Monitoring On-line Timing Information to Support Mixed-Critical Workloads

A. Kritikakou*, O. Baldellon[†], C. Pagetti^{*}, C. Rochange[‡], M. Roy[†] and F. Vargas[§] * ONERA, † LAAS/CNRS, ‡ Universit de Toulouse, France, § PUCRS, Brazil

Introduction

- Many- and multi-core architectures enable
 Application domain execution time of several tasks
- Critical embedded systems, e.g. Aeronautical systems, impose static Worst Case Execution Time (WCET) to ensure safety

Memory accesses are upper bound considering a fully congested memory bus.

 \geq Safe, but unusable bounds

Target system

- - > A critical task T_c with nested loops and manifest conditions and with computable WCET
 - > A set of less critical tasks
- Platform domain
 - Time-predicable bus-based multicore system, especially FPGA with hardware

Off-line analysis

- Annotated CFG with monitoring points at the first binary instruction of each basic block
- Nested level of monitoring points
- Integer Linear Programming (ILP) formulation to compute partial WCET between monitoring points for *isolation scenario*, i.e. w_{x} and $d_{xx'}$ level

>Overwhelmingly conservative schedules

Motivation

- Consider a critical task T_c and a less critical task T₁ scheduled over two cores
- **Existing approaches:**

Maximum load execution: When both tasks are executed the WCET_{TC}>D_C (*Fig.1a*)

Isolation execution: When only TC is executed the WCET_{TC} $< D_C$, but with the core utilization factor is low (*Fig.1b*)

Proposed methodology: Maximum load as long as $WCET_{TC} < D_C$ is execution guaranteed to be met, increasing the core utilization factor. Otherwise, isolation *execution* is used (*Fig.* 1*c*)

monitoring capabilities

Proposed Methodology

Off-line analysis:

>Binary code of T_C is presented as Control Flow Graph (CFG) The CFG is analyzed for several monitoring points to estimate partial WCET for *maximum load scenario* and isolation scenario

On-line decision:

>Hardware monitor to observe the real execution time of T_c

> Check if a risk exists that the T_c misses its deadline, and if so, apply *isolation*



Fig.3: *CFG* with nesting levels and partial WCETs

On-line decision

• Monitor real execution time of $T_C: ET(b)$ Compute remaining WCET in *isolation* execution: $RWCET_{iso}(b)$

Check condition for safety:

 $ET(b) + RWCET_{iso}(b) + WCET_{mon} + t_{over} > D_C$

* If condition holds, switch to *isolation* execution



Fig.1: Motivation example: Scheduling based on WCET when are considered for execution (a) both tasks, (b) only the critical task and (c) proposed hybrid methodology.

execution





Fig.2: Overview of the proposed methodology













