Schedulability Analysis for Dual Priority Scheduling

I. Dual Priority

Theorem 1: τ_i is schedulable on a single processor using dual priority scheduling if for each absolute deadline of a job $d_{i,x}$ where $x \in N$, there exists a t' where $t - D_i \leq t' \leq t$ $(= d_{i,x})$ for which the following condition holds:

$$C_i + F_i(\tau, t', t) \leq t' \tag{1}$$

where $F_i(\tau, t')$ denotes the maximum possible execution resource consumed by the other jobs released by tasks in the system except $J_{i,x}$ during $[0, t']$

Proof 1: We can prove the statement that if τ_i is not schedulable, then Equation 1 would not hold. Suppose that τ_i is not schedulable, then there are legal event sequences in which deadline $d_{i,x}$ is missed when τ_i is assigned with the current priority and promotion point. Let S' denote such a sequence where $J_{i,x}$ misses deadline at the earliest time at t (i.e., $d_{i,x} = t$).

It must be the cases that during $[0, t]$ some other jobs are executing because otherwise the time interval between the last idle instant to t could also construct such a sequence. Then it must be that during the time interval $[0, t']$, other jobs have consumed an amount of resource more than

$$F_i(\tau, t', t) > t' - C_i$$

which contradicts the Equation 1.

Therefore a system is schedulable on a single processor using dual priority scheduling if and only if all the tasks in the system meet the requirement in Theorem 1, and hence we can derive the following theorem.

Theorem 2: A system τ is schedulable by dual priority scheduling if the following condition holds: $\forall\ \tau_i \in \tau :\ \forall t \geq D_i :\ \exists t' \in [t - D_i, t]$ so that

$$C_i + F_i(\tau, t', t) \leq t'$$

Unfortunately, it is very hard to know the exact value of $F_i(\tau, t', t)$, and hence we choose to derive an upper bound by considering each task separately. Let $f_i(\tau_j, t', t)$ denotes the maximum possible resource consumed by τ_j during $[0, t']$ in the scenario when all other tasks do not release any jobs (except $J_{i,x}$). In this case, the execution τ_j is independent of interference from other tasks, and hence

$$F_i(\tau, t') \leq \sum_{\tau_j \in \tau \setminus \tau_i} f_i(\tau_j, t', t) + \lfloor \frac{t - D_i}{T_i} \rfloor \times C_i$$

where $\lfloor \frac{t-D_i}{T_i} \rfloor \times C_i$ upper bounds the execution of τ_i itself before $J_{i,x}$. .

II. Interference Bound Function

When calculating $f_i(\tau_j, t', t)$, we ignore the interference from the other tasks on τ_j except $J_{i,x}$. Thus we have the following lemma.

Lemma 1 (Maximum Execution): $f_i(\tau_j, t', t)$ is maximized when $J_{j,1}$ is released at 0 and all jobs are released as soon as possible with period T_j, and each job of τ_j executes as early as possible as long as it has higher priority than $J_{i,x}$.

Proof 2: When $J_{j,1}$ is released at 0 and all jobs are released as soon as possible with period T_j, the number of jobs that released during $[0, t']$ is maximized. Meanwhile as we either shift the release pattern left or right, the total execution would only decrease or stay the same.

- $J_{j,y}$ is the job that has its release time $r_{i,y} \leq t' < r_{i,y} + T_j$. Since $r_{j,1} = 0$, we know $r_{j,y} = \lfloor \frac{t'}{T_j} \rfloor \times T_j$
- $[a]_0 = \max(a, 0)$

With Lemma 1, we can easily calculate $f_i(\tau_j, t', t)$, and in the following section, we present the detailed equations.

A. $f_i(\tau_j, t', t)$ when $j < i$

We first consider the case when index j is smaller than index i which means τ_j has higher origin priority than τ_i. Thus if $J_{j,y}$ is still active, $J_{i,x}$ would not execute unless it is during the interval $[P_{i,x}, P_{j,y}]$ (if $P_{i,x} < P_{j,y}$).

Case 1 ($P_{j,y} \leq P_{i,x}$) as shown in Figure 1: Job $J_{i,x}$ always has lower priority than $J_{j,y}$. Thus the maximum possible resource consumed by τ_j before t' is bounded by

$$f_i(\tau_j, t') = \lfloor \frac{t'}{T_j} \rfloor . C_j + \min\left(C_j, t' - r_{j,y}\right)$$

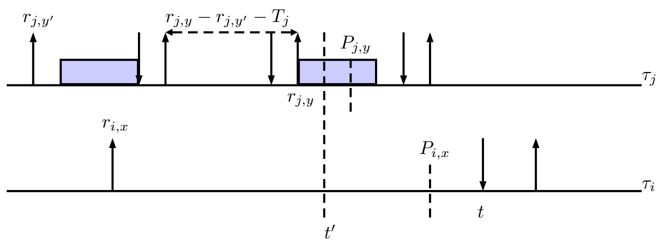

Fig. 1. $P_{j,y} \leq P_{i,x}$

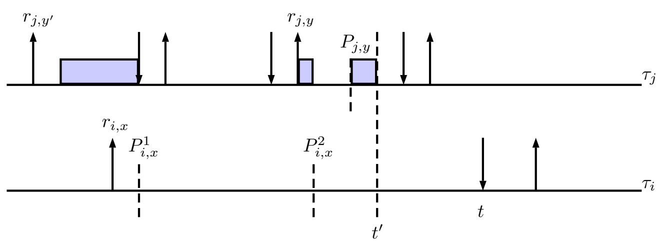

Fig. 2. $P_{j,y} > P_{i,x}$

Case 2 ($P_{j,y} > P_{i,x}$): as shown in Figure 2, $J_{i,x}$ has higher priority than $J_{j,y}$'s during $[\max(r_{j,y}, P_{i,x}), P_{j,y}]$ (if $P_{i,x} \leq P_{j,y}$). Thus $J_{j,y}$ can not execute during $[\max(r_{j,y}, P_{i,x}), P_j]$ unless $J_{i,x}$ has already finished. Thus we have

$$f_i(\tau_j, t') = \lfloor \frac{t'}{T_j} \rfloor . C_j +$$
$$\min\left(C_j, t' - r_{j,y} - (\min(t', P_{j,y}) - \max(r_{j,y}, P_{i,x}))\right)$$

B. $f_i(\tau_j, t')$ *when* $j > i$

Case 1.1 ($P_{i,x} \leq P_{j,y} \wedge r_{j,y} \leq r_{i,x}$): as shown in Figure 3, τ_j may execute during $[r_{j,y}, r_{i,x}]$, but τ_j would not execute after $r_{i,x}$ unless τ_i has finished.

$$f_i(\tau_j, t') = \lfloor \frac{t'}{T_j} \rfloor \times C_j + \min(C_j, r_{i,x} - r_{j,y})$$

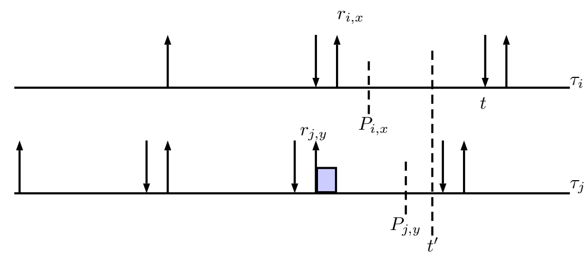

Fig. 3. $P_i \leq P_j \wedge r_j \leq r_i$

Case 1.2 ($P_{i,x} \leq P_{j,y} \wedge r_{j,y} > r_{i,x}$): as shown in Figure 4, $J_{j,y}$ would not execute unless τ_i has completed. Thus we have

$$f_i(\tau_j, t') = \lfloor \frac{t'}{T_j} \rfloor \times C_j$$

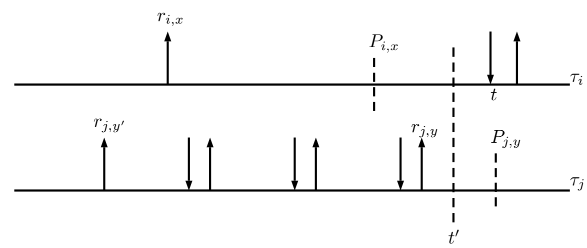

Fig. 4. $P_{i,x} \leq P_{j,y} \wedge r_{j,y} > r_{i,x}$

Case 2.1 ($P_{i,x} > P_{j,y} \wedge r_{j,y} \leq r_{i,x}$): as shown in Figure 5, after $r_{i,x}$, $J_{j,y}$ can only execute during $[\max(r_{i,x}, P_{j,y}), \min(t', P_{i,x})]$

$$f_i(\tau_j, t') = \lfloor \frac{t'}{T_j} \rfloor \times C_j +$$
$$\min\left(C_j, r_{i,x} - r_{j,y} + [\min(t', P_{i,x}) - \max(P_{j,y}, r_{i,x})]_0\right)$$

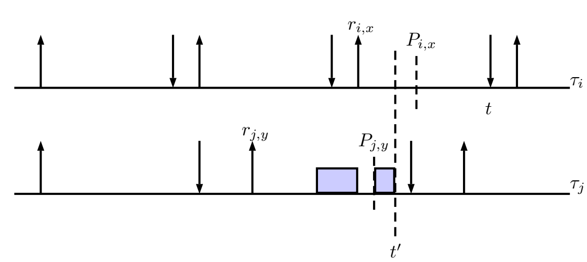

Fig. 5. $P_{i,x} > P_{j,y} \wedge r_{j,y} \leq r_{i,x}$

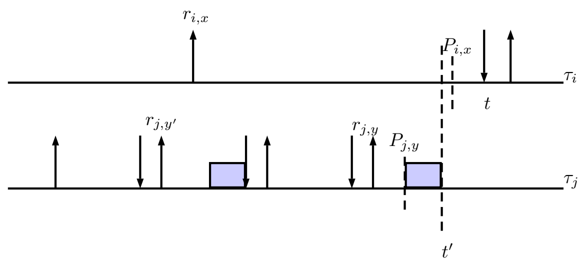

Fig. 6. $P_{i,x} > P_{j,y} \wedge r_{j,y} > r_{i,x}$

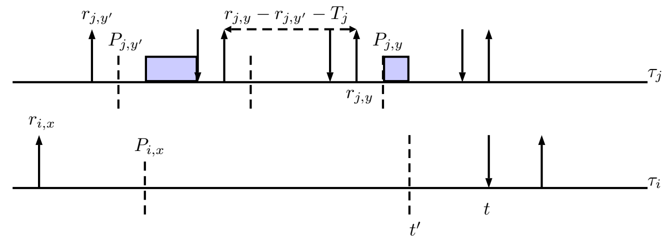

Fig. 7. 1

Case 2.2 ($P_{i,x} > P_{j,y} \wedge r_{j,y} > r_{i,x}$): as shown in Figure 6, $J_{j,y}$ would only execute during $[P_{j,y}, P_{i,x}]$ before $J_{i,x}$ finishes. Thus we have

$$f_i(\tau_j, t') = \lfloor \frac{t'}{T_j} \rfloor . C_j + \min\left(C_j, [\min(t', P_{i,x}) - P_{j,y}]_0\right)$$

III. Optimization Technique

Since the total exectuion happens before $P_{i,x}$ could not exceed $P_{i,x}$, we will introduce an optimization technique based on this fact. Since $f_i(\tau_j, t', t)$ denotes the maximum resource used by τ_j during $[0, t']$, and here we denfine another function $g_i(\tau_j, t')$ which denotes the maximum possible resource used by τ_j during $[P_{i,x}, t']$ (only if $P_{i,x} < t'$).

Lemma 2: If $D_j - p_j \geq C_j$, then $g_i(\tau_j, t')$ maximizes when $r_{j,y'} = [P_{i,x} + C_j - D_j]_0$ and $r_{j,y} = r_{j,y'} + \lfloor \frac{t' - r_{j,y'}}{T_j} \rfloor T_j$, as shown in Figure 7.

Proof 3: Case 1: If $P_{i,x} + C_j - D_j \geq 0 \Rightarrow r_{j,y'} = P_{i,x} + C_j - D_j$, (scenario in Figure 7) as we shift the pattern left, $J_{j,y'}$ execution after $P_{i,x}$ will decrease linearly up to C_j, while execution of $J_{j,y}$ will increase at most linearly. On the other hand, as we shift the pattern right, $g_i(\tau_j, t')$ would only decrease or stay the same.

Case 2 :If $P_{i,x} + C_j - D_j < 0 \Rightarrow r_{j,y'} = 0$, as we shift the pattern left, execution of $J_{j,y'}$ decreases from C_j to 0, while the increase of $J_{j,y}$ is bounded by C_j. On the other hand, as we shift the pattern right, $g_i(\tau_j, t', t)$ would only decrease or stay the same.

With the above lemma, we have

$$g_i(\tau_j, t', t) = \begin{cases} \min(C_j, t' - P_{i,x}) & \text{if } r_{j,y} = r_{i,y'} \\ C_j + \frac{r_{j,y} - r_{j,y'} - T_j}{T_j} C_j & \\ + \min(C_j, [t' - P_{j,y]_0}) & \text{otherwise} \end{cases} \quad (2)$$

Lemma 3: If $D_j - p_j < C_j$, then $g_i(\tau_j, t')$ maximizes when $r_{j,y'} = [P_{i,x} - p_j]_0$ and $r_{j,y} = r_{j,y'} + \lfloor \frac{t' - r_{j,y'}}{T_j} \rfloor T_j$, as shown in Figure 8.

Proof 4: Each job of J_y could execute at most $D_j - p_j$ time units after $P_{i,x}$ before $J_{i,x}$ finishes. Case 1: If $P_{i,x} - p_j \geq 0 \Rightarrow r_{j,y'} = P_{i,x} - p_j$ (scenario in Figure 7) as we shift the pattern left, $J_{j,y'}$ execution after $P_{i,x}$ will decrease linearly up to $D_j - p_j$, while execution of $J_{j,y}$ will increase at most linearly. On the other hand, as we shift the pattern right, $g_i(\tau_j, t', t)$ would only decrease or stay the same.

Case 2 :If $P_{i,x} - p_j < 0 \Rightarrow r_{j,y'} = 0$, as we shift the pattern left, execution of $J_{j,y'}$ decreases from $D_j - p_j$ to 0, while the increase of $J_{j,y}$ is bounded by $D_j - p_j$. On the other hand, as we shift the pattern right, $g_i(\tau_j, t', t)$ would only decrease or stay the same.

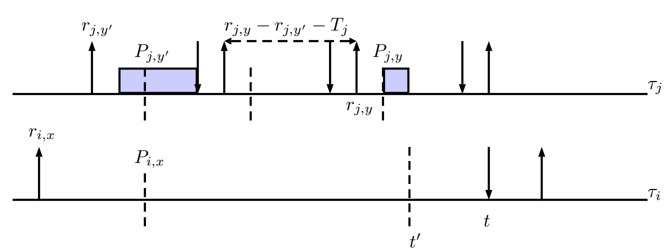

Fig. 8. 1

With the above lemma, we have

$$g_i(\tau_j, t', t) = \begin{cases} \min(D_j - p_j, t' - P_{i,x}) & \text{if } r_{j,y} = r_{i,y'} \\ D_j - p_j + \frac{r_{j,y} - r_{j,y'} - T_j}{T_j}(D_j - p_j) & \\ + \min(D_j - p_j, [t' - P_{j,y}]_0) & \text{otherwise} \end{cases}$$

Since total execution before $P_{i,x}$ (if $t' > P_{i,x}$) is bounded by $P_{i,x}$, we can use a simple optimization technique to further tighten the test. For τ_i itself, its maximum possible execution after $P_{i,x}$ is C_j. Therefore we have

$$F_i(\tau, t', t) = \begin{cases} \sum\limits_{\tau_j \in \{\tau \setminus \tau_i\}} f_i(\tau_j, t', t) + \lfloor \frac{t - D_i}{T_i} \rfloor C_i & \text{If } t' \le P_{i,x} \\ \min\left(P_{i,x}, \lfloor \frac{t - D_i}{T_i} \rfloor C_i + \sum\limits_{\tau_j \in \{\tau \setminus \tau_i\}} f_i(\tau_j, t', t) - g_i(\tau_j, t', t) \right) & \\ + \sum_{\tau_j \in \{\tau \setminus \tau_i\}} g_i(\tau_j, t', t) & \text{Otherwise} \end{cases} \tag{3}$$

We also need to derive an upper bound of t because otherwise t can tends to infinity. Suppose there exists a t so that

$$\forall\ t' \in (t - D_i, t] : \ F_i(\tau, t, t') + C_i > t'$$
$$\Rightarrow \min_{t' \in (t - D_i, t]} \frac{F(\tau_i, t, t') + C_i}{t'} > 1$$

, and let

$$\begin{aligned} H_i(t) =& U \times t + \sum_{\tau_j \in \{\tau \setminus \tau_i\}} C_j + C_i \\ & \ge t \times u_i + \sum_{\tau_j \in \{\tau \setminus \tau_i\}} (\lfloor \frac{t'}{T_j} \rfloor + 1) C_j + C_i \\ & \ge F(\tau_i, t, t') + C_i \end{aligned}$$

Then it must be that

$$\frac{H_i(t)}{t - D_i} > 1 \Rightarrow t - D_i < U \times t + \sum_{\tau_j \in \{\tau\}} C_j$$
$$\Rightarrow t < \frac{D_i + \sum_{\tau_j \in \tau} C_j}{1 - U}$$