Paper Presentation

Amo Guangmo Tong

University of Taxes at Dallas

gxt140030@utdallas.edu

February 11, 2014
Overview

1. Techniques for Multiprocessor Global Schedulability Analysis

2. New Response Time Bounds for Fixed Priority Multiprocessor Scheduling
Techniques for Multiprocessor Global Schedulability Analysis

Sanjoy Baruah
Hard real-time system $T$

- $n$ sporadic tasks $T_n = (e_n, D_n, p_n)$
- $m$ identical processors, $m > 1$
- minimum inter-arrival time or period, $p_i > 0$
- execution time $e_i < p_i$
- relative deadline $D_i \leq p_i$ (constrained)
- utilization $u_i = e_i / p_i$
Partitioned FJP (fixed job priority) algorithms
Assign tasks to processors. Schedule tasks on each uniprocessor.

Example:

\[ \tau_1(1,2) \rightarrow M_1 \]
\[ \tau_2(1,2) \rightarrow M_2 \]
\[ \tau_3(2,2) \rightarrow M_1 \]

Global FJP algorithms
In this schedule algorithm, tasks can execute and resume on any processors.

Scheduleable and Schedulability test
Contributions

- Global FJP and partitioned FJP are incomparable.
- Provide a schedulability test for global EDF.
Global and partitioned FJP scheduling are incomparable.

Step 1

Sometimes global FJP scheduling is better than partitioned FJP scheduling.

Example:

- \( \tau_1 = (2, 3), \tau_2 = (3, 4), \tau_3 = (5, 12); u_1 = 2/3, u_2 = 3/4, u_3 = 5/12. \)
- Because \( u_1 + u_2 > 1, u_2 + u_3 > 1, u_1 + u_3 > 1, \) partitioned FJP does not work.
- The following figure shows global FJP works.
Global and partitioned FJP scheduling are incomparable.

**Step 2**

Sometimes **partitioned FJP scheduling** is better than global FJP scheduling.

**Example:**

- $\tau_1 = (2, 3), \tau_2 = (3, 4), \tau_3 = (3, 12), \tau_4 = (4, 12); u_1 = 2/3, u_2 = 3/4, u_3 = 3/12, u_4 = 4/12$. Because $u_1 + u_4 = 1, u_2 + u_3 = 1$, partitioned FJP works.
- However no global FJP works. (If we assume $\tau_{3,1}$ has higher priority than $\tau_{4,1}$, then $\tau_{4,1}$ misses its deadline.)
Let focus on $\tau_{i,j}$.

- $r_{i,j} = t_a$, $d_{i,j} = t_d$, $D_i = t_d - t_a$, $t_o$ is the latest time instant before $t_a$ at which at least one processor is idle.
- $L$: the union of intervals in $[t_a, t_d)$ during which all $m$ processors are executing jobs other than $\tau_{i,j}$.
If $\tau_{i,j}$ misses its deadline, then $|L| > D_i - e_i$. So there exists an $L^* \subseteq L$ where $|L^*| = D_i - e_i$. And the total execution of all tasks done in $[t_o, t_a) \cup L^*$ is $m(A_i + D_i - e_i)$.
A schedulability test for global EDF

- Let $I(\tau_k, \tau_i, A_i)$ denote the **upper bound** of the workload of $\tau_k$ that **could be done in** $[t_0, t_a) \cup L'$ where $L'$ is any subset of $[t_a, t_d)$ with length $D_i - e_i$.

$$I(\tau_k, \tau_i, A_i) = I(e_k, D_k, A_i, e_i, D_i)$$

- If, for any value of $A_i$, $\sum_{k=1}^{n} I(\tau_k, \tau_i, A_i) < m(A_i + D_i - e_i)$, then any job $\tau_{i,j}$ of $\tau_i$ cannot miss its deadline.
If for any value $A$,
\[\sum_{k=1}^{n} I(\tau_k, \tau_1, A) < m(A + D_1 - e_1),\] jobs in $\tau_1$ will meet deadline.
\[\sum_{k=1}^{n} I(\tau_k, \tau_2, A) < m(A + D_2 - e_2),\] jobs in $\tau_2$ will meet deadline.
\[\vdots\]
\[\sum_{k=1}^{n} I(\tau_k, \tau_n, A) < m(A + D_n - e_n),\] jobs in $\tau_n$ will meet deadline.

Thus, if for all $i$ from 1 to $n$, and any value $A_i$,
\[\sum_{k=1}^{n} I(\tau_k, \tau_i, A_i) < m(A_i + D_i - e_i),\] then no job in $\tau$ will miss deadline.
$I(\tau_k, \tau_i)$

$I(\tau_k, \tau_i, A_i)$ denote the **upper bound** of the workload of $\tau_k$ that **could be done in** $[t_0, t_a) \cup L^*$ where $L^*$ is any subset of $[t_a, t_d)$ with length $D_i - e_i$.

$I(\tau_k, \tau_i, A_i) = I(e_k, D_k, A_i, e_i, D_i)$
Pseudo-polynomial

- Previous: \( O(P(n)) \)
- Now: \( O(P(m, e_i, d_i)) \)

Other weakness of improvement.
New Response Time Bounds for Fixed Priority Multiprocessor Scheduling

Nan Guan, Martin Stigge, Wang Yi and Ge Yu
sporadic real-time system $T$

- $n$ sporadic tasks $T_n = (e_n, d_n, p_n)$
- $m$ identical processors, $m > 1$
- minimum inter-arrival time or period, $p_i > 0$
- execution time $e_i < p_i$
- relative deadline $D_i$
  - $D_i \leq p_i$ (constrained-deadline)
  - $D_i$ and $P_i$ have no relationship (arbitrary-deadline)
- utilization $u_i = e_i / p_i$
- $f_{i,j}$ denotes the finish time of $\tau_{i,j}$, define response time of $\tau_{i,j}$ as $f_{i,j} - r_{i,j}$
Content

- Response time bound for constrained-deadline system
- Response time bound for arbitrary-deadline system
Constrained-deadline system

To analyze deadline miss

Some proc. Is idle

$A_i \quad D_i$

$|L^*| = D_i - e_i$

$t_0 \quad t_a \quad t_d$

To analyze response time bound

Some proc. Is idle

$A_i \quad D_i$

$t_0 \quad t_a \quad t_f = f_{i,j}$
Constrained-deadline system

When we analyze $\tau_{i,j}$:
Let $\Omega_i(x)$ be the maximum value of the sum of all higher-priority tasks’ interference among all possible cases.

\[
\begin{align*}
& \Omega_i(x) \text{ be the maximum value of the sum of all higher-priority tasks’ interference among all possible cases.}
\end{align*}
\]
Then we solve the equation:

\[
x = \left\lfloor \frac{\Omega_i(x)}{m} \right\rfloor + e_i
\]

Remember: \(\Omega_i(x)\) be the maximum value of the sum of all higher-priority tasks’ interference against \(\tau_{i,j}\).
Then $x$ is an upper bound of response time of $\tau_{i,j}$.
Arbitrary-deadline system

For constrained-deadline system, we analyze:

For arbitrary-deadline system, we analyze:
Similarly, we solve the equation:

\[ x = \left\lceil \frac{\Omega_i(x,h)}{m} \right\rceil + h \cdot e_i \]

Remember: \( \Omega_i(x, h) \) be the maximum value of the sum of all higher-priority tasks’ interference against \( \tau_{i,j} \).
Then an upper bound of response time of $\tau_{i,j}$ is

$$x - (h - 1) \cdot p_i$$
Weakness or improvement
The End