Modeling Priorities at the CPU

Daniel A. Menascé
Department of Computer Science
George Mason University

Motivation

• Most operating systems use priorities when scheduling the CPU to processes.
• We will use preemptive resume priorities.
• QNs cannot handle this: need approximation.
• P priority groups (1, ..., P): priority group 1 has the highest priority.
Approach

- Add one more CPU (a shadow CPU) for priority groups 2, ..., P.
- Priority group jobs use the original CPU (CPU 1).
- Priority group i uses CPU i.
- Priority group i does not see priority groups i+1, ..., P at its CPU because they have lower priority.
- At CPU i, we need to adjust the service demand to account for use of the actual CPU by jobs of priority groups 1, ..., i-1.
SWIC Algorithm (Menascé and Almeida)

For p = 1 to P do
   Begin
      build a QN model \( Q_p \) containing classes \( r \) such that \( \text{Prior}(r) \leq p \)
      model \( Q_p \) contains \( p \) shadow CPUs
      Inflate CPU service demands as follows:
      \[
      D_{cpu,r}^p = \begin{cases} 
      0 & r \notin \Omega(p) \\
      D_{cpu,r} / \left( 1 - \sum_{q \in \Omega(p)} U_{cpu,r} \right) & r \in \Omega(p)
      \end{cases}
      \]
      Solve \( Q_p \) and find \( X_0,r \) for all classes \( r \) s.t. \( \text{Prior}(r) \leq p \)
      For all \( r \) s.t. \( \text{Prior}(r) \leq p \) do \( U_{cpu,r} = D_{cpu,r} \times X_0,r \)
   End;
   Compute final metrics from model \( Q_p \).

Priority Example

<table>
<thead>
<tr>
<th>Table 15.4. Input Parameters for Priority Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Service Demand (sec)</td>
</tr>
<tr>
<td>Processor</td>
</tr>
<tr>
<td>D1</td>
</tr>
<tr>
<td>D2</td>
</tr>
<tr>
<td>Number of clients</td>
</tr>
<tr>
<td>Think time (sec)</td>
</tr>
</tbody>
</table>

The corporate class has a higher CPU priority than the consumer class.

If they had the same priority, the response times would be:
Consumers: 1.75 sec
Corporate: 3.74 sec.
Solving the Priority Example

1. *Build a single-class model.* This model includes the corporate class only, the original CPU, the terminals, and the disks. Using the parameters of Table 15.4 in the MVA algorithm, yields the throughput of the corporate class as 11.44 request/sec. Therefore, the CPU utilization can be computed as $U_{cpu,p} = D_{cpu,p} \times X_{cpu,p} = 0.033 \times 11.44 = 0.378$.

2. *Add the shadow CPU and the consumer class.* Compute the service demands at the shadow and original CPUs as follows:

$$D_{cpu,c}^{avg} = 0 \quad \text{and} \quad D_{cpu,p}^{avg} = 0$$

$$D_{cpu,p}^{shad} = D_{cpu,p} = 0.033 \text{ sec}$$

$$D_{cpu,c}^{kw} = \frac{D_{cpu,c}}{1 - U_{cpu,p}} = \frac{0.015}{1 - 0.378} = 0.024 \text{ sec}$$

Using the MVA algorithm, the response time for the corporate class is computed as 1.04 sec and as 1.78 sec for the consumer class. So there was a significant improvement in the response time of the corporate class (from 3.74 sec to 1.04 sec) with very little increase in the response time of the consumer class (2% increase). The reason is that the consumer class has a much smaller service demand at the processor than that of the corporate class. So, giving priority to the corporate class at the processor does not hurt the lower-priority class too much.