Dynamic Scheduling of Real-Time Tasks

Joakim, Jakob, Fredrik, Mattias

2019-10-23

Introduction

- Scheduling tasks
- Periodic, independent
- Sporadic, unpredictable
- Accept or block task
- Single processor machine.

Task model (periodic aperiodic)

Periodic

- For periodic tasks J a given task is defined by the following.
  $J = \{T_i(s_i, C_i, R_i, P_i), i = 1\ to\ n\}$
- $\sum \frac{C_i}{P_i} \leq 1$

Sporadic

- For sporadic tasks $\mathbb{S}$ a given task is defined by the following.
  $\mathbb{S}\ = \{S_i(r_i, C_i, d_i), i = 1\ to\ n\}$
- $S_i \to S_j$

Scheduling independent tasks

Periodic

$$\sum_{i=1} \frac{C_i}{R_i} \leq 1$$

Sporadic

$$\sum_{r_k \leq r_1,\, d_k \leq d_j} C_k \leq d_j - r_i$$

Synchronous tasks

$[0, P]$ where $P =$ least common multiple of $\{P_1, P_2 \ldots P_n\}$ ## Asynchronous tasks $[0, s + 2P]$ where $s = \max\{s_1, s_2 \ldots s_n\}$ ## Sporadic tasks $[\tau, D + P]$ where $\tau =$ is the current time of the machine and $D =$ the latest sporadic deadline supported at time τ # Scheduling dependent tasks

Dependent tasks

- Group of sporadic in partial order
- Timing and precedence constraint

New results about dependent task scheduling

- Dependent and independent tasks
- No discrimination
- Constraints are obeyed
- Proof for all deadlines

New results about dependent task scheduling

Modified deadline

$$f_i \leq f_j \text{ and } f_i \leq d_j - C_j$$
$$\text{Set } d_i^* := \min(d_i, \min(d_j^* - C_j; S_i \rightarrow S_j))$$

Modified release time

$$k_i \geq k_j + C_j \text{ and } k_i \geq r_i$$
$$\text{Set } r_i^* := \max(r_i, \max(r_j^* + C_j; S_j \rightarrow S_i))$$

Variables

s	Task
C	Completion time
f	Completion time of S_i
d	Deadline
k	Starting time of S_i
r	Release time

New results about dependent task scheduling

Apply ED

- $r_i^* < r_j^*, d_i^* < d_j^*$ where $S_i < S_j$
- Apply Earliest Deadline
- $\mathcal{L}_r$ so at any time t, $S_i \in \mathcal{L}_r$, if $r_i^* \geq t$

Conclusion of algorithm

- Schedulability of $\mathbb{S}$ implies scheduability of $\mathbb{S}^*$
- Timing and precedence constraints met according to ED

Decision Algorithm

1. Define global variables.

τ current time

P period

$\mathbb{S}^\tau$ linked list of sporadic tasks at time τ

2. Calculate new timing parameters.

$$\mathbb{S}^\tau = \text{calc\_readytimes\_and\_deadlines}(\mathbb{S}^\tau)$$

Decision Algorithm

3. Initialize data structures.

$$d = \min(\mathbb{S}^{\tau}.d^{*})$$
$$D = \max(\mathbb{S}^{\tau}.d^{*})$$
$$\mathcal{L}^{*} = \{\ \}$$
$$\mathbb{S}^{*} = \text{sort\_deadline}(\text{filter}(D + P,\ \mathbb{S}^{\tau}))$$
$$q = \text{index\_of}(d,\ \mathbb{S}^{*})$$

Same sum as before

$$\sum_{r_k \leq r_i,\, d_k \leq d_j} C_k \leq d_j - r_i$$

4. Acceptance condition.

$$\text{for } j \in [q, \text{sizeof}(\mathbb{S}^{*})]$$
$$k = \text{index\_where}(r_j^{*} < r_k^{*},\ \mathcal{L}^{*})$$
$$\mathbb{S}_{k-1}^{*} = \mathbb{S}_j^{*}$$
$$\mathcal{L}_q^{*} = \mathbb{S}_j^{*}$$
$$N_{k,\, j} = 0$$
$$\text{for } i \in [q, 0] :$$
$$N_{i,\, j} = N_{i+1,\, j} + C_i$$
$$\text{if } N_{i,\, j} > d_j^{*} - r_i^{*} : \text{Return false}$$

Conclusion

- Calculating r^* and d^* for all S makes all tasks sporadic.
- If an incoming sporadic task group $\mathbb{S}^\tau$ completes the acceptance condition without breaking the loop, it is accepted and scheduled according to $\mathbb{S}^*$.
- In essence, this is ED.