Science of Computer Programming ••• (••••) •••–•••

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Towards optimal solutions for the low power hard real-time task allocation on multiple heterogeneous processors

Eduardo Valentin *, Rosiane de Freitas, Raimundo Barreto

Graduate Program in Informatics (PPGI), Federal University of Amazonas (UFAM), Av. General Rodrigo Otávio, 6.200, Coroado I, Cep: 69080-900, Manaus-AM, Brazil

ARTICLE INFO

Article history:
Received 27 January 2017
Received in revised form 12 August 2017
Accepted 14 August 2017
Available online xxxx

Keywords:
Hard real-time
MGAP
Schedulability
DVFS

ABSTRACT

The usage of heterogeneous multicore platforms is appealing for applications, e.g. hard real-time systems, due to the potential reduced energy consumption offered by such platforms. However, even in such platforms the power wall phenomena still imposes limits to performance. Hard real-time systems are part of life critical environments and reducing the energy consumption on such systems is an onerous and complex process. We tackle the problem from the perspective of different representative integer programming mathematical formulations and their interplay on the search for optimal solutions for Rate Monotonic (RM) and Earliest Deadline First (EDF) scheduling algorithms. The proposed models are based on a well-established formulation in the operational research literature, namely, the Multilevel Generalized Assignment Problem (MGAP). This paper, therefore, assesses the problem of finding optimal allocations and frequency assignments of hard real-time tasks among heterogeneous processors targeting low power consumption, but taking into account timing constraints. Computational experiments show that finding optimal solutions reduces the estimated energy consumption of the evaluated cases when compared to state-of-the-art algorithms.

1. Introduction

The power wall is a barrier to improvement in the processor design process due to the power consumption of components. Power consumption has become the primary influence in overall microprocessor design complexity [28], due to ideal geometric scaling and non-ideal electrical scaling. It is no longer viable to simply increase clock speeds of existing designs [11]. Power consumption is a major aspect that limits the performance of computers in different sides of the computing spectrum. The pursuit of energy efficiency is useful, for instance, for mobile devices to improve operating duration and also helpful for server systems to reduce power bills [10].

One of the strategies to avoid the power wall is to adopt multiple processing elements to enhance the computing capability and to reduce the power consumption, especially for embedded systems [10]. Besides, modern multicore processors for the embedded market are often heterogeneous in nature [3]. Therefore, the heterogeneous multicore platforms have become the *de-facto* solution to cope with the rapid increase of system complexity, reliability, and energy consumption [16].

Practitioners execute applications with hard deadline restrictions on multiple heterogeneous processors due to the expected energy consumption reduction. Nevertheless, developing software with timing constraints for multiple heterogeneous

* Corresponding author.
E-mail addresses: eduardo.valentin@icomp.ufam.edu.br (E. Valentin), rosiane@icomp.ufam.edu.br (R. de Freitas), rbarreto@icomp.ufam.edu.br (R. Barreto).

http://dx.doi.org/10.1016/j.scico.2017.08.005
0167-6423/

processors is a complex task. Scheduling becomes especially hard to deal with, particularly under low power constraints. Our approach aims at life-critical hard real-time systems such as unmanned aerial vehicles, control system in the automotive area, and distributed computing under severe constraints, e.g., tracking and target monitoring, military, and environmental remote monitoring.

A classical mathematical model that resembles modern heterogeneous multicore platforms is the Multilevel Generalized Assignment Problem (MGAP), even though it was originally conceived in the manufacturing context. The MGAP consists of minimizing the assignment cost of a set of jobs to machines, each having associated therewith a capacity constraint. Each machine can perform a job with different performance states that entail different costs and amount of resources required. The MGAP is originally in the context of large manufacturing systems as a more general variant of the well-known Generalized Assignment Problem (GAP) [13]. In this paper, we correlate MGAP model with the problem of assigning frequencies and distributing hard real-time tasks on heterogeneous processors minimizing energy consumption.

Modern processors may be seen as machines with several performance states due to Dynamic Voltage and Frequency Scaling (DVFS) technique. DVFS is a well established power reduction strategy and it has already been a research topic for decades. The premises are the variation of processors' workloads and the quadratic relationship between energy consumption and voltage [7]. The energy consumption depends on dynamic and idle energy [28]: $E_{system} = E_{dyn} + E_{idle}$, where E_{idle} is the energy consumption while the system is idle and accounts for leakage, E_{dyn} is the energy consumption in active use cases. The dynamic energy consumption E_{dyn} is estimated using: $E_{dyn} = C_l \times N_{cycle} \times V_{dd}^2$, where E is energy, C_l is circuitry capacitance, N_{cycle} is number of cycles, and V_{dd} is the voltage. Although DVFS yields meaningful energy consumption reduction, its usage requires care, especially when considering timing constraints.

The implementation of DVFS-capable chips creates three types of platforms: full-chip, per-core, and cluster-based [16]. The categories differ depending on the Dynamic Phased Lock Loop (DPLL) network across the circuit and on the voltage delivery distribution. In full-chip platforms, the design allows changing the clock, and voltage, of all cores at once. The per-core platform, in contrast, implements a DPLL network to achieve frequency (and voltage) manipulation granularity on each individual core. Cluster-based architecture is a generalization of full-chip and per-core platforms. In cluster-based platforms, clusters group the cores, where each cluster acts as a full-chip platform. But clusters are independent of each other having their own clock and voltage network. The cluster-based platforms allow changing clock and voltage of each cluster independently, but the change affects all cores within the cluster. In this paper, we are only considering per-core multicore platforms.

Therefore, the problem we are addressing in this paper is: *how to find optimal hard real-time tasks distribution among heterogeneous processors respecting timing constraints and targeting low power consumption?* The contributions of this work are: (i) comprehensive and representative mathematical formulations that (ii) accounts characteristics of different hard real-time scheduling policies and that (iii) delivers optimal hard real-time task allocation and optimal frequency to task assignment, (iv) with system energy consumption minimization, but still (v) using the advantage of a classical combinatorial optimization model: MGAP. The results we present on this research question focus on solving the problem optimally on practical instances sizes. Effective methods support reducing power bills, improve system reliability, and increase the efficient usage of energy; last but not least, assisting to reduce environmental impacts.

The organization of this paper is as follows. The processor model and task model are defined in Section 2. Formulations for different scheduling policies and a model growth analysis are discussed in Section 3. We describe the implementation of solvers and of a evolutionary algorithm in Section 4. Computational experiments are detailed in Section 5. We compare our results with existing literature in Section 6. Section 7 closes this paper with final comments and future work.

2. System models

In this section we present the system models. Section 2.1 describes the processor model we consider. We describe the real-time task model in Section 2.2.

2.1. Processor model

The processor model resembles a Multi-Processor System-On-Chip (MPSoC) architecture, such as Exynos 5 Octa [23]. The system is composed by a set, $\mathcal{H}$, of m processors, $\mathcal{H} = \{H_1, H_2, \ldots, H_m\}$. Each core may operate on l different performance states, $1 \leq k \leq l$. The set of frequencies of one core is not necessarily the same of other cores. Also, a task may have different code size and execution time for different processors, due to instruction set differences. The frequency of performance state k on the processor i is F_{ik} and the power consumption is P_{ik}. The idle power of processor i is $P_{idle,i}$.

Our proposal can be used with no extra effort on other architectures. Even though we focus on per-core heterogeneous platforms in our experiments, we have exercised on multiple heterogeneous clusters [26]. The models discussed here may be applicable to full-chip and cluster-based platforms as long as the intrinsic architectural interference is accounted and we recommend the interested reader to consider more sophisticated schedulability tests [26].

Please cite this article in press as: E. Valentin et al., Towards optimal solutions for the low power hard real-time task allocation on multiple heterogeneous processors, Sci. Comput. Program. (2017), http://dx.doi.org/10.1016/j.scico.2017.08.005