Empirical Study of Energy Minimization Issues for Mixed-Criticality Systems with Reliability Constraints

Zheng Li, Xiayu Hua, Chunhui Guo and Shangping Ren
Email: {zli80, xhua, cguo13}@hawk.iit.edu, ren@iit.edu

Illinois Institute of Technology

November 3, 2014

Outline

Introduction

Mixed-Criticality (MC) System

To reduce size, weight, power consumption and cost, tasks at different criticality levels share the same platform, e.g., Unmanned Aerial Vehicle (UAV).

- HI-criticality task: flight control
- LO-criticality task: photo capturing

MC Research Issues

- Schedulability: EDF-VD [Baruah, 2012], demand bound function analysis [Ekberg, 2012]
- Reliability: recovery [Zhao, 2012]
- Energy efficiency: Dynamic Voltage and Frequency Scaling (DVFS) [Zhu, 2004]

System Model

Processor Model

Processor: single DVFS-enabled processor

Available frequencies: $F_a = \{f_1, ..., f_q\}$ in descending order where $f_{\mathtt{max}} = f_1 = 1$, $f_{\mathtt{min}} = f_q$

Task Model

Mixed-criticality task: $\tau_i = (C_i, T_i, L_i)$

- $C_i = \{C_i(LO), C_i(HI)\}$: worst-case execution times under $f_{\mathtt{max}}$
- T_i: period, which is equal to deadline
- $L_i = \{LO, HI\}$: criticality level

HI-criticality task: $C_i(LO) \leq C_i(HI)$
LO-criticality task: $C_i(LO) = C_i(HI)$

System Model

Fault Recovery Model

Fault detection: at end of job execution, cost is counted in C_i

Backward recovery: re-executed under $f_{\mathtt{max}}$, deadline is same with the corresponding job

Reliability Model

Transient fault rate: $\lambda(f_i) = \hat{\lambda}_0 10^{-\hat{d} f_i}$

Job-level reliability: $R_i(f_i) = e^{-\lambda(f_i) \cdot \frac{C_i}{f_i}}$

Task-level reliability within a hyper-period H:
$\Phi_i(f_i, r_i, k_i) = R_i(f_i)_i^k + \sum_{j=1}^{r_i} \binom{k_i}{j} (1 - R_i(f_i))^j R_i(f_i)^{k_i - j}$ where r_i is the recovery number and $k_i = H/T_i$ is the number of jobs

System Model

Energy Model

Energy consumption of a job: $E(f_i, C_i) = P_{\mathtt{ind}} \frac{C_i}{f_i} + C_{\mathtt{ef}} C_i f_i^{\theta-1}$

Expected energy consumption of recovery: ignored, as probability of fault occurrence is very small (e.g., $\leq 10^{-6}$)

Expected energy consumption of the system within a hyper-period H: $EC(F) = \sum_{\tau_i \in \Gamma} (H/T_i) \cdot E(f_i, C_i)$ where $F = \{f_1, ..., f_i, ..., f_{|\Gamma|}\}$ is the frequency assignment of task set Γ

Assumption: all jobs of the same task have the same execution frequency

System Model

- The system has two running modes, i.e., LO-mode and HI-mode, and the system runs at the LO-mode initially.
- In the LO-mode, each task τ_i can run up to $\frac{C_i(LO)}{f_i}$ within each period. If any task τ_i executes beyond $\frac{C_i(LO)}{f_i}$ in a period, the system enters into the HI-mode.
- In HI-mode, all LO-criticality tasks are removed from the system and every HI-criticality task τ_i's maximum execution time is $\frac{C_i(HI)}{f_{\mathtt{max}}}$.
- It is worth pointing out that fault recovery time is not counted toward task execution time for triggering system mode switch.

Problem Formulation

Problem Formulation

Given a DVFS-enabled processor with q different processing frequencies $F_a = \{f_1, \cdots, f_q\}$ and a mixed-criticality task set $\Gamma = \{\Gamma_{LO}, \Gamma_{HI}\}$, develop an algorithm to **assign an execution frequency** to every task so that the system's expected energy consumption is minimized and the system's schedulability constraint and reliability requirement are guaranteed.

Reliability requirement: $\forall 1 \leq i \leq |\Gamma| \quad \Phi_i(f_i, r_i, k_i) \geq \Phi_i(f_{\max}, 0, k_i)$

Assumption: task execution frequency does NOT change once it is assigned.

Strategy

- LO-mode: tasks run at lower frequencies
- HI-mode: all HI-criticality tasks run at f_{max}

Energy Consumption Analysis

- Lower frequency reduces system's energy consumption.
- Lower frequency results in longer execution time and lower reliability, the system needs more recoveries to maintain reliability requirement. Both longer execution time and more recoveries negatively impact schedulability.
- There is a tradeoff between energy consumption and schedulability and reliability constraints.

Reliability Analysis

- In the LO-mode, at least r_i recoveries are needed to maintain reliability requirement under frequency f_i. Given the reliability requirement, we can calculate r_i.
- In the HI-mode, as the system runs under $f_{\mathtt{max}}$, no recovery is needed.

Schedulability Analysis

Schedulability Condition

$$\forall l \in [0, H] : \sum_{\tau_i \in \Gamma} \text{dbf}_{\text{LO}}(\tau_i, l) \leq l$$

$$\forall l \in [0, H] : \sum_{\tau_i \in \Gamma_H} \text{dbf}_{\text{HI}}(\tau_i, l) \leq l$$

Worst Case

r_i recoveries take place in the first r_i jobs of τ_i

Virtual Deadline

HI-criticality task: $\mathtt{VD}_i \leq T_i$
LO-criticality task: $\mathtt{VD}_i = T_i$

Schedulability Analysis

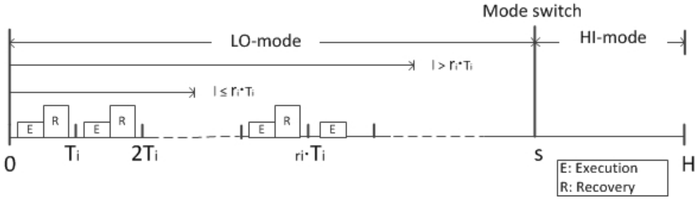

Figure: LO-mode Job Execution

LO-mode Demand Bound Function

$$\mathrm{dbf}_{\mathrm{LO}}(\tau_i, l, f_i) = \begin{cases} [\![(\lfloor \frac{l - VD_i}{T_i} \rfloor + 1) \cdot (1 + \frac{f_{max}}{f_i}) \cdot C_i(LO)]\!]_0 & \text{if } l \leq r_i \cdot T_i \\ r_i \cdot (1 + \frac{f_{max}}{f_i}) \cdot C_i(LO) + & \\ [\![\frac{f_{max}}{f_i} \cdot (\lfloor \frac{l - r_i \cdot T_i - VD_i}{T_i} \rfloor + 1) \cdot C_i(LO)]\!]_0 & \text{otherwise} \end{cases}$$

where $[\![x]\!]_0 = \max\{x, 0\}$

Schedulability Analysis

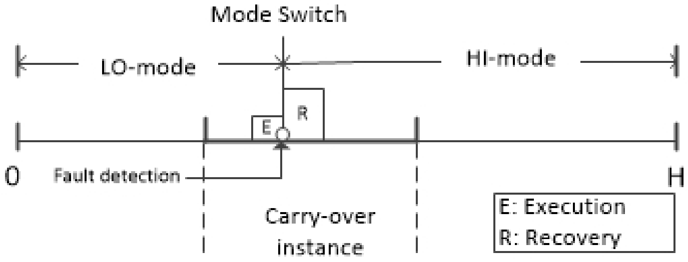

Figure: Carry-over Job

HI-mode Demand Bound Function

$$\mathrm{dbf}_{\mathrm{HI}}(\tau_i, l) = [\![(\lfloor \frac{l - (T_i - \mathrm{VD}_i)}{T_i} \rfloor + 1) \cdot C_i(HI)]\!]_0$$

Algorithm

Heuristic Search based Energy Minimization (HSEM) Algorithm

1. Initially, all the tasks are running under the f_{max}.
2. Heuristically select a task to scale down its frequency without violating the reliability and schedulability constraints.
3. Repeat step (2) until no task in step (2) is available.

Schedulability Test

Use GREEDY algorithm [Ekberg, 2012] to determine virtual deadlines and test schedulability.

Metric (Heuristic Criteria)

The ratio between energy consumption change and execution time demand change

$$\mathtt{ED}(F_i, F_i') = \frac{\mathtt{EC}(F) - \mathtt{EC}(F')}{\mathtt{Gap}(F) - \mathtt{Gap}(F')}$$

where $F_i = \{f_1, ..., f_i, ..., f_{|\Gamma|}\}$

$F_i' = \{f_1, ..., f_i', ..., f_{|\Gamma|}\}$

$$\mathtt{Gap}(F) = \min_{l \in [0,H]} \{g(l) | g(l) \neq l\}$$

$$g(l) = l - \sum_{\tau_i \in \Gamma} \mathtt{dbf}_{\mathtt{LO}}(\tau_i, l)$$

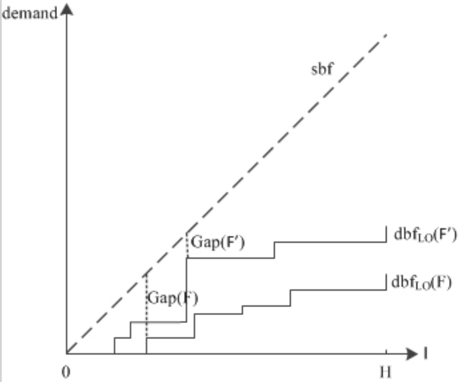

Figure: Demand Bound Function

Algorithm

Task Selection Policy (TSP)

$$\mathtt{ED} = \begin{cases} \max\limits_{i \in [1,|\Gamma|]} \{\mathtt{ED}(F_i, F_i') | \mathtt{ED}(F_i, F_i') < 0\} & \text{if } \exists \mathtt{ED}(F_i, F_i') < 0 \\ \max\limits_{i \in [1,|\Gamma|]} \{\mathtt{ED}(F_i, F_i')\} & \text{otherwise} \end{cases}$$

Algorithm

ALGORITHM 1: HSEM$(\Gamma, F = \{f_{\mathtt{max}}, \ldots, f_{\mathtt{min}}\})$

int$[|\Gamma|]$ $FI = \{0, .., 0\}$;
while $\exists i : FI[i] < |F|$ **do**
    find
    $I = \{i | \mathtt{CHECK}(\Gamma, F, FI, i) == \mathtt{TRUE} \wedge 0 \leq i \leq |\Gamma| - 1\}$
    if *I is not empty* **then**
        find τ_k based on TSP
        $FI[k] = FI[k] + 1$;
    end
    else
        break;
    end
end
return FI;

Algorithm

ALGORITHM 2: CHECK (Γ, F, FI, i)

int[$|\Gamma|$] A={0,...,0};
$FI[i] = FI[i] + 1$
for *(j = 0; j < $|\Gamma|$; i++)* **do**
    A[j] = Obtain from MRT with $F[FI[j]]$ and MRT;
end
if *(GREEDY(Γ, F, FI, A) == SUCCESS)* **then**
    return TRUE;
end
return FALSE;

Performance Evaluation

Comparisons

- LUF (Largest Utilization First): chooses the task with **largest** HI-mode utilization to scale down its execution frequency as low as it does not violate the reliability and schedulability constraints.
- SUF (Smallest Utilization First): chooses the task with **smallest** HI-mode utilization to scale down its execution frequency as low as it does not violate the reliability and schedulability constraints.

Performance Evaluation

Experiment Setting

3 LO-criticality tasks, 3 HI-criticality tasks

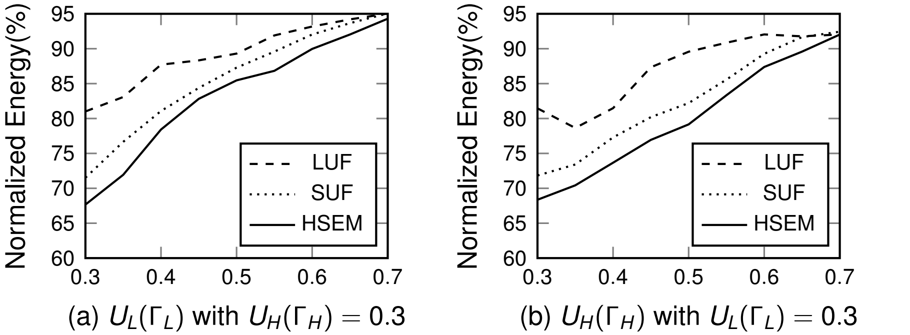

Figure: Normalized Energy Consumption (%)

Conclusion

- Analyze the resource demand of a mixed-criticality task set with both deadline and reliability constraints under a given frequency assignment.
- Propose HSEM algorithm to find a frequency assignment which can minimize system's energy consumption without violating system's reliability and schedulability constraints.
- Compared with LUF and SUF, the HSEM algorithm performs better with respect to energy saving.
- Future work is to extend our developed demand analysis and algorithm to address the problem that recovery is needed even under $f_{\mathtt{max}}$.

Questions?

Thank You