Rate Monotonic Analysis

Introduction

Periodic tasks

Extending basic theory

Synchronization and priority inversion

Aperiodic servers

Case Study: BSY-1 Trainer
A Sample Problem - Periodics

τ₁’s deadline is 20 msec before the end of each period.
**Periodic Tasks**

**Periodic task**
- initiated at fixed intervals
- must finish before start of next cycle

Task’s CPU utilization: $U_i = \frac{C_i}{T_i}$

- $C_i =$ compute time (execution time) for task $\tau_i$
- $T_i =$ period of task $\tau_i$

CPU utilization for a set of tasks:

$$U = U_1 + U_2 + \ldots + U_n$$
Example of Priority Assignment

Semantic-Based Priority Assignment

VIP:

<table>
<thead>
<tr>
<th>0</th>
<th>10</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IP:

<table>
<thead>
<tr>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VIP: misses deadline

IP: \[ U_{IP} = \frac{1}{10} = 0.10 \]

VIP: \[ U_{VIP} = \frac{11}{25} = 0.44 \]

Policy-Based Priority Assignment

IP:

<table>
<thead>
<tr>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VIP:

<table>
<thead>
<tr>
<th>0</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

VIP: misses deadline

IP: \[ U_{IP} = \frac{3}{10} = 0.30 \]

VIP: \[ U_{VIP} = \frac{6}{25} = 0.24 \]
Schedulability: UB Test

*Utilization bound (UB) test:* a set of $n$ independent periodic tasks scheduled by the rate monotonic algorithm will always meet its deadlines, for all task phasings, if

$$\frac{C_1}{T_1} + \ldots + \frac{C_n}{T_n} \leq U(n) = n \left(2^{1/n} - 1\right)$$

- $U(1) = 1.0$
- $U(2) = 0.828$
- $U(3) = 0.779$
- $U(4) = 0.756$
- $U(5) = 0.743$
- $U(6) = 0.734$
- $U(7) = 0.728$
- $U(8) = 0.724$
- $U(9) = 0.720$

For harmonic task sets, the utilization bound is $U(n) = 1.00$ for all $n$.

Note: UB test = Techniques 1 and 2 in handbook.
Sample Problem: Applying UB Test

<table>
<thead>
<tr>
<th>Task  ( \tau ):</th>
<th>( C )</th>
<th>( T )</th>
<th>( U )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task  ( \tau_1 ):</td>
<td>20</td>
<td>100</td>
<td>0.200</td>
</tr>
<tr>
<td>Task  ( \tau_2 ):</td>
<td>40</td>
<td>150</td>
<td>0.267</td>
</tr>
<tr>
<td>Task  ( \tau_3 ):</td>
<td>100</td>
<td>350</td>
<td>0.286</td>
</tr>
</tbody>
</table>

Total utilization is \( 0.200 + 0.267 + 0.286 = 0.753 < U(3) = 0.779 \)

The periodic tasks in the sample problem are schedulable according to the UB test.
Timeline for Sample Problem

\[ \tau_1 \]

\[ \tau_2 \]

\[ \tau_3 \]

Scheduling Points
Exercise: Applying the UB Test

Given:

<table>
<thead>
<tr>
<th>Task ( \tau_i )</th>
<th>( C )</th>
<th>( T )</th>
<th>( U )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_1 )</td>
<td>1</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>( \tau_2 )</td>
<td>2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>( \tau_3 )</td>
<td>1</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

a. What is total utilization?

b. Is the task set schedulable?

c. Draw the timeline.

d. What is the total utilization if \( C_3 = 2 \)?
Toward a More Precise Test

UB test has three possible outcomes:

\[ 0 \leq U \leq U(n) \implies \text{Success} \]

\[ U(n) < U \leq 1.00 \implies \text{Inconclusive} \]

\[ 1.00 < U \implies \text{Overload} \]

UB test is conservative.

A more precise test can be applied.
Schedulability: RT Test

Theorem: for a set of independent, periodic tasks, if each task meets its first deadline, with worst-case task phasing, the deadline will always be met.

Response time (RT) test: let $a_n$ = response time of task $i$. $a_n$ may be computed by the following iterative formula:

$$a_{n+1} = C_i + \sum_{j=1}^{i-1} \left( \frac{a_n}{T_j} \right) C_j$$

where $a_0 = \sum_{j=1}^{i} C_j$

Test terminates when $a_{n+1} = a_n$.

Task $i$ is schedulable if its response time is before its deadline: $a_n \leq T_i$
Example: Applying RT Test -1

Taking the sample problem, we increase the compute time of $\tau_1$ from 20 to 40; is the task set still schedulable?

Utilization of first two tasks: $0.667 < U(2) = 0.828$
  - first two tasks are schedulable by UB test

Utilization of all three tasks: $0.953 > U(3) = 0.779$
  - UB test is inconclusive
  - need to apply RT test
Example: Applying RT Test -2

Use RT test to determine if $\tau_3$ meets its first deadline: $i = 3$

$$a_0 = \sum_{j=1}^{3} C_j = C_1 + C_2 + C_3 = 40 + 40 + 100 = 180$$

$$1 = C_i + \sum_{j=1}^{i-1} \left\lfloor \frac{a_0}{T_j} \right\rfloor C_j = C_3 + \sum_{j=1}^{2} \left\lfloor \frac{a_0}{T_j} \right\rfloor C_j$$

$$= 100 + \left\lfloor \frac{180}{100} \right\rfloor (40) + \left\lfloor \frac{180}{150} \right\rfloor (40) = 100 + 80 + 80 = 260$$
Example: Applying the RT Test -3

\[ C_3 + \sum_{j=1}^{2} \left( \frac{a_1}{T_j} \right) C_j = 100 + \left\lceil \frac{260}{100} \right\rceil (40) + \left\lceil \frac{260}{150} \right\rceil (40) = 30 \]

\[ C_3 + \sum_{j=1}^{2} \left( \frac{a_2}{T_j} \right) C_j = 100 + \left\lceil \frac{300}{100} \right\rceil (40) + \left\lceil \frac{300}{150} \right\rceil (40) = 30 \]

\[ a_3 = a_2 = 300 \quad \text{Done!} \]

Task \( \tau_3 \) is schedulable using RT test.

\[ a_3 = 300 < T = 350 \]
Timeline for Example

\[ \tau_1 \]
\[ \tau_2 \]
\[ \tau_3 \]

\[ \tau_3 \text{ completes its work at } t = 300 \]
Exercise: Applying RT Test

Task $\tau_1$: $C_1 = 1 \quad T_1 = 4$

Task $\tau_2$: $C_2 = 2 \quad T_2 = 6$

Task $\tau_3$: $C_3 = 2 \quad T_3 = 10$

a) Apply UB test

b) Draw timeline

c) Apply RT Test
Summary

UB test is simple but conservative.

RT test is more exact but also more complicated.

To this point, UB and RT tests share the same limitations:

- all tasks run on a single processor
- all tasks are periodic and noninteracting
- deadlines are always at the end of the period
- there are no interrupts
- rate monotonic priorities are assigned
- there is zero context switch overhead
- tasks do not suspend themselves