The Journal of Systems and Software 113 (2016) 44–58

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Scalable and efficient configuration of time-division multiplexed resources

Anna Minaeva [a,c,*], Přemysl Šůcha [a], Benny Akesson [a,b], Zdeněk Hanzálek [a,c]

[a] Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
[b] CISTER/INESC TEC and ISEP, Portugal
[c] Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague, Czech Republic

ARTICLE INFO

Article history:
Received 4 June 2015
Revised 1 October 2015
Accepted 12 November 2015
Available online 19 November 2015

Keywords:
Real-time systems
Resource scheduling
Branch-and-price

ABSTRACT

Consumer-electronics systems are becoming increasingly complex as the number of integrated applications is growing. Some of these applications have real-time requirements, while other non-real-time applications only require good average performance. For cost-efficient design, contemporary platforms feature an increasing number of cores that share resources, such as memories and interconnects. However, resource sharing causes contention that must be resolved by a resource arbiter, such as Time-Division Multiplexing. A key challenge is to configure this arbiter to satisfy the bandwidth and latency requirements of the real-time applications, while maximizing the slack capacity to improve performance of their non-real-time counterparts. As this configuration problem is NP-hard, a sophisticated automated configuration method is required to avoid negatively impacting design time.

The main contributions of this article are: (1) an optimal approach that takes an existing integer linear programming (ILP) model addressing the problem and wraps it in a branch-and-price framework to improve scalability. (2) A faster heuristic algorithm that typically provides near-optimal solutions. (3) An experimental evaluation that quantitatively compares the branch-and-price approach to the previously formulated ILP model and the proposed heuristic. (4) A case study of an HD video and graphics processing system that demonstrates the practical applicability of the approach.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

The trend of consumer-electronics systems becoming more and more complex is not possible to overlook. An increasing number of applications has resulted in a transition to multi-core platforms, where the number of cores and hardware accelerators has been growing exponentially; a trend that is expected to continue in the coming decade (ITRS, 2011). The applications have different types of requirements, as illustrated in Fig. 1. Some of them (colored white in the figure) have *real-time requirements* and must always satisfy their deadlines, while other non-real-time applications (colored gray) only require sufficient average performance (Van Der Wolf and Geuzebroek, 2011). The cores and accelerators access shared resources, such as memories, interconnects and peripherals (Kollig et al., 2009; Van Berkel, 2009) on behalf of the applications they execute and are referred to as resource *clients*. However, this resource sharing causes *contention* between clients that must be resolved by an *arbiter*. Time-Division Multiplexing (TDM) is a commonly used arbiter that schedules clients based on a statically computed schedule with a fixed number of time slots.

An important challenge with TDM arbitration in these systems is to find a schedule that assigns the time slots to the clients in a way that satisfies the *bandwidth and latency requirements* of the real-time clients, while *minimizing their resource utilization* (maximizing slack capacity) to improve performance of the non-real-time clients. This configuration process is required to complete in reasonable time, even for large problems, to avoid negatively impacting the design time of the systems, which is required to stay unchanged (ITRS, 2011). This is particularly important since the schedule configuration may be a part of the design-space exploration, thus requiring it to be repeated many times during system design.

The four main contributions of this article are: (1) We present an exact approach that takes an existing integer linear programming (ILP) model addressing the TDM configuration problem and wraps it in a branch-and-price framework (Feillet, 2010) to improve its scalability. The computation time of this algorithm is optimized using several techniques, including lazy constraints generation. (2) We present a stand-alone heuristic algorithm that can be used to solve

* Corresponding author. Tel.: +420776879621.
E-mail addresses: anna.minaeva@cvut.cz (A. Minaeva), suchap@fel.cvut.cz (P. Šůcha), kbake@isep.ipp.pt (B. Akesson), zdenek.hanzalek@cvut.cz (Z. Hanzálek).

http://dx.doi.org/10.1016/j.jss.2015.11.019
0164-1212/© 2015 Elsevier Inc. All rights reserved.

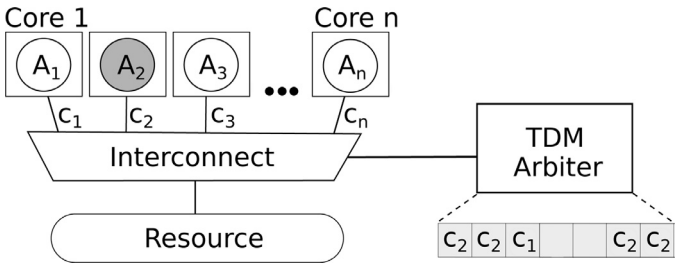

Fig. 1. Example of a multi-core system, where n applications (A_i) with different real-time requirements are mapped to the cores. The cores act as resource clients (c_i) accessing a shared resource through an interconnect controlled by a TDM arbiter.

the problem, providing a trade-off between computation time and efficiency that is useful when sub-optimal solutions are acceptable. (3) We experimentally demonstrate the improved scalability of the branch-and-price approach and compare it both to the previously formulated ILP model and to an existing heuristic. We also quantify the trade-off between efficiency and computation time for the optimal and heuristic algorithms. (4) We demonstrate the practical relevance of the approach by applying it to a case study of an HD video and graphics processing system. In addition, the source code of our approach is released as open-source software and can be found in Minaeva et al. (2015).

The rest of this article is organized as follows. Related work is discussed in Section 2. Section 3 proceeds by presenting background information necessary to understand the main contributions of the article. Then, the configuration problem is formalized in Section 4, followed by a description of the existing ILP formulation in Section 5. The branch-and-price approach is introduced in Section 6 and its computation time optimizations are discussed in Section 7. The heuristic algorithm for slot assignment is then explained in Section 8. Section 9 presents the experimental evaluation, before we demonstrate the practical applicability of our approach in a case study in Section 10. Lastly, the article is concluded in Section 11.

2. Related work

Scalability is a critical issue in system design, since design time must remain unchanged despite an exponential increase in system complexity. Most works in the area of design automation that use exact optimization techniques do not scale well enough to be able to manage the complexity of future consumer electronics systems (Lukasiewycz et al., 2012; Gomony et al., 2015a; Yi et al., 2009; Lin et al., 2012; Hanzalek et al., 2010), and only a few propose advanced techniques to address the complexity problem. These techniques can be classified into two major groups of approaches: 1) a decomposition of the problem into smaller sub-problems, and 2) navigating the search smartly during design-space exploration. The first approach deals with large problems by decomposing them into many smaller problems. This method is used in Wildermann et al. (2014), Liu et al. (2008). The second branch of improvements uses problem-specific information while searching the design space, which is more efficient compared to using general design-space exploration methods. Examples of this approach are shown in Reimann et al. (2011) and Lukasiewycz and Chakraborty (2012), where the authors look for a minimal reason for constraint violation and prevent this situation in the rest of the search, while using boolean satisfiability and ILP approaches, respectively.

Another way of dealing with the scalability issue is to use a heuristic approach. Some methodologies to configure TDM arbiters have been proposed in the context of off-chip and on-chip networks. An approach for synthesizing TDM schedules for TTEthernet with the goal of satisfying deadlines for time-triggered traffic, while minimizing the latency for rate-controlled traffic is proposed in Tamas-Selicean et al. (2012). The methodologies in Hansson et al. (2007), Lu and Jantsch (2007) consider slot assignment in contention-free TDM networks-on-chips. All of these approaches are heuristics and the efficiencies of the proposed methods have not been quantitatively compared to optimal solutions. Furthermore, the problem of scheduling networks is different from ours, as it considers *multiple resources* (network links) and is dependent on the problem of determining paths through the network.

The problem of TDM arbiter configuration with simplified client requirements is considered in Hassan et al. (2015), where unlike this work, the authors propose a harmonic scheduling strategy. One of the two previous solutions to the problem considered in this article is the configuration methodology for multi-channel memory controllers in Gomony et al. (2015a). The authors apply a commonly used heuristic for TDM slot assignment, called *continuous allocation* (Goossens et al., 2013b; Foroutan et al., 2013; Goossens et al., 2013a; Vink et al., 2008), where slots allocated to a client appear consecutively in the schedule. The reasons for its popularity are simplicity of implementation and negligible computation time of the configuration algorithm. However, with growing problem sizes, this strategy results in significant over-allocation, making satisfaction of a given set of requirements difficult. This is experimentally shown in Section 9 when comparing to our approach. The considered TDM configuration problem was furthermore addressed in Akesson et al. (2015), where an ILP formulation is proposed to solve the problem. Although this solution shows good results for systems of the past and present, complex future systems require a more scalable approach to avoid negatively impacting design time.

Besides the difference in the problem formulation, this article advances the state-of-the-art by being the *first to apply a theoretically well-founded advanced optimization approach, called branch-and-price* (Feillet, 2010) *in the field of consumer-electronics systems design*. Branch-and-price combines both of the mentioned approaches to manage complexity; it decomposes the problem into smaller sub-problems and uses more sophisticated search-space exploration methods. Although Schenkelaars et al. (2011) applies branch-and-price to the problem of FlexRay scheduling in the automotive domain, this article gives more elaborate explanation of the approach and concentrates on the computation time optimizations. Moreover, *this work extends* (Akesson et al., 2015) *by using the previously proposed ILP model (Section 5) as a building block in the novel branch-and-price framework* (Sections 6, 7 *and* 8) *to improve its scalability to satisfy the needs of future design problems.*

3. Background

This section presents relevant background information to understand the work in this article. First, we present the concept of latency-rate servers, which is an abstraction of the service provided to a client by a resource arbiter. We then proceed by discussing how TDM arbitration fits with this abstraction and explain how to derive its latency and rate parameters.

3.1. Latency-rate servers

Latency-rate ($\mathcal{LR}$) (Stiliadis and Varma, 1998) servers is a shared resource abstraction that guarantees a client $c_i$sharing a resource a minimum allocated rate (bandwidth), ρ_i, after a maximum service latency (interference), Θ_i, as shown in Fig. 2. The figure illustrates a client requesting service from a shared resource over time (upper solid red line) and the resource providing service (lower solid blue line). The $\mathcal{LR}$ service guarantee, the dashed line indicated as service bound in the figure, provides a lower bound on the amount of data that can be transferred to a client during any interval of time.

The $\mathcal{LR}$ service guarantee is conditional and only applies if the client produces enough requests to keep the server busy. This is captured by the concept of *busy periods*, which intuitively are periods

Download English Version:

https://daneshyari.com/en/article/459238

Download Persian Version:

https://daneshyari.com/article/459238

Daneshyari.com