Probabilistic Real-Time Guarantees: There is life beyond the i.i.d. assumption

Bernardo Villalba Frías¹, Luigi Palopoli¹, Luca Abeni² and Daniele Fontanelli³

¹ Department of Information Engineering and Computer Science – University of Trento
² ReTiS Laboratory – Scuola Superiore Sant'Anna
³ Department of Industrial Engineering – University of Trento
• Introduction.
• Limitations in the current Probabilistic Guarantees analysis.
• The Markov Computation Time Model (MCTM).
• Stochastic Analysis of the MCTM.
• Estimating the MCTM parameters.
• Experimental Validation.
Soft Real-Time Systems

• Hard real-time systems:
  – Mature and effective methods.
  – However, many recent real-time systems are not hard real-time.

• Soft real-time systems:
  – Resilient to occasional and controlled timing failures.
Examples

Web TV

Infotainment
Examples

Web TV

Infotainment

Visual Control
Probabilistic Guarantees

• **Performance specifications** stated in **probabilistic terms**:
  – Associate each task with a probability of respecting the deadline.
  – Timing requirement to enforce: respect the deadline with a given probability.

• State of the art: **Numerical methods and analytic bounds**.

• Stochastic **analysis** based on the **i.i.d. assumption**.
• Resource Reservations:
  – Scheduling parameters: \( Q^s \) and \( T^s \).
  – Executes \( Q^s \) time units in every reservation period \( T^s \).
  – Guarantees temporal isolation.

• CBS scheduler: SCHED_DEADLINE.

• The probability of respecting the deadline is:

\[
\Pr \{ \delta_j \leq T \}
\]
CBS as a Markov chain

• Model for the evolution of $\delta_j$:

$$v_1 = c_1$$

$$v_j = \max\{0, v_{j-1} - \frac{T}{T^s} Q^s\} + c_j$$

$$\delta_j = \left[ \frac{v_j}{Q^s} \right] T^s$$

with $C(c) = \Pr \{c_j = c\}$
CBS as a Markov chain

- Model for the evolution of $\delta_j$:
  
  $v_1 = c_1$

  $v_j = \max\{0, v_{j-1} - \frac{T}{T^s} Q^s\} + c_j$

  $\delta_j = \left[\frac{v_j}{Q^s}\right] T^s$

  with $C(c) = \Pr\{c_j = c\}$

- State of the system:
  
  $S_h(j) = \{v_j = c^{\text{min}} + h\}$
CBS as a Markov chain

• Model for the evolution of $\delta_j$:

$$v_1 = c_1$$

$$v_j = \max\{0, v_{j-1} - \frac{T}{T^s} Q^s\} + c_j$$

$$\delta_j = \left[ \frac{v_j}{Q^s} \right] T^s$$

with $C(c) = \Pr \{c_j = c\}$

• State of the system:

$$S_h(j) = \{v_j = c_{\text{min}} + h\}$$

• Define the probability:

$$\pi_h(j) = \Pr \{S_h(j)\}$$
CBS as a Markov chain

- Dynamic of the workload modeled as a queue:\n
\[ v_j = c_{min}^{\text{max}} \]
\[ v_j = c_{min}^{\text{max}} + Q^i \]
\[ v_j = c_{min}^{\text{max}} + 2Q^i \]

---

CBS as a Markov chain

- Dynamic of the workload modeled as a queue\(^1\):

- Introduce the vector:

\[
\Pi(j) = [\pi_0(j) \pi_1(j) \pi_2(j) \ldots]
\]

CBS as a Markov chain

• Dynamic of the workload modeled as a queue\(^1\):

\[ \begin{bmatrix} v_1 = \text{c}^\text{mn} & v_1 = \text{c}^\text{mn} + \text{Q}^\text{c} & v_1 = \text{c}^\text{mn} + 2\text{Q}^\text{c} \end{bmatrix} \]

• Introduce the vector:

\[ \Pi(j) = [\pi_0(j) \ \pi_1(j) \ \pi_2(j) \ \ldots] \]

• Probability evolution:

\[ \Pi(j) = \Pi(j - 1)P \]

\[ P = \begin{bmatrix} a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & 0 & \ldots \\ a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ldots \\ a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & 0 & \ldots \\ 0 & a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & \ldots \\ 0 & 0 & a_0 & a_1 & a_2 & \ldots & a_n & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ldots \end{bmatrix} \]

CBS as a Markov chain

- Dynamic of the workload modeled as a queue:\n
  \[ \pi_0, \pi_1, \pi_2, \ldots \]

  \[ v_j = c^{\text{min}} + Q \]

- Introduce the vector:

  \[ \Pi(j) = [\pi_0(j), \pi_1(j), \pi_2(j), \ldots] \]

- Probability evolution:

  \[ \Pi(j) = \Pi(j - 1)P \]

- Steady state:

  \[ \overline{\Pi} = \lim_{j \to \infty} \Pi(j) \]

\[ P = \begin{bmatrix}
  a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & 0 & \ldots \\
  a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & 0 & \ldots \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
  a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & 0 & \ldots \\
  0 & a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & \ldots \\
  0 & 0 & a_0 & a_1 & a_2 & \ldots & a_n & 0 & \ldots \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots 
\end{bmatrix} \]

Results on i.i.d. execution times

- Test for independence\(^1\): Runs of “above and below the mean” with 0.05 significance level:

<table>
<thead>
<tr>
<th>z-statistic</th>
<th>p-value (&gt;0.05)</th>
<th>hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.5295</td>
<td>0.2938</td>
<td>Accepted</td>
</tr>
</tbody>
</table>

Results on non i.i.d. execution times

- Test for independence: Runs of “above and below the mean” with 0.05 significance level:

<table>
<thead>
<tr>
<th>z-statistic</th>
<th>p-value (&gt;0.05)</th>
<th>hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>-100.5715</td>
<td>0.0000</td>
<td>Rejected</td>
</tr>
</tbody>
</table>
A motivating example

- Extended use of **randomized algorithms** in robotics.
- Path-following robot with image processing.
- Possibly, different types of environment.
- The robot will remain in **one mode** for a while.
- Then, it will **switch mode**... continuously.
The modeling idea

- **A Markov chain to model the “mode change”:**

![Diagram of a Markov chain with two states and transition probabilities.](image)
The modeling idea

- A Markov chain to model the “mode change”:

- The switching behavior introduces dependencies.
- Finite number of modes.
- If transitions are defined as a Markov Process.
- Model corresponds to a Markov Modulated Process.
A Markov Computation Time Model (MCTM) is defined as the triple \( \{ M, P, C \} \)

\[ M = \{ m_1, \ldots, m_N \} \]

\[ P = (p_{a, b}), \quad \forall a, b \in M \]

\[ p_{a, b} = \text{Pr} \{ m_j = b | m_{j-1} = a \} \]

\[ C = \{ C_{m_j} : m_j \in M \} \]
A Markov Computation Time Model (MCTM) is defined as the triple $\{M, P, C\}$

$$M = \{m_1, \ldots, m_N\}$$

$$P = (p_{a,b}), \ \forall a, b \in M$$

$$p_{a,b} = \Pr \{m_j = b | m_{j-1} = a\}$$

$$C = \{C_{m_j} : m_j \in M\}$$

### Assumptions:

- Job’s execution time **only depends** on the current mode.
- The “mode change” event is **independent** both from:
  - The current computation workload.
  - The execution time required by the previous job.
By modeling the dependencies as a Markov system with a switching behavior, it is possible to describe, more precisely, the non i.i.d. execution times.
The new model of the system:
The new model of the system:

\[ S_h(j) = \{ v_j = c^{\text{min}} + h \} \]
\[ \pi_h(j) = \Pr \{ S_h(j) \} \]
\[ \Pi(j) = [\pi_0(j) \ \pi_1(j) \ \pi_2(j) \ \ldots] \]

\[ S_{g,h}(j) = \{ m_j = g \} \land \{ v_j = c^{\text{min}} + h \} \]
\[ \pi_{g,h}(j) = \Pr \{ S_{g,h}(j) \} \]
\[ \Pi_{h}(j) = [\pi_{1,h}(j) \ \pi_{2,h}(j) \ \ldots \ \pi_{N,h}(j)] \]
\[ \Pi(j) = [\Pi_0(j) \ \Pi_1(j) \ \Pi_2(j) \ \ldots] \]
Stochastic Analysis

- The new model of the system:

\[ S_h(j) = \{ v_j = c^{\text{min}} + h \} \]
\[ \pi_h(j) = \Pr \{ S_h(j) \} \]
\[ \Pi(j) = [\pi_0(j) \ \pi_1(j) \ \pi_2(j) \ \ldots] \]

\[ S_{g,h}(j) = \{ m_j = g \} \land \{ v_j = c^{\text{min}} + h \} \]
\[ \pi_{g,h}(j) = \Pr \{ S_{g,h}(j) \} \]
\[ \Pi_{h}(j) = [\pi_{1,h}(j) \ \pi_{2,h}(j) \ \ldots \ \pi_{N,h}(j)] \]
\[ \Pi(j) = [\Pi_0(j) \ \Pi_1(j) \ \Pi_2(j) \ \ldots] \]

- Probability evolution:

\[ \Pi(j) = \Pi(j - 1)P \]
\[ \overline{\Pi} = \lim_{j \to \infty} \Pi(j) \]
A Markov chain

$P = \begin{bmatrix}
a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & 0 & \ldots \\
a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & 0 & \ldots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & 0 & \ldots \\
0 & a_0 & a_1 & a_2 & \ldots & a_n & 0 & 0 & \ldots \\
0 & 0 & a_0 & a_1 & a_2 & \ldots & a_n & 0 & \ldots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \vdots & \ldots \\
\end{bmatrix}$
A Markov chain

\[ P = \begin{bmatrix}
  a_0 & a_1 & a_2 & \cdots & a_n & 0 & 0 & 0 & \cdots \\
  a_0 & a_1 & a_2 & \cdots & a_n & 0 & 0 & 0 & \cdots \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
  a_0 & a_1 & a_2 & \cdots & a_n & 0 & 0 & 0 & \cdots \\
  0 & a_0 & a_1 & a_2 & \cdots & a_n & 0 & 0 & \cdots \\
  0 & 0 & a_0 & a_1 & a_2 & \cdots & a_n & 0 & \cdots \\
  \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots 
\end{bmatrix} \]
Stochastic Analysis

- Each individual block \( (P_e) \) is given by:

\[
P_e = \begin{bmatrix}
p_{1,1} \cdot \alpha_{1,e} & p_{1,2} \cdot \alpha_{2,e} & \cdots & p_{1,N} \cdot \alpha_{N,e} \\
p_{2,1} \cdot \alpha_{1,e} & p_{2,2} \cdot \alpha_{2,e} & \cdots & p_{2,N} \cdot \alpha_{N,e} \\
\vdots & \vdots & \ddots & \vdots \\
p_{N,1} \cdot \alpha_{1,e} & p_{N,2} \cdot \alpha_{2,e} & \cdots & p_{N,N} \cdot \alpha_{N,e}
\end{bmatrix}
\]

with:

\[
p_{a, b} = \Pr \{ m_j = b \mid m_{j-1} = a \}
\]

\[
\alpha_{b, h} = C_b (c_{\text{min}} + h)
\]
Stochastic Analysis

• Each individual block \((P_e)\) is given by:

\[
P_e = \begin{bmatrix}
p_{1,1} \cdot \alpha_{1,e} & p_{1,2} \cdot \alpha_{2,e} & \cdots & p_{1,N} \cdot \alpha_{N,e} \\
p_{2,1} \cdot \alpha_{1,e} & p_{2,2} \cdot \alpha_{2,e} & \cdots & p_{2,N} \cdot \alpha_{N,e} \\
\vdots & \vdots & \ddots & \vdots \\
p_{N,1} \cdot \alpha_{1,e} & p_{N,2} \cdot \alpha_{2,e} & \cdots & p_{N,N} \cdot \alpha_{N,e}
\end{bmatrix}
\]

with:

\[
p_{a, b} = \Pr \{m_j = b | m_{j-1} = a\}
\]

\[
\alpha_{b, h} = C_b (c^{\min} + h)
\]

• Numerically solved:
  – Cyclic or Logarithmic Reduction.
Stochastic Analysis

• Each individual block \((P_e)\) is given by:

\[
P_e = \begin{bmatrix}
p_{1,1} \cdot \alpha_{1,e} & p_{1,2} \cdot \alpha_{2,e} & \cdots & p_{1,N} \cdot \alpha_{N,e} \\
p_{2,1} \cdot \alpha_{1,e} & p_{2,2} \cdot \alpha_{2,e} & \cdots & p_{2,N} \cdot \alpha_{N,e} \\
\vdots & \vdots & \ddots & \vdots \\
p_{N,1} \cdot \alpha_{1,e} & p_{N,2} \cdot \alpha_{2,e} & \cdots & p_{N,N} \cdot \alpha_{N,e}
\end{bmatrix}
\]

with:

\[
p_{a, b} = \Pr \{m_j = b | m_{j-1} = a\}
\]

\[
\alpha_{b, h} = C_b (c_{\text{min}} + h)
\]

• Numerically solved:
  – Cyclic or Logarithmic Reduction.

• Steady state distribution of the response time:

\[
\lim_{j \to \infty} \Pr \{\delta_j \leq D\}
\]
By presenting the **MCTM** as a QBDP and using the **available numeric solutions**, it is possible to obtain the **probability of respecting the deadline** for the proposed model.
• **Identify** the values of \( \{M, P, C\} \)

• **Solved by the Baum-Welch algorithm:**
  – Iterative estimation of the parameters.
  – Convergence to the maximum likelihood matrices.
**Parameter Validation**

- **Validate** the estimated values of \( \{M, P, C\} \)
- **Solved by the Viterbi algorithm:**
  - Generates a sequence of hidden states.
  - Obtains N sub-sequences of execution times.
- **Perform a numerical test for independence.**
Results on non i.i.d. execution times

<table>
<thead>
<tr>
<th>Mode</th>
<th>z-statistic</th>
<th>p-value (&gt;0.05)</th>
<th>hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.6585</td>
<td>0.7449</td>
<td>Accepted</td>
</tr>
<tr>
<td>2</td>
<td>-0.3214</td>
<td>0.3739</td>
<td>Accepted</td>
</tr>
</tbody>
</table>
By applying **standard identification techniques** on hidden Markov models, it is possible to **estimate the parameters** describing the Markov Computation Time Model.
Experimental Setup

• Robotic vision application.
• Two operating conditions.
• 100 sequences of observations.
• 18400 execution times per sequence.
• WandBoard running Ubuntu.
• Linux Kernel 4.8.1
Parameter Estimation

- Execution times obtained from image processing algorithm.
- Strong autocorrelation.
- Test for independence: Runs of “above and below the mean” with 0.05 significance level.

<table>
<thead>
<tr>
<th>z-statistic</th>
<th>p-value (&gt;0.05)</th>
<th>hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>-37.3271</td>
<td>0.0000</td>
<td>Rejected</td>
</tr>
</tbody>
</table>
Parameter Estimation

- Applying the estimation technique for different input data.
- Consistently identified a 5-modes MCTM.
- Test for independence: Runs of “above and below the mean” with 0.05 significance level.

<table>
<thead>
<tr>
<th>Mode</th>
<th>z-statistic</th>
<th>p-value (&gt;0.05)</th>
<th>hypothesis</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.3929</td>
<td>0.0818</td>
<td>Accepted</td>
</tr>
<tr>
<td>2</td>
<td>-1.1932</td>
<td>0.1164</td>
<td>Accepted</td>
</tr>
<tr>
<td>3</td>
<td>-1.1088</td>
<td>0.1338</td>
<td>Accepted</td>
</tr>
<tr>
<td>4</td>
<td>-1.5830</td>
<td>0.0567</td>
<td>Accepted</td>
</tr>
<tr>
<td>5</td>
<td>-0.6522</td>
<td>0.2571</td>
<td>Accepted</td>
</tr>
</tbody>
</table>
Fixed bandwidth (16%)

- **Scheduling parameters:**
  - $T = 100$ ms.
  - $(T^s, Q^s) = (25$ ms, 4 ms).

- **Probability of respecting the deadline:**
  \[
  \frac{\text{# of jobs respecting the deadline}}{\text{# of jobs}}
  \]

- **Overestimation** when considering the i.i.d. model.

- **Good match:** MCTM vs. Real application.
Scheduling parameters:
- $T = 200$ ms.
- $(T^s, Q^s) = (50 \text{ ms}, 35 \text{ ms})$.

Probability of respecting the deadline:
\[
\frac{\# \text{ of jobs respecting the deadline}}{\# \text{ of jobs}}
\]

Overestimation when considering the i.i.d. model.

Good match: MCTM vs. Real application.
**Fixed deadline (D = T)**

- Different values for the bandwidth are explored.
- **Good performance** of the approach compared with the real application.
Conclusions

• Provided **probabilistic guarantees** for soft real-time systems **characterized by dependencies in the execution times**.

• Introduced a **Markovian representation** of the system to **model** these **dependencies**.

• Adapted the techniques for **probabilistic guarantees** to the case of **MCTM**.

• Shown a technique for the **estimation** of the **MCTM parameters**.
Probabilistic Real-Time Guarantees: There is life beyond the i.i.d. assumption

Thanks.

Bernardo Villalba Frías
br.villalbafrías@unitn.it