$u_i^H$</th>
<th>MC-Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$\theta_i^L$</td>
</tr>
<tr>
<td>$\tau_1$</td>
<td>5</td>
<td>0.3</td>
<td>0.9</td>
<td>-</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>7</td>
<td>0.4</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>35</td>
<td>0.1</td>
<td>0.3</td>
<td>-</td>
</tr>
<tr>
<td>$\tau_4$</td>
<td>35</td>
<td>0.45</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\sum$</td>
<td></td>
<td>1.25</td>
<td>1.7</td>
<td>-</td>
</tr>
</tbody>
</table>
MC-Sort algorithm

Example: \( m = 2 \)

<table>
<thead>
<tr>
<th>Task</th>
<th>( T_i )</th>
<th>( u_i^L )</th>
<th>( u_i^H )</th>
<th>( \theta_i^L )</th>
<th>( \theta_i^H )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_1 )</td>
<td>5</td>
<td>0.3</td>
<td>0.9</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( \tau_2 )</td>
<td>7</td>
<td>0.4</td>
<td>0.5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( \tau_3 )</td>
<td>35</td>
<td>0.1</td>
<td>0.3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( \tau_4 )</td>
<td>35</td>
<td>0.45</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>( \sum )</td>
<td>1.25</td>
<td>1.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- Sort all tasks with \( u_i^H \)
MC-Sort algorithm

Example: m = 2

<table>
<thead>
<tr>
<th>Task</th>
<th>$T_i$</th>
<th>$u_i^L$</th>
<th>$u_i^H$</th>
<th>MC-Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>5</td>
<td>0.3</td>
<td>0.9</td>
<td>- -</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>7</td>
<td>0.4</td>
<td>0.5</td>
<td>- -</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>35</td>
<td>0.1</td>
<td>0.3</td>
<td>- -</td>
</tr>
<tr>
<td>$\tau_4$</td>
<td>35</td>
<td>0.45</td>
<td>-</td>
<td>- -</td>
</tr>
<tr>
<td>$\sum$</td>
<td>1.25</td>
<td>1.7</td>
<td>-</td>
<td>- -</td>
</tr>
</tbody>
</table>

- Compute $\rho_i = \max \left\{ \left( \frac{u_i^H}{m} \right), u_i^H \right\}$
- $\rho_1 = 0.9$ $\rho_2 = 0.75$ $\rho_3 = 0.75$
**MC-Sort algorithm**

**Example: m = 2**

<table>
<thead>
<tr>
<th>Task</th>
<th>$T_i$</th>
<th>$u_i^L$</th>
<th>$u_i^H$</th>
<th>MC-Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>5</td>
<td>0.3</td>
<td>0.9</td>
<td>-</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>7</td>
<td>0.4</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>35</td>
<td>0.1</td>
<td>0.3</td>
<td>-</td>
</tr>
<tr>
<td>$\tau_4$</td>
<td>35</td>
<td>0.45</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$\sum$</td>
<td>1.25</td>
<td>1.7</td>
<td>-</td>
<td>1.96</td>
</tr>
</tbody>
</table>

- Initial assignment ($\frac{u_i^H}{\rho}$) is done
- Allocate remaining slack to task with maximum $u_i^H$
MC-Sort algorithm

Solution:

<table>
<thead>
<tr>
<th>Task</th>
<th>( T_i )</th>
<th>( u^L_i )</th>
<th>( u^H_i )</th>
<th>MC-Sort</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_1 )</td>
<td>5</td>
<td>0.3</td>
<td>0.9</td>
<td>( \theta^L_i ) 0.84 ( \theta^H_i ) 0.93</td>
</tr>
<tr>
<td>( \tau_2 )</td>
<td>7</td>
<td>0.4</td>
<td>0.5</td>
<td>( \theta^L_i ) 0.47 ( \theta^H_i ) 0.67</td>
</tr>
<tr>
<td>( \tau_3 )</td>
<td>35</td>
<td>0.1</td>
<td>0.3</td>
<td>( \theta^L_i ) 0.2 ( \theta^H_i ) 0.4</td>
</tr>
<tr>
<td>( \tau_4 )</td>
<td>35</td>
<td>0.45</td>
<td>-</td>
<td>( \theta^L_i ) 0.45 ( \theta^H_i ) -</td>
</tr>
<tr>
<td>( \sum )</td>
<td>1.25</td>
<td>1.7</td>
<td>1.96</td>
<td>2.0</td>
</tr>
</tbody>
</table>

\( \theta^L_i \) is computed same way as MC-Fluid
MC-Slope algorithm

- **MC-Sort limitation**: Does not consider the **difference in utilization** between criticality levels
  - Task that does maximum execution after mode switch may not get maximum rate allocation
MC-Slope algorithm

**MC-Slope: HI rate assignment**

- **Objective:** Minimize $\sum (\theta^L_i - u^L_i)$
- **Initial rate:** $\theta^H_i = u^H_i$
- **Sorts all HI tasks with** $R(\theta^H_i)$
  - $R(\theta^H_i) = \frac{d^2(\theta^L_i - u^L_i)}{d\theta^H_i}$
- **Assign maximum rate to task with larger** $R(\theta^H_i)$
- **Linearithmic complexity** *(i.e., nlogn)*
Introduction and Background
- Mixed-Criticality (MC) System
- Fluid Scheduling
- Dual-rate MC Fluid Scheduling

Motivation
- Challenges in Dual-rate MC Fluid Model

 Proposed Strategy
- MC-Sort algorithm
- MC-Slope algorithm

Evaluation
- Schedulability

Future Work
- Multi-rate model
Schedulability

![Graph showing acceptance ratio vs normalized utilization bound for different scheduling strategies.](image-url)
Outline

1. Introduction and Background
   - Mixed-Criticality (MC) System
   - Fluid Scheduling
   - Dual-rate MC Fluid Scheduling

2. Motivation
   - Challenges in Dual-rate MC Fluid Model

3. Proposed Strategy
   - MC-Sort algorithm
   - MC-Slope algorithm

4. Evaluation
   - Schedulability

5. Future Work
   - Multi-rate model
Multi-rate model: Each task executes with more than 2 rates
Future Work

Multi-rate model: Each task executes with more than 2 rates
Multi-rate model: Each task executes with more than 2 rates

- $\tau_1$, $\tau_2$, $\tau_3$ represent execution requirements.
- $C_1^H$, $C_2^H$, $C_3^H$ represent completion times.
- $\delta^L_i$, $\delta^H_i$, $\delta_H^*$ represent rates of jobs released in HI mode.
- Mode switch at $T_i$.
- Execution requirement is $\tau_i$.
**Future Work**

**Multi-rate model:** Each task executes with **more than 2 rates**

\[
\begin{align*}
&\delta^L_i, \delta^H_i \text{ - rate of jobs in LI and HI mode from mode switch until the earliest period of a carry-over, respectively} \\
&\delta^H_i \text{ - rate of jobs in HI mode from mode switch until the earliest period of a carry-over} \\
&\delta^C_i \text{ - rate of carry-overs for task } i \\
&\tau^1, \tau^2, \tau^3, \ldots \\
&\delta^H_1, \delta^H_2, \delta^H_3, \ldots
\end{align*}
\]
Future Work

Multi-rate model: Each task executes with more than 2 rates
Multi-rate model

Example: $m = 2$

<table>
<thead>
<tr>
<th>Task $T_i$</th>
<th>$u_i^L$</th>
<th>$u_i^H$</th>
<th>MC-Fluid</th>
<th>Multi-rate model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\theta_i^L$</td>
<td>$\theta_i^H$</td>
<td>$\delta_i^L$</td>
<td>$\delta_i^H*$</td>
</tr>
<tr>
<td>$\tau_1$</td>
<td>5</td>
<td>0.3</td>
<td>0.8</td>
<td>0.64</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>7</td>
<td>0.4</td>
<td>0.7</td>
<td>0.70</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>35</td>
<td>0.1</td>
<td>0.3</td>
<td>0.22</td>
</tr>
<tr>
<td>$\tau_4$</td>
<td>35</td>
<td>0.45</td>
<td>-</td>
<td>0.45</td>
</tr>
<tr>
<td>$\sum$</td>
<td>2.01</td>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
</tr>
</tbody>
</table>
Thank you..!

Questions..?