CISTER
Research Center in
Real-Time & Embedded
Computing Systems

Technical Report

A Multi-DAG Model for Real-Time Parallel Applications with Conditional Execution

José Fonseca*
Vincent Nélis*
Gurulingesh Raravi
Luis Miguel Pinho*

CISTER-TR-141207

2015/04/13

A Multi-DAG Model for Real-Time Parallel Applications with Conditional Execution

José Fonseca*, Vincent Nélis*, Gurulingesh Raravi, Luis Miguel Pinho*

*CISTER Research Center

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: jcnfo@isep.ipp.pt, nelis@isep.ipp.pt, guhri@isep.ipp.pt, lmp@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

Owing to the current trends for higher performance and the ever growing availability of multiprocessors in the embedded computing (EC) domain, there is nowadays a strong push towards the parallelization of modern embedded applications. Several real-time task models have recently been proposed to capture different forms of parallelism. However, they do not deal explicitly with control flow information as they assume that all the threads of a parallel task must execute every time the task is activated. In contrast, in this paper, we present a multi-DAG model where each task is characterized by a set of execution flows, each of which represents a different execution path throughout the task code and is modeled as a DAG of sub-tasks. We propose a two-step solution that computes a single synchronous DAG of servers for a task modeled by a multi-DAG and show that these servers are able to supply every execution flow of that task with the required cpu-budget so that the task can execute entirely, irrespective of the execution flow taken at run-time, while satisfying its precedence constraints. As a result, each task can be modeled by its single DAG of servers, which facilitates in leveraging the existing single-DAG schedulability analyses techniques for analyzing the schedulability of parallel tasks with multiple execution flows.