Fault-Tolerant Hierarchical Real-Time Scheduling with Backup Partitions on Single Processor

Hyun-Wook Jin
System Software Laboratory
Konkuk University
jinh@konkuk.ac.kr
http://sslab.konkuk.ac.kr
Contents

• Introduction
  – Motivation
  – Related work

• System Model
  – Basic system model
  – Fault model
  – Problem statement

• Schedulability Analysis of Fault-Tolerant Hierarchical Real-Time Scheduling
Introduction

- Hierarchical Real-Time Scheduling
  - Provides efficient composition of multi-threaded real-time applications
  - Partition (a.k.a server)
  - Global scheduling
  - Local scheduling
Vehicular CPS

• Partitioning Software Platforms
  – ARINC-653
  – AUTOSAR

• Aiming for
  – Temporal resource partitioning
    ▶ Meets real-time requirements
  – Transparent composition
    ▶ Resolves Size, Weight, and Power (SWaP) issues
  – User mode execution
    ▶ Guarantees fault isolation
Linux-based ARINC 653

- Mixture of Kernel and VMM-Level Partitioning
  - Low overhead and jitter
    - Kernel-level design
  - Software reusability
    - VMM-level design
  - Flexibility
What we are facing

• **New Operational Flight Program (OFP)**
  – High chance of crash

• **R/C Switch**
  – Switch to the manual control mode to keep the UAV alive
  – Need an expert

• **Disadvantages**
  – Additional payload
  – Cumbersome process
  – Money
What we are thinking

Primary Partition (Better Flight)  ZZZ…
Backup Partition (Stable Hovering)

Fault Detection

Primary Partition (Better Flight)  ZZZ…
Backup Partition (Stable Hovering)
Goals

- **Supporting Primary-Backup Partitions**
  - Primary partition: new (but yet to be verified)
  - Backup partition: safe (but inefficient)

- Handling unrecoverable software faults caused by the primary partition

- Extension of existing partition model
- Schedulability analysis
Related Work

• **Fault-Tolerant Real-Time Scheduling**
  – Liestman[1986], Burns[1996], Bertossi[1999], Han[2003], Yang[2004], Bertossi[2006], Cirinei[2007]
  – Mostly considers hardware-fault on multi-processor systems
  – No considerations for partitions

• **Fault-Tolerant Hierarchical Real-Time Scheduling**
  – Hyun[2012]
  – Considers a recovery job of a partition
Existing Partition Model (Shin[2003/2008])

• Scheduling Unit (i.e., Partition)
  – $S_i = S(W_i, \Gamma_i, A_i)$
  – $W_i$: workload = \{T_1, T_2, \ldots, T_3\}
    • Task $T_i = T(p_i, e_i)$
  – $\Gamma_i$: resource model = $\Gamma(\Pi_i, \Theta_i)$
    • $\Pi_i$: Period
    • $\Theta_i$: Supply time
  – $A_i$: scheduling algorithm = RM
Existing Partition Model (Shin[2003/2008])

- **Schedulability Analysis**
  - $\text{sbf}_{\Gamma}(t)$
    - Minimum resource supplies that resource $\Gamma$ can provide during time interval $t$
  - $\text{dbf}_{RM}(W_i, t)$
    - Maximum resource demand that workload $W$ can request during time interval $t$ under RM
  - Schedulable if $\forall t \; \text{sbf}_{\Gamma}(t) \geq \text{dbf}_{RM}(W_i, t)$
Extended Partition Model

• Scheduling Units
  – \{S_1, S_2, \ldots, S_{2k-1}, S_{2k}, \ldots, S_{n-1}, S_n\}
  – Primary partition: S_{2k-1}
  – Backup partition: S_{2k}

  • Context-Dependent Tasks (CDTs) = CDT(W_{2k})
  • Context-Independent Tasks (CITs) = CIT(W_{2k})
  • CDT(W_{2k}) \cup CIT(W_{2k}) = W_{2k}
Fault Model

• **Modes**
  – Primary mode
    • Primary partition + CDTs of backup partition
  – Recovery mode
    • A fault is detected
    • CITs have to be finished by $\Pi_{2k}$
  – Backup mode
    • Backup partition

• $\Gamma_i = \Gamma(\Pi_i, \Theta_{i,p}, \Theta_{i,b})$
  – $\Theta_{i,p}$: Supply time in primary mode
  – $\Theta_{i,b}$: Supply time in backup mode
Fault Model

- **Assumptions**
  - A fault is detected at the end of execution (overrun)
  - Maximum one fault can happen for $\Pi_{\text{max}}$
  - No returning to the primary mode
Problem Statement

- Schedulability Analysis for
  - CIT($W_{2k}$) during a recovery phase
  - Lower priority partitions right after the recovery phase
Schedulability Analysis

- \( \text{CIT}(W_{2k}) \) during a Recovery Phase

\[ V_{2k} = \Pi_{2k} - R_{2k} - B_{2k-2}(R_{2k}, \Pi_{2k}) \]

- \( R_{2k} \): Response time
- \( B_{2k-2} \): Busy time

\[ V_{2k} \geq \sum_{T_i \in \text{CIT}(W_{2k})} e_i \]
Schedulability Analysis

• Lower Priority Partitions after Recovery Phase

\[ \Pi_i - \left( R_{2k} + \sum_{T_j \in CIT(W_{2k})} e_j + B_i(R_{2k}, \Pi_i) \right) \geq 0, \text{ for } i > 2k \]
Schedulability Analysis

- **Primary Mode**
  - \( \Gamma_{2k-1} = \Gamma(\Pi_{2k-1}, \Theta_{2k-1}^p) \)
  - \( \Gamma_{2k} = \Gamma(\Pi_{2k}, \Theta_{2k}^p) \)

- **Backup Mode**
  - \( \Gamma_{2k} = \Gamma(\Pi_{2k}, \Theta_{2k}^b) \)

\[
\begin{align*}
    sbf_{\Gamma_{2k-1}}(t) & \geq dbf_{RM}(W_{2k-1}, t) \\
    sbf_{\Gamma_{2k}}(t) & \geq dbf_{RM}(CDT(W_{2k}), t)
\end{align*}
\]
Example

- **Workload**

<table>
<thead>
<tr>
<th>Partitions</th>
<th>Tasks</th>
<th>Period (ms)</th>
<th>Exec. Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₁</td>
<td>T₁</td>
<td>40</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>T₂</td>
<td>80</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>T₃</td>
<td>160</td>
<td>2</td>
</tr>
<tr>
<td>S₂</td>
<td>T₁ (CIT)</td>
<td>55</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>T₂ (CDT)</td>
<td>80</td>
<td>6</td>
</tr>
<tr>
<td>S₃</td>
<td>T₁</td>
<td>40</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>T₂</td>
<td>40</td>
<td>3</td>
</tr>
</tbody>
</table>

- **Resource Models**
  - $\Gamma_1 (5, 1.5, 0)$
  - $\Gamma_2 (15, 4, 5)$
  - $\Gamma_3 (20, 2, 0)$
  - $\Gamma_4 (20, 0, 0)$
Example

- **Schedulability of CIT**
  - $V_2 = 15 - 7 - 0 = 8$
  - $\sum_{\text{CIT}} e_j = 4$
  - $V_2 \geq \sum_{\text{CIT}} e_j$

- **Schedulability of low-priority partitions**
  - $B_3(7, 20) = 9$
  - $20 - (7 + 4 + 9) \geq 0$
Conclusions

• Extended the Hierarchical Real-Time Scheduling Model
  – Primary-backup partition
    • CDTs and CITs

• Provided Schedulability Analysis
  – Schedulability of CIT
  – Schedulability of low-priority partitions
Future Work

- **Relaxed Assumptions**
  - The fault may occur more than once during $\Pi_{\text{max}}$ time unit
  - Returning to the primary mode

- **Simulation**
  - Analyze more cases

- **Actual Implementation**
  - Apply to UGV and UAV (quadcopter)
Thanks!

Hyun-Wook (JIN) Jin
http://home.konkuk.ac.kr/~jinh
jinh@konkuk.ac.kr

System Software Laboratory
Konkuk University
http://sslab.konkuk.ac.kr