Probabilities – a key solution for tomorrow real-time compositional frameworks

L. Cucu-Grosjean and co-authors
Outline

- Design of a physical system with time constraints
  - Verification of time constraints
- Probabilities: how do we compose?
- Measurement-based approaches
  - The (missunderstood) independence
  - The impact of the measurement protocol
- Analytical vs. measurement-based
- Back to models to solve the representativity
- Conclusion
Design of a physical system with time constraints

- Real-time systems
- Cyber-physical systems
Design of a physical system with time constraints (2)

Control Theory

\[ \frac{dx}{dt} = f(t) \]

Models

\[ x_{solution} \]

Budgets/programs

Processeur

WCETs

Time Verification

Function analysis
Implementation

Controller

\[ y(t) \]

\[ r(t) \]

Thales Divisions: Different Approaches, Environment

Tool: MelodyCCM – MyCCM C/C++

Architecture: distributed

Application: complex communication schemes

Timing constraints: soft and hard

Tool: MelodyCCM Spatial – MyCCM Ada

Architecture: mono-processor

Application: component implementation in UML

Timing constraints: hard

Tool: MelodyAdvance

Architecture: distributed and heterogeneous

Application: data-flow (functional chains)

Timing constraints: soft

Tool: SoftArc – XML Description

Architecture: mono-processor

Application: call graph

Timing constraints: hard

Indeed the execution times distribution plot (as shown in Figure 1.15 for a dedicated embedded processor [Hansen 2009]) is expected to spread out, so that approaching the worst case execution time is foreseen to become a rare event.

Therefore the amount of wasted computing power is expected to increase, leading to the over-sizing of embedded computers, power supplies and cooling systems, among other issues.

Nowadays, even if many attempts are proposed to give an upper bound of the WCET (e.g: [Rochange 2007]), both the traditional and current approaches are difficult to apply to modern processor generations and produce values which are more and more pessimistic.

Lately, a new approach to improve the robustness of the task scheduling has been proposed in [Midonnet 2010]) where on a temporal fault occurrence, the slack time (the difference between the allocated time and the real execution time) can be dynamically determined and assigned to the faulty task in order to complete its treatment.

As the authors mean " deadline miss" when they talk about "faults", this approach implies that the computing resource has enough spare resource to add more time slack.

This approach has two drawbacks in practice. First, the dynamic addition of the time slack will cause indeterminacy and worsen the predictability of the system. Second, from a control system point of view, it is known that delays decrease the control performance and moreover, as the time slack is dynamically added, it will result into a control system with time-varying and non bounded delay, leading to more difficulties in the analysis.

On the other hand, the method proposed in this thesis allows for the enhancement of predictability and determinism of the system while enhancing control performance. This is achieved by "weakening real-time constraints".
Verification of time constraints

One processor, fixed-priority solution

\( \tau_1 \) \( (0, 1, 3, 3) \)

\( \tau_2 \) \( (3, 2, 6, 6) \)

\( \begin{pmatrix} 2 & 4 \\ 0.2 & 0.8 \end{pmatrix} \)

\( \tau_3 \) \( (2, 1, 6, 6) \)

Execution time

Response time
Probabilities: how do we compose?

Measurements (statistical approaches)
Static analyses (probabilistic approaches)
Hybrid methods

\[ \text{pET: probabilistic Execution Time; } \text{pWCET: probabilistic Worst Case ET} \]
How do we deal with probabilities?

For a program and a processor the execution time extremes are bounded by a Extreme Value Theory Distribution [Edgar et Burns at RTSS2001]

- Independence hypothesis
- Identically distributed hypothesis
Classes of independence

- (Functional) Independence between programs
- Statistical independence
- Probabilistic independence
Functional independence between programs

\[ C_A = C_A \quad \text{and} \quad C_B = C_B \]
Statistical dependence

The two sets of execution times are dependent

Two programs with (functional) dependences
Statistical independence

if y odd then
  { x = 2*y
    wait(y)
  }
else  {x =y + 3
    wait(x)
}

for i= 1 to x
  wait (1)

{68, 59, 84, 94, 100, 57}  {39, 27, 39, 36, 34, 41}

{69, 63, 85, 95, 101, 61}  {39, 27, 39, 36, 34, 41}

The two sets of execution times are independent

Two programs that have (functional) dependences
Multi-path programs

- The execution times are obtained per path and studied in different buckets

- All execution times are in one single bucket
Multi-paths and dependences

ตรวจสอบ: กรณีการขนส่งทางราง: ระหว่าง-และภายใน-ของถัง
ตรวจสอบ: กรณีการขนส่งทางราง: ภายใน-ของถัง
ตรวจสอบ: กรณีการขนส่งทางการบิน: ภายใน-ของถัง
Dependences

• Decreasing the number of dependences is good, hoping to make them disappear is not realistic

• In presence of dependences, the order of execution times becomes important
  – A WCET measurement-based estimator should come with its own measurement protocol

• Manipulating the input execution times has a direct impact on the estimated pWCET
  – Monotonic property
  – Shuffling the input execution times
What dependences?

Cluster 1

Cluster 2

Cluster 3
Multi-paths and identically distributed

Both railway and avionics: Within bucket
- When identically distributed test is successful, it is successful for all paths
- When it fails, it fails only for some buckets
Composing probabilities - a representativity concern?

Set of benchmarks

Learning stage

$a_1C_1 + a_2C_2 + a_3C_3$

Measurements protocol

Utilization stage

Program

A proof of representativity requires elements from the other design levels
Representativity requires convergence

The statistical methods estimating extremes are not monotonic
The measurement protocol and the representativeness

A\textsubscript{i} is representative with respect to A if pWCET (A) is close to pWCET(A\textsubscript{i})
The reproducibility of the measurement protocol

Any two different (and complete) utilizations of the measurement protocol from the same set of execution conditions should provide the same pWCET estimate.

\[ pWCET(A_m) \approx pWCET(A_n) \]
The reproducibility of the pWCET estimation method

Any two different applications of the same set of execution conditions should provide the same pWCET estimate.
Validation of a statistical test

- Arguments compliant DO178B and IEC-61508

GEV

GPD

Estimation

Program

Threshold Block Size

Paris, France 26/02/2015

GEV

GPD

22/35
1. Dependent?
   - YES
     - 2. GEV dependent
     - 3. GPD dependent
   - NO
     - 4. GPD independent
     - 5. GEV independent

6. Sufficient variability?
   - NO
     - 7. Min(GEV, GPD)
   - YES
     - 8. Max(GEV, GPD)

pWCET estimation

INRIA confidential
Reproducibility of the pWCET method + Representativity of the measurement

Execution Conditions (1) → $A_1$

Execution Conditions (2) → $A_2$

Execution Conditions (n) → $A_n$

$pWCET(A_1) \approx pWCET(A_2) \approx \ldots \approx pWCET(A_n) \approx pWCET(A)$

Reproducibility of the pWCET method
Avionics case study

• FP7 STREP PROARTIS Case study
  – IMA application performing maintenance of the flight control computers
  – Randomized cache replacement policies
Avionics case study (2)

- Less than 5 minutes to provide a pWCET estimation

<table>
<thead>
<tr>
<th>Function</th>
<th>FUNC 1</th>
<th>FUNC 2</th>
<th>FUNC 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collected Values</td>
<td>600</td>
<td>250</td>
<td>300</td>
</tr>
<tr>
<td>Iterations</td>
<td>12</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Outline

- Design of a physical system with time constraints
  - Verification of time constraints
- Probabilities: how do we compose?
- Measurement-based approaches
  - The (misunderstood) independence
  - The impact of the measurement protocol
- Analytical vs. measurement-based
- Back to models to solve the representativity
- Conclusion
Average versus worst case

What is the impact on an analysis?

- Average number of arrivals within a time interval
  \[ \tau_1 = \begin{pmatrix} 1 & 2 & 4 \\ 0.4 & 0.3 & 0.3 \end{pmatrix}, \text{ for } t_\Delta = 12 \]

- Minimal inter-arrival times between two consecutive arrivals
  \[ \tau_1^* = \begin{pmatrix} 5 & 10 \\ 0.3 & 0.7 \end{pmatrix} \]
Optimal fixed-priority scheduler

• Rate Monotonic is not optimal

\[ \tau_1 = \left( \begin{array}{c} 1 \\ 1 \end{array} \right), 2, 2, 40\% \] \hspace{2cm} \tau_2 = \left( \begin{array}{c} 3 \\ 0.5 \end{array}, 4 \\ 0.5 \end{array} \right), 6, 6, 30\% \]
• A feasible task fixed-priority assignment

\[
\tau_1 = \left(\left(\frac{1}{1}\right), 2, 2, 40\%\right) \quad \quad \tau_2 = \left(\left(\begin{array}{cc} 3 \\ 0.5 \end{array}\right), 4, 6, 6, 30\%\right)
\]
Optimal (task) fixed-priority scheduler (3)

• Theorem (Maxim, 2011)
  The order of higher priority tasks does not have any impact on the probability of missing the deadline of a task
• Audsley reasoning may be proposed
Analytical verification of time constraints

The first response time calculation for systems with multiple probabilistic parameters (DC13)

- Probabilistic independence required between the probabilistic parameters

Analytical versus simulation

- Observed response time
- Deadline
- Calculated response time

Probability of not meeting the deadline: $9.24819 \times 10^{-14}$
Outline

✓ Design of a physical system with time constraints
  o Verification of time constraints
✓ Probabilities: how do we compose?
✓ Measurement-based approaches
  o The (misunderstood) independence
  o The impact of the measurement protocol
✓ Analytical vs. measurement-based
☐ Back to models to solve the representativity
☐ Conclusion
Design of a physical system with time constraints

Control Theory

Models

Processeur

Time Verification

x_solution

Programs/budgets

WCETs

dx/dt = f(t)

r(t)

y(t)

Controller

Probabilistic descriptions

Classification of programs

Figure 1.15: Typical execution time distribution

Therefore the amount of wasted computing power is expected to increase, leading to the over-sizing of embedded computers, power supplies and cooling systems, among other issues.

Nowadays, even if many attempts are proposed to give an upper bound of the WCET (e.g: Rochange 2007), both the traditional and current approaches are difficult to apply to modern processor generations and produce values which are more and more pessimistic.

Lately, a new approach to improve the robustness of the task scheduling has been proposed in Midonnet 2010 where on a temporal fault occurrence, the slack time (the difference between the allocated time and the real execution time) can be dynamically determined and assigned to the faulty task in order to complete its treatment. As the authors mean "deadline miss" when they talk about "faults", this approach implies that the computing resource has enough spare resource to add more time slack. This approach has two drawbacks in practice. First, the dynamic addition of the time slack will cause indeterminacy and worsen the predictability of the system. Second, from a control system point of view, it is known that delays decrease the control performance and moreover, as the time slack is dynamically added, it will result into a control system with time-varying and non bounded delay, leading to more difficulties in the analysis. On the other hand, the method proposed in this thesis allows for the enhancement of predictability and determinism of the system while enhancing control performance. This is achieved by "weakening real-time constraints".

WCETs

TOSA@TRT
Possibles steps (and open problems)

• Worst case probabilistic models
  – Understanding the relations between different design levels
  – Choice of properties to be probabilistically described
  – Proposition of new models

• Time constraints analyses

• Validation and certification of the framework
  – Proposition of a complementary transformation
CONCLUSIONS

• Time critical embedded systems are everywhere
• There is an important bareer while building tomorrow time critical embedded systems
• Proving correct such framework requires an important effort from different communities
Je vous remercie pour votre attention

liliana.cucu@inria.fr