Using Linked List in Exact Schedulability Tests for Fixed Priority Scheduling

Jiaming Lv, Xingliang Zou, Albert M. K. Cheng, and Yu Jiang

Heilongjiang University and University of Houston

IEEE RTAS 2016 WiP
Outline

- The classic model (LL) and the response time analysis (RTA)
- Current best exact test
- Our linked list-based methods
- Conclusion
The classic model (LL)

- The classic preemptive model

- For *fixed priority scheduling*, synchronous release defines critical instant of tasks.

- The *Rate-Monotonic* (RM) fixed priority scheduling is optimal for periodic/sporadic tasks that comply with a restrictive system model and that have implicit deadlines.
The RTA-based exact test

- Audsley, Burns, Richardson, Tindell, and Wellings presented the famous response time analysis (RTA) algorithm, calculating in an iterative way.
  - \( R_i = C_i + \sum_{0<j<i} (\text{ceiling}(R_i/T_j)\times C_j) \)
- The RTA-based test is a kind of exact (necessary and sufficient) schedulability test.
Current best exact test

- Davis, Zabos, and Burns presented the current best RTA-based test with better initial values (in general sense, i.e., no particular restrictions such as harmonic on task periods).

- Eisenbrand and Rothvoss regarded that response time computation is NP-hard.
Our Method

- We employ the linked list structure in the schedulability test and RTA, and
- use a linked list node to represent a *busy period* of a time interval.
- We use also a memory management in user space.
- An example
  
  3-task set: $C_1=1, T_1=4; \ C_2=2, T_2=5; \ C_3=2, T_3=20;$ implicit deadlines; RM scheduling

April 2016
An Example

$C_1 = 1, \ T_1 = 4; \quad C_2 = 2, \ T_2 = 5; \quad C_3 = 2, \ T_3 = 20; \quad$ implicit deadlines

The number of total jobs: 10 ($= 5 + 4 + 1$). Response Time: $RT_1 = 1, \ RT = 3, \ RT_3 = 8$
TABLE I
HET, RTA, AND LList-based Algorithms, Execution Time in Counter Clock Cycles × 1,000

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Orders of magnitude spanning tasks periods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1(a)</td>
</tr>
<tr>
<td>HET</td>
<td>6.9</td>
</tr>
<tr>
<td>RTA</td>
<td>29.56</td>
</tr>
<tr>
<td>LList</td>
<td>1.01</td>
</tr>
<tr>
<td>RTA/LList</td>
<td>29.3</td>
</tr>
</tbody>
</table>

The range of periods starts from $10$ and $10^4$ for (a) and (b), respectively.
Fig. 1. Average execution time required by the HET, RTA, and LLList-based exact schedulability tests versus number of orders of magnitude range of task periods. The range of periods starts from 10 and $10^4$ for (a) and (b), respectively. Note both x and y-axes are logarithmic scales.
For an $n$-task set under fixed priority RM scheduling, the total number of jobs, $N$, in the test time interval $[0, T_n)$ is at most $N = \sum_{i=1}^{n} \text{ceiling}(T_n/T_i)$ needed to be simulated in the LList-based exact test.

When $T_n/T_1 \leq 10$ (one order of magnitude), there will be at most $9n+8$ jobs needed to be simulated, and our method needs only $O(n)$ time for simulating an $n$-task set, regardless of the length of the maximum period of the task set as well as the total utilization factor.
Within three orders of magnitude spanning task periods, the memory space needed by the linked list is completely affordable.

**TABLE II**

**THE MAXIMUM NUMBER OF NODES AND CORRESPONDING MEMORY SPACE**

<table>
<thead>
<tr>
<th>Orders of magnitude spanning tasks periods</th>
<th>1(a)</th>
<th>1(b)</th>
<th>2(a)</th>
<th>2(b)</th>
<th>3(a)</th>
<th>3(b)</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>max # nodes</td>
<td>28</td>
<td>58</td>
<td>236</td>
<td>441</td>
<td>2302</td>
<td>3246</td>
<td>20996</td>
</tr>
<tr>
<td>KB(int32)</td>
<td>0.3</td>
<td>0.7</td>
<td>2.8</td>
<td>5.2</td>
<td>27.0</td>
<td>38.0</td>
<td>246.0</td>
</tr>
</tbody>
</table>
Conclusion

- We provide an LLList-based method for the response time analysis and exact scheduability test for fixed priority preemptive systems.
- Our preliminary results have shown that the LLList-based exact test is a better candidate in exact RTA-based tests when task periods span no more than three orders of magnitude.
Thanks for your interest!

April 2016