PROBABILISTIC ADMISSION CONTROL
TO GOVERN REAL-TIME SYSTEMS UNDER OVERLOAD

CLAUDE-J. HAMANN, MICHAEL ROITZSCH, LARS REUTHER, JEAN WOLTER, HERMANN HÄRTIG
MOTIVATION

- desktop real-time
- there are no hard real-time applications on desktops
- there is a lot of firm and soft real-time
  - low-latency audio processing
  - smooth video playback
  - desktop effects
  - user interface responsiveness
H.264 DECODING

WCET
Requirements even slightly below 100% can dramatically reduce resource allocation.
- Statistical Rate Monotonic Scheduling
  - local admission ensures percentage of successful jobs
  - execution time of each job must be known in advance
PROBLEMS

- WCET largely exceeds average case
- poor utilization efficiency
- restricted to specific task types
- tough runtime requirements
- missed deadlines can at best be predicted
DESIGN GOALS

- use distribution instead of WCET
- relax guarantees, improve utilization
- hard, firm, preemptible, non-preemptible
- minimal runtime dispatcher requirements
- controllable fraction of missed deadlines
Use probabilistic admission control to model the actual run-time dispatching.
Hamann et al.  Probabilistic Admission Control
KEY IDEA

WCET
\[ P(J \text{ does not run longer than } r \land \quad J \text{ is completed until its relative deadline}) \geq q \]
tasks $T_i$ are sequences of periodic jobs
period length = relative deadline $d_i$
jobs are partitioned into one mandatory part and $m_i$ optional parts
mandatory part’s execution time $X_i$ with WCET $w_i$
optional part’s execution time $Y_i$
quality $q_i$: fraction of completed optional parts
ADMISSION GOAL

priorities and reservation times for all jobs to generate a feasible schedule

- all mandatory parts meet their deadlines
- all optional parts meet their requested qualities
Quality-Assuring Scheduling (RTSS’01)

- priority assignment:
  - all mandatory parts first
  - higher quality $\rightarrow$ higher priority

reservation times:

$$p_i(r) = \mathbb{P}(Y_i \leq r \land \sum_{i=1}^{n} X_i + \sum_{j=1}^{i-1} \min(Y_j, r_j) + Y_i \leq d)$$
3 Tasks: 1 mandatory, 1 optional part each

\[
p_i(r) = \Pr(Y_i \leq r) \land \\
\sum_{i=1}^{n} X_i + \sum_{j=1}^{i-1} \min(Y_j, r_j) + Y_i \leq d
\]
Downside

- expensive computation for arbitrary periods
- hyperperiod explodes for task sets with close-by period lengths (LCM of 503 and 510 anyone?)
- new algorithm differs in three ways
  - priority assignment
  - notion of reservation time
  - very low-cost admission
Quality-Rate-Monotonic Scheduling

cut down the exact modeling of dispatcher behavior in favor of a simpler algorithm:
- priorities are assigned to tasks as in RMS
- combined reservation for all parts of a job
- reservation time regarded constant execution time in the admission

tasks are independent for admission
QAS:

QRMS:
\[ r'_i = \min(r \in \mathbb{R} \mid \frac{1}{m_i} \sum_{k=1}^{m_i} \mathbb{P}(X_i + k \cdot Y_i \leq r) \geq q_i) \]

\[ r_i = \max(r'_i, w_i) \quad i = 1, \ldots, n \]

- Where is the deadline?
- consider reservation as constant execution time of a rate monotonic task
- use any RMS admission criterion
- aborting by deadline does not happen
COST

- Admission
  - computational cost dominated by convolutions
  - $O(\text{number optional parts} \times (\text{number of bins in distribution})^2)$
  - 5ms per part for hundreds of bins

- Runtime
  - static priorities
<table>
<thead>
<tr>
<th>Period</th>
<th>Mandatory Part</th>
<th>Optional Part</th>
<th>Requested Quality</th>
<th>Achieved Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>N(5,1), w=6.5</td>
<td>N(3,1)</td>
<td>70%</td>
<td>70.23%</td>
</tr>
<tr>
<td>30</td>
<td>E(0.33), w=4</td>
<td>N(2,3)</td>
<td>90%</td>
<td>89.72%</td>
</tr>
<tr>
<td>50</td>
<td>E(0.25), w=2</td>
<td>N(5,19.5)</td>
<td>80%</td>
<td>78.44%</td>
</tr>
</tbody>
</table>
### QRMS VS. SRMS

<table>
<thead>
<tr>
<th>Period</th>
<th>Mandatory Part</th>
<th>Optional Part</th>
<th>Requested Quality</th>
<th>QRMS Quality</th>
<th>SRMS Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>N(2,0.5), w=3</td>
<td>N(1.5,0.5)</td>
<td>70%</td>
<td>70.06%</td>
<td>85.9%</td>
</tr>
<tr>
<td>20</td>
<td>E(0.33), w=6</td>
<td>N(2,1)</td>
<td>50%</td>
<td>99.95%</td>
<td>77.5%</td>
</tr>
<tr>
<td>60</td>
<td>N(6,3), w=10</td>
<td>E(10)</td>
<td>75%</td>
<td>74.76%</td>
<td>79.3%</td>
</tr>
</tbody>
</table>

\[
r_i = \max(r_i', w_i) \quad i = 1, \ldots, n
\]
- performed simulations: random qualities, random distributions

uniform, optional only

uniform

harmonic

- yet to come: quantitative analysis, utilization discussion, application studies
CONCLUSION

- handles arbitrary, empiric distributions
- high utilization by probabilistic guarantees
- mandatory and optional parts, subjobs
- static priority dispatching
- intuitive quality parameter