Probabilistic Deadline Miss Analysis of Real-Time Systems Using Regenerative Transient Analysis

L. Carnevali\textsuperscript{1}, A. Melani\textsuperscript{2}, L. Santinelli\textsuperscript{3}, G. Lipari\textsuperscript{4}

\textsuperscript{1}Department of Information Engineering, University of Florence
laura.carnevali@unifi.it - http://stlab.dinfo.unifi.it/carnevali

\textsuperscript{2}ReTiS Lab, Scuola Superiore Sant’Anna, Pisa

\textsuperscript{3}ONERA, Toulouse - \textsuperscript{4}Université de Lille 1

RTNS, Versailles - October 10, 2014
Some motivations

- Non-functional requirements prescribed by certification standards e.g., performance requirements on response time and scalability
- Increasing relevance for verification of RAMS requirements: not only Safety, but also Reliability, Availability, Maintainability

Some areas of investigation

- WCET (Worst Case Execution Time) estimation
  - Static methods (Thesing et al, Healy et al, etc.)
  - Measurement-based methods (Puaut et al, Santinelli et al, etc.)
- Analysis of probabilistic real-time systems
  - Analytical methods (Diaz et al, Cucu-Grosjean et al)
  - State-space based methods (Lindemann et al, Vicario et al, etc.)
The contribution

- A probabilistic approach for the analysis of real-time systems with stochastic parameters estimated from real measurements

- Combines results consolidated in the areas of:
  - Estimation of probabilistic WCETs (Extreme Value Theory)
  - Probabilistic analysis (method of stochastic state classes)

- Experimented on a case study including probabilistic WCETs estimated from benchmarks and real system executions

- Targeted to the evaluation of the probability of deadline miss
Given $C_k$ the distribution of the execution time of a task measured in a certain configuration / condition $k$, the probabilistic Worst-Case Execution Time distribution $\bar{C}$ is an upper-bound on the execution time distribution $C_k$ of all the possible execution conditions of the task.
Probabilistic WCET estimation

- Check *identical distribution* (KS test)
  - If OK, verify subsequent hypothesis
- Check *independence* (AR test)
  - If KO, check *stationarity* (LB test) or *independence of extremal samples* (extremogram / extremal index)

If hypothesis verification is successful, apply the Extreme Value Theory (EVT) to derive a Generalized Extreme Value (GEV) distribution (Gumbel / Fréchet / Weibull)

- Derive the tightest Erlang distribution $\hat{C}$ that upper-bounds the GEV distribution $\bar{C}$
Environment for tasks implemented from the Mälardalen benchmark

- A machine with 2 Intel® Xeon® E5620 2.4 GHz sockets
- each machine having 4 cores and 3 cache levels
- SchedMCore\(^2\), for precise real-time execution
- LTTng (Linux Trace Toolkit new generation)\(^3\), for performance monitoring

Results of hypothesis verification

<table>
<thead>
<tr>
<th>Task</th>
<th>KS (Identical distribution)</th>
<th>LB (Independence)</th>
<th>AR (Stationarity)</th>
<th>(\hat{\rho}(5)) (Extremogram)</th>
<th>(\theta) (Extremal index)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ns</td>
<td>0.90</td>
<td>(AR(40))</td>
<td>0</td>
<td>0.005</td>
<td>0.995</td>
</tr>
<tr>
<td>cnt_isol</td>
<td>0.56</td>
<td>(AR(9))</td>
<td>0.0005</td>
<td>0.003</td>
<td>1</td>
</tr>
<tr>
<td>cnt_mc</td>
<td>0.26</td>
<td>(AR(38))</td>
<td>0</td>
<td>0.005</td>
<td>1</td>
</tr>
<tr>
<td>edn</td>
<td>0.82</td>
<td>(AR(39))</td>
<td>0</td>
<td>0.01</td>
<td>1</td>
</tr>
</tbody>
</table>

The algorithm for pWCET estimation can be safely applied

---

- Single-processor real-time system
- Fixed-priority non-preemptive scheduling policy
  - A higher number corresponds to a higher priority level
- Periodic real-time tasks
  - each associated with a relative deadline
  - each having either a *deterministic* or a *probabilistic* execution time
    (specified by a *non-Markovian* probability distribution function)
- A job is discarded as soon as its deadline is missed
- Jobs do not use mutex semaphores to synchronize
- A job cannot self-suspend before its completion
A class of non-Markovian Stochastic Petri Nets (NMSPN) \(^4\)
- Encompass concurrent GEN transitions with bounded support
- State = marking + remaining times of GEN transitions

Extended with enabling functions, flush functions, and priorities
- Change the enabling condition of transitions and the token moves
- Neither restrict the model expressivity nor impact on the analysis

<table>
<thead>
<tr>
<th>Transition</th>
<th>Enabling function</th>
<th>Flush function</th>
<th>Prio</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>h9 == 1</td>
<td>{ p4, p5 }</td>
<td>-</td>
</tr>
<tr>
<td>hour8</td>
<td>-</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td>hour9</td>
<td>-</td>
<td>-</td>
<td>10</td>
</tr>
</tbody>
</table>

Stochastic model of a task

- The underlying process is a Markov Regenerative Process (MRP) including multiple concurrent generally distributed (GEN) timers
  - Regeneration point: the future is independent from the past
  - The model regenerates at each hyper-period

- Example: period=10; deadline=10; execution time supported over \([1, 2]\)

```
<table>
<thead>
<tr>
<th>Transition</th>
<th>Enabling function</th>
<th>Flush function</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>deadline</td>
<td>-</td>
<td>-</td>
<td>9</td>
</tr>
<tr>
<td>release</td>
<td>-</td>
<td>-</td>
<td>8</td>
</tr>
<tr>
<td>wait</td>
<td>-</td>
<td>-</td>
<td>3</td>
</tr>
<tr>
<td>job</td>
<td>-</td>
<td>{dStarted}</td>
<td>-</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
```
Solution technique: the method of stochastic state classes

- A stochastic state class is a tuple $\langle m, D, f \rangle$
  - $m : P \rightarrow \mathbb{N}$ is a marking
  - $D \subseteq \mathbb{R}_{\geq 0}^n$ is a set of values for times-to-fire $\bar{\tau}$
  - $f : D \rightarrow [0, 1]$ is the PDF of the random vector $\bar{\tau}$ over $D$

- The successor $\Sigma' = \langle m', D', f' \rangle$ of $\Sigma$ through transition $t$ holds all the possible states after the firing of $t$ in $\Sigma$ and their joint PDF
  - $D$ is a Difference Bounds Matrix zone (DBM zone)
  - $f$ is a continuous function (piecewise over DBM subdomains)

- Symbolic calculus implemented in ORIS for expolynomials

- Classes allow to derive kernels of the underlying MRP

  - The transient probability of reachable markings is derived through numerical solution of generalized Markov renewal equations

---

5 V. G. Kulkarni, Modeling and analysis of stochastic systems, CRC Press, 1996

The ORIS Tool

- Graphical Petri net editor
- Transient analysis of non-Markovian stochastic Petri nets
- Full Java implementation (cross-platform)
- Available at http://oris-tool.org
Experimental setting

- A task-set made of 5 periodic tasks
  - 2 tasks have a *deterministic* WCET
  - 3 tasks have a *probabilistic* WCET

<table>
<thead>
<tr>
<th>Task</th>
<th>Period</th>
<th>Priority</th>
<th>Execution time</th>
<th>Benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>tsk_1</td>
<td>1.0</td>
<td>5</td>
<td>DET(0.2)</td>
<td>-</td>
</tr>
<tr>
<td>tsk_2</td>
<td>1.2</td>
<td>4</td>
<td>DET(0.4)</td>
<td>-</td>
</tr>
<tr>
<td>tsk_3</td>
<td>1.5</td>
<td>3</td>
<td>Erlang(3, 120.19)</td>
<td>ns</td>
</tr>
<tr>
<td>tsk_4</td>
<td>2.0</td>
<td>2</td>
<td>Erlang(3, 40.27)</td>
<td>cnt_isol</td>
</tr>
<tr>
<td>tsk_5</td>
<td>3.0</td>
<td>1</td>
<td>Erlang(4, 59.50)</td>
<td>cnt_mc</td>
</tr>
</tbody>
</table>

- Analysis repeated for different distributions of $tsk_3$ execution time
  - Erlang(2, 50.05), *ns* benchmark
  - Erlang(2, 28.70), *cnt_mc* benchmark
  - Erlang(2, 18.95), *cnt_isol* benchmark
  - Erlang(2, 10.67), *edn* benchmark

- $\sim15''$ to enumerate state-space, $\sim30'$ to solve renewal equations
Experimental results: deadline miss probability within time $t$

(a) $t_{sk3}$ pWCET: Erlang(2, 50.05)

(b) $t_{sk3}$ pWCET: Erlang(2, 28.70)

(c) $t_{sk3}$ pWCET: Erlang(2, 18.95)

(d) $t_{sk3}$ pWCET: Erlang(2, 10.67)
Concluding remarks

A probabilistic approach for the analysis of real-time systems
- Estimation of pWCETs through the Extreme Value Theory
- Probabilistic analysis through the method of stochastic classes
- Experiments with pWCETs derived from real benchmarks

Future issues
- How do guarantees of pWCETs reflect on schedulability results?
- What is the accuracy attained by schedulability results if different distributions are used in pWCET estimation?
- Can other scheduling policies be encompassed in the approach?